Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 142 (2017)

On small-depth Frege proofs for Tseitin for grids

Johan Håstad (*johanh@kth.se*)
KTH Royal Institute of Technology

October 31, 2018

Abstract

We prove that a small-depth Frege refutation of the Tseitin contradiction on the grid requires subexponential size. We conclude that polynomial size Frege refutations of the Tseitin contradiction must use formulas of almost logarithmic depth.

1 Introduction

This paper is in the setting of propositional proof complexity. We are given a propositional statement and some reasoning rules. The most basic proof system is resolution. In this proof system we study clauses, i.e. disjunctions of literals and have a simple way to derive new clauses from existing clauses. If we derive the empty clause we have reached a contradiction refuting the original formula.

Resolution has been studied extensively and by now we have a large body of work understanding the strengths and limitations of resolution. In an early paper [17], Tseitin defined the set of contradictions based on graphs studied in this paper and proved that any regular resolution proof of this contradiction requires exponential size proofs in general. A later result by Haken [7] gave the first strong lower bound for unrestricted resolution proving that the pigeon-hole principle (PHP) requires exponential size proofs. As this paper is not about resolution let us not discuss the many strong results obtained but only mention the paper of Ben-Sasson and Wigderson [4] as a high point which in particular established the importance of width when studying resolution proofs.

There are many proof systems which are more powerful than resolution and in this paper we study the case when each formula appearing in the proof is restricted to be a Boolean formula of small depth d. Here $d = 1$ essentially corresponds to resolution. There are many alternatives for the reasoning rules and what is said below applies to any constant size set of reasoning rules that are consistent. The first strong result in this setting was obtained by Ajtai [1] showing that the PHP cannot be proved in constant depth and polynomial size.

Ajtai did not give an explicit lower bound for the depth of polynomial size proofs but in a later reformulation by Bellantoni et al. [2], a lower bound of $\Omega(\log^* n)$ was given. This was later strengthened [11, 9] to obtain $\Omega(\log \log n)$

ISSN 1433-8092

lower bounds for PHP. Similar bounds were later proved by Urquhart and Fu [18] and Ben-Sasson [3] for Tseitin contradictions for the complete graph and for constant-degree expander graphs, respectively.

On the positive side Buss [5] proved that there are polynomial size $O(\log n)$-depth proofs for the PHP and similar proofs can be constructed by similar methods for the Tseitin contradiction for any constant-degree graph.

The exponential gap between the depth bounds $\log \log n$ and $\log n$ was recently partly closed by Pitassi et al. [12] obtaining a $\Omega(\sqrt{\log n})$ lower bound for Tseitin contradictions on a certain 3-regular expander graph. It is curious to note the size lower bounds of [12] when considering depth d is exponential in $\Omega((\log n)^2/d^2)$ and thus only weakly superpolynomial. For small values of d, this bound is weaker than the bounds of the form $exp(n^{c^{-d}})$ obtained in previous paper but extends the range of d for which the bound is superpolynomial.

In the current work we study the Tseitin contradictions for the 2-dimensional grid and almost close the gap obtaining size lower bounds $exp(\Omega(n^{1/58(d+1)}))$ for depth d proofs and hence the depth lower bound $\Omega(\log n/\log \log n)$ for polynomial size proofs. Our proofs follow the same paradigm as earlier proofs and let us sketch the underlying mechanisms at a semi high level to put our contribution in perspective.

When studying circuits of small depth it has turned out to be profitable to study restrictions that fix most of the input variables to constants. This is useful as for suitably chosen restrictions it is possible to decrease the depth of almost all small circuits by one. This was first used to prove lower bounds for circuit-size [6, 16, 19, 8] and the simplest case is when proving lower bounds for the size of depth-d circuits computing parity. Let us briefly discuss this case.

In this situation one uses the simplest space of random restrictions usually denoted by R_p. In such a restriction, each input variable is, independently of all other variables, kept with probability p and otherwise set to 0 or 1 with equal probabilities. The key notion for decreasing depth is a switching lemma which says that if you are given a depth two circuit with bottom fanin t then, if you at the same time apply a restriction, it can be switched to a depth 2 circuit of the other type of bottom fanin s, except with probability at most $(5pt)^s$.

Using this switching property for the two layers closest to the input creates two adjacent layers of gates of the same type which makes it possible to decrease the depth of the circuit by one. To prove a lower bound for parity one just needs to make the trivial observation that the resulting circuit must compute the parity (or the negation) of the remaining variables. Applying $d-1$ restrictions we are able to make the circuit simple enough to be analyzed directly. The number of remaining variables is about $p^{d-1}n$ and we need a large enough p to make this this number non-trivial.

To prove lower bounds for the size of proofs for various families of formulas one needs more subtle restrictions. We are no longer computing a function but instead given a set of axioms. We want that a restriction reduces the problem to a smaller problem of the same type. This is more or less equivalent to that each axiom is either reduced to an axiom of the smaller instance or to something that is a tautology. We must, at all cost, make sure that no axiom is made false as this would imply that the contradiction we are trying to prove cannot be produced, is available as an axiom. In most cases each axiom is of constant size and this implies that we cannot use restrictions, such as those of R_p, that treat the variables independently. Restrictions that give values in a dependent way cause problems with the proof (or even validity) of the switching lemma. The key is thus finding a balance between the property of preserving the axioms of the formula we are studying while still being able to prove a switching lemma with good parameters.

Note that this is a balance to be kept as when studying k-CNF formulas, we need to preserve these particular clauses while switching implies that we can simplify all functions defined by depth-d circuits. This is not, however, as impossible as it sounds as we are allowed to make most clauses true while making sure that a small fraction of the clauses remain undetermined. We must, however, as stated above, avoid making any clause false.

On the high level, the strength of a switching lemma is controlled by the size of the smaller instance obtained (which corresponds to the parameter p for independent restrictions) and how the failure probability depends on the parameters s and t. To fully understand the tradeoffs possible here requires very detailed understanding of the space of restrictions but let us give some superficial remarks.

In most situations, the probability of keeping a variable must be lower than the probability of it taking either the value 0 or 1. When the two values are balanced this is not a severe problem. For the PHP, however, where a variable taking the value 1 signifies that a particular pigeon flies to a particular hole this is a limiting factor. In fact this leads to choices corresponding to $p = n^{-c}$ for some positive constant c. This implies that the size of the problem goes from n to n^{1-c} in order to reduce the depth of the formulas in the proof by one. This can only be repeated $O(\log \log n)$ times before the problem becomes trivial. This is a bottleneck in some previous arguments.

The set of formulas introduced by Tseitin on a graph G has variables corresponding to edges and the formula says[1] that the edges adjacent to a node sum to one modulo two. For any odd sized graph this is a contradiction. For assignments to variables satisfying these conditions locally, 0 and 1 are symmetric and hence the problem of biased bits does not exist for the Tseitin formulas.

The switching lemma of [12], however, has failure bounds on the form $(cpt2^t)^s$. The reason for the factor 2^t is a bit mysterious and indeed [12] conjectures that it is not needed. We note that the paper by Mehta [10] describes similar situations where the factor is indeed needed.

We are not quite able to get optimal parameters in the current proof but we do improve the troublesome factor 2^t of [12] to t^c for a constant c. This implies that the loss in one application of the switching lemma roughly corresponds to c applications of the lemma with the optimal parameters and thus we get this

[1] For readers familiar with this family we are using the case when all charges are one as opposed to the general case.

multiplicative factor in front of d. As this is a constant we get asymptotically sharp bounds for the depth of polynomial size proofs.

A key point in the proof is the choice of the space of restrictions. The high level picture is not surprising. Given a $n \times n$ grid we pick sub-squares of size $T \times T$ (where T is poly-logarithmic when studying polynomial size proofs and $n^{\Omega(1/d)}$ in general) and in each sub-square we pick a node and connect the picked nodes by paths. For each path P we have a new variable x_P, and for each edge e on P the variable x_e is either replaced by x_P or its negation $\bar{x}_P$. This is done in a way that independent of the values of these new variables all constraints, except at the picked nodes are automatically satisfied while the constraints at the picked nodes give the constraints of the smaller instance.

In order to be able to prove a switching lemma we have to be slightly careful. First of all, as we have limited independence it turns out to be easier to use a labeling argument of Razborov [13] as opposed to a reasoning with conditional probability of Håstad [8]. Once we have found some variable that is still alive, the rather rigid topology of the grid reveals other variables that are likely to be alive. It is advantageous for the analysis if we can immediately tell which other variables are also alive, and if these depend on the same remaining variable, these are essentially for free. The easiest way to achieve this would be that any edge determines the entire path on which it lies. This is impossible to achieve in a constant degree graph such as the grid, as edges close to the picked nodes must lie on many different paths. For the paths that we use this is the only part of the paths that intersect and this limited ambiguity of which path(s) an edge might belong to can be handled. An important property is that even though an edge can lie on many paths, we are able to make sure that all these paths share an endpoint and this is sufficient for the argument.

The essential new part of the current paper is the choice of restrictions and the proof of the switching lemma. The way to analyze how restrictions make all sub-formulas be represented by small-depth decision trees is done as in previous papers.

An overview of the paper is as follows. We start with some preliminaries in Section 2 and proceed with some properties of the grid and assignments that satisfies some parity conditions in Section 3. We define our restrictions in Section 4. The final, full, restriction is picked by a two-stage process. We first pick a relatively small but fairly dense set of nodes to be potentially used by the restriction. The key property here is that they can be picked independently and we can still be sure that each sub-square has roughly the expected number of potential surviving nodes. We may then, in the second stage, pick one of the nodes to be the actual survivor in essentially any way. The first independent picking of surviving nodes is the main probabilistic event that is analyzed in the switching lemma.

We proceed to recall the formalism of t-evaluations in Section 6 after having described some basic properties of consistent decision trees in Section 5. Assuming the switching lemma we are able to complete the proof of our main theorem also in Section 6 and we end by the proof of the switching lemma in Section 7.

2 Some preliminaries

We have a graph G which we call "the grid" but to avoid problems at the perimeter we in fact use the torus. In other words we have nodes indexed by (i,j), for $0 \leq i,j \leq n-1$ where n is an odd integer and a node (i,j) is connected to the four nodes at distance 1, i.e. where one coordinate is identical and the other moves up or down by 1 modulo n. For each node v we have a *charge* α_v and for each edge e in the graph we have a variable x_e. A Tseitin formula is given by a set of linear set of equalities modulo 2. In particular for each v we have

$$\sum_{e \ni v} x_e = \alpha_v.$$

The main case we consider, which we call "the Tseitin contradictions" is when $\alpha_v = 1$ for each v. We do use more general charges in intermediate steps and hence the following lemma is useful for us.

Lemma 2.1 *Consider the Tseitin formulas with charges α_v. If $\sum_v \alpha_v = 0$ this formula is satisfiable and has 2^{r_n} solutions where the positive integer r_n depends only on n and not on the value of α_v.*

Proof: Let us first establish that the system is satisfiable. Take any assignment to all variables x_e and suppose we have at least two nodes v_1 and v_2 whose constraints are violated. Take a path connecting v_1 and v_2 and negate all variables on this path. This new assignment satisfies the constraints at v_1 and v_2 and does not change the validity at any other node, as for other nodes either zero or two adjacent variables change their values. We can repeat this process until at most one constraint is violated. Summing all constraints shows that the number of violated constraints is even and thus in fact all constraints must be satisfied at the end of this process.

As the number of satisfying assignments to a satisfiable system of linear equations does not depend on the right hand sides, the other part of the lemma is immediate. ∎

As a converse to the above lemma, when $\sum_v \alpha_v = 1$ it is easy to see, by summing all equations, that the system is contradictory. In particular the Tseitin contradictions with $\alpha_v = 1$ for all v are indeed contradictions for graphs with an odd number of nodes. We note that each Tseitin formula can be written as a 4-CNF formula by having 8 clauses of length four for each node.

We are interested in proofs in the form of deriving the constant false from these axioms. The exact reasoning rules turn out not to be of central importance but are stated in Section 6. The important properties of these rules are that they are sound and of constant size.

The sub-formulas that appear in this proof are allowed to contain only $\vee$-gates and negations. We simulate $\wedge$ using $\wedge F_i = \neg \vee \neg F_i$ and we define the depth of a formula to be the number of alternations of $\vee$ and $\neg$.

3 Properties of assignments on the grid and dynamic matchings

We are interested in solutions to subsystems of the Tseitin contradictions. It follows from Lemma 2.1 that as soon as we drop the constraints at a single node we have a consistent system and indeed many solutions.

On a set X of nodes we say that a partial assignment is *complete* if it gives values to exactly all variable with at least an endpoint in X. The support of a partial assignment is the set of nodes adjacent to a variable given a value. Note that the support of a complete assignment on X also includes the neighbors of X.

We consider partial assignments that give values to few variables and in particular we are interested in cases where the size of the set X is at most $n/2$ and hence cannot touch all rows or columns of the grid. Let X^c denote the complement of X.

In this case, X^c contains a giant component containing almost all nodes of the grid. This follows as there are at least $n/2$ complete rows and columns in X^c and the nodes of these rows and columns are all connected. The other, small, components of X^c are important to control as an assignment on X might fail to extend in a consistent way to such a component. To avoid this problem for a set X we let the *closure of* X, $cl(X)$ denote all nodes either in X or in small connected components of X^c. Note that $cl(X)^c$ is exactly the giant component of X^c.

Definition 3.1 *An assignment α supported on a set X is* locally consistent *if it can be extended to a complete assignment on $cl(X)$ that satisfies all parity constraints on this set.*

We extend this definition to say that two assignments are consistent with each other if they do not give different values to the same variable and when you look at the union of the two assignment this gives a locally consistent assignment. Let us prove a lemma that is fairly obvious but still central for our argument.

Lemma 3.2 *Suppose α is a locally consistent assignment supported on a set of size at most $n/2$ and x_e a variable not in the support of α. Then there is a locally consistent assignment α' that extends α and gives a value to x_e.*

Proof: Suppose the support of α is X and let X^+ be X with the endpoints of e added. First extend α to be an assignment that satisfies the constraints on $cl(X)$ and then take any further extension that gives values to all variables touching $cl(X^+)$. Suppose this assignment violates the parity constraint at a node v. Take a path that starts at v and ends in the giant component of $cl(X^+)^c$ and does not pass through any node in $cl(X)$. This is possible as $cl(X)^c$ is connected and the given assignment satisfies all constraints on $cl(X)$ and hence $v \in cl(X)^c$. Negate the variables corresponding to edges on this path. The new assignment satisfies the constraint at v, still extends α and does not cause any new violations on $cl(X^+)$. Repeating this procedure for any $v \in cl(X^+)$ that has its constraint violated creates a locally consistent assignment that extends α and gives a value to x_e. ■

A process that is important for us is the following dynamic matching game. We have two players, one adversarial player that supplies nodes while the other, matching player P_M, is supposed to dynamically create a matching that contains the nodes given by the adversarial player. As the full grid is of odd size and hence does not have a perfect matching the adversarial player will eventually win, but clearly P_M can survive for a while and this will be sufficient for us. To be more precise we have the below lemma.

Lemma 3.3 *When the dynamic matching game is played on the $n \times n$ grid, P_M can survive for at least $n/2$ moves.*

Proof: P_M maintains a matching of part of the grid (containing the supplied nodes and some extra nodes) and if the supplied node is in the support of this matching P_M gives the already predetermined answer. If this is not the case then P_M needs to extend the matching.

The partial matching matches a set which is a cross-product of a set R of rows and a set C of columns. We maintain the property that both these sets are the unions of a number of intervals each of even size. To avoid a degenerate case we start with R and C both being two adjacent points covering the first node supplied by the adversary.

Faced with a node (x, y) outside this set, P_M, proceeds as follows. If x is not in R then P_M adds x to R and as the matching P_M adds pairs (x, c), (x, c') with c and c' adjacent to cover $x \times C$. This is easy as C is a union of intervals of even size. This process makes R have exactly one interval of odd size. This might be the singleton x or a longer interval if x was adjacent to an interval already in R. In either case it is easy to find an x' to add to R to make this interval of even size. This might cause two intervals of R to merge but as the union of two intervals of even size is an interval of even size, this is not a problem. A matching on $x' \times C$ is found and added to complete the process of adding rows.

Turning to columns, if $y \in C$ we are done but it this is not the case we can add two columns in an analogous way. As we add at most two rows and two columns in each step the described process can go on for at least $n/2$ steps. ■

4 Restrictions

The plan is to make a probabilistic assignment to variables of the grid that reduces the Tseitin contradiction to a smaller contradiction of the same type in a way that enables us to simplify all formulas appearing in an attempted proof. As the final product is a rather rigid object we utilize an intermediate partial restriction that leaves slightly more variables unset but has better independence properties. We start by defining the full restrictions.

4.1 Full restrictions

In an $n \times n$ grid we make sub-squares of size $T \times T$ where T is odd. In each sub-square we choose[2] $\Delta = \sqrt{T}/2$ of the nodes and call them *centers*. These are located evenly spaced on the diagonal of the $3T/4 \times 3T/4$ central sub-square. This implies that they have separation $3\sqrt{T}/2 = 3\Delta$ in both dimension. A schematic picture of this is given in Figure 1.

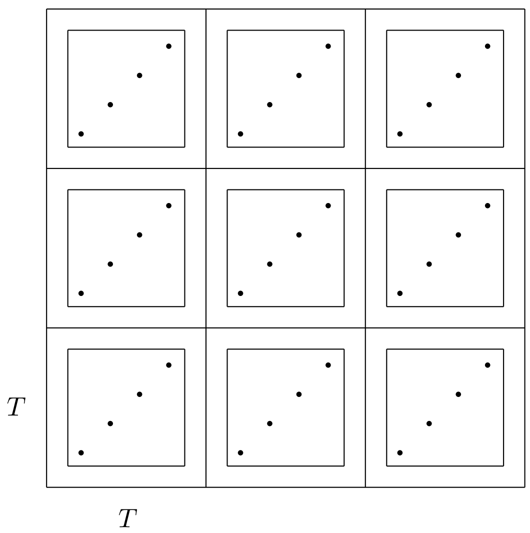

Figure 1: The centers and central areas

The centers in neighboring sub-squares are connected by paths that are edge-disjoint except close to the endpoints. Let us describe how to connect a given center to a center in the sub-square on top. As there are $T/4 = \Delta^2$ rows between the two central areas, for each pair of centers (the jth center, c_j in the bottom sub-square and ith center c'_i in the top sub-square) we can designate a unique row, r_{ij} in this middle area.

To connect c_j to c'_i we first go i steps to the left and then straight up to the designated row r_{ij}. This is completed by starting at c'_i and then going j steps to the right and down to the designated row. We finally use the appropriate segment from the designated row to complete the path (which might be in either direction). A rough picture of this is given in Figure 2. We index the centers from 1 to Δ and hence each path consists of 5 non-empty segments. The first and last segments are totally within the central area while the middle segment is totally in the area between the central areas. Segments two and four go from the central areas to the area in-between.

[2] For simplicity we assume that some arithmetical expressions that are supposed to be integers are in fact exact as integers. By a careful choice of parameters this can be achieved but we leave this detail to the reader.

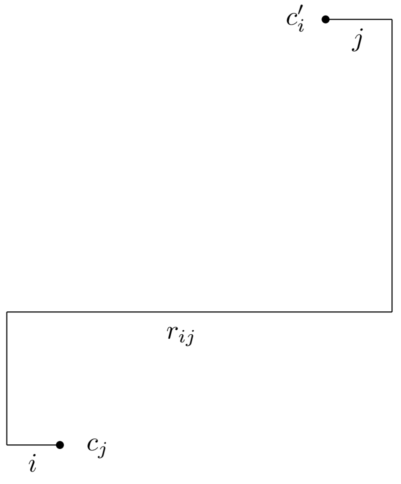

Figure 2: A path

Connecting c_j to a center c'_i in a sub-square to the left is done in an analogous way. There is a unique column c_{ij} reserved for the pair and the path again consists of five non-empty segments. The first and last segments consist of i vertical edges up from c_j, and j vertical edges down from c'_i. We add horizontal segments connecting to the designated column c_{ij} the middle segment is along this column. We state a formal property of these paths.

Lemma 4.1 *The described paths are edge-disjoint except for the at most Δ edges closest to an endpoint. For each edge e, if there is more than one path containing e, these paths all have the same endpoint closest to e.*

Proof: We start by checking the disjointness property. Let us first consider a horizontal edge inside the central area. If it is on the same row as a center then it can only be as the first or last part of a path connecting two centers in two sub-squares on top of each other. As the length of these segments are at most Δ and the horizontal separation between centers is 3Δ these segments originating at different centers do not overlap.

A horizontal edge not on the rows of a center can only appear on the second and fourth segments of a path connecting two centers which are sideways of each other. As the length of the first segment of these paths is at most Δ, the center to which it connects is unique and the vertical distance to the row of this center uniquely identifies the other endpoint.

The above argument continues to hold for horizontal edges in the area between two central areas which are sideways of each other. For vertical edges in the same area each column uniquely identifies the two endpoints by definition.

In the area outside the central area but between two central areas, one of the top of the other, the situation is symmetric. The case of vertical edges in the central area is also analogous to the case of horizontal edges.

Finally in the area outside the central areas and outside the just described parts, i.e. close to the corners of the $T \times T$ squares, there are no paths.

Thus the paths are edge-disjoint except the first and last segments close to the endpoints. ∎

The edges on the three middle segments on a path determine both endpoints of the path. Using this would slightly improve some constants but for simplicity we do not. We let the term *closest endpoint* of an edge denote the closest endpoint of its path if it is in the first or last segment. For the other segments we could chose either endpoint and we can take the literally closest endpoint breaking ties in an arbitrary way. The key property we need is that the "closest endpoint" of a path through an edge is uniquely defined by the edge.

We define the *direction* of a path to be the relative positions of the sub-squares of its two endpoints. It is true that the paths are undirected but at times when we consider paths from a fixed center v it is convenient to think of such paths as starting at v and thus speak of paths going left or right from v rather than sideways. We note that apart from having the same closest endpoint, all paths through one fixed edge e have the same direction.

A restriction is defined by first picking one center in each $T \times T$ sub-square and then the paths described above connecting these centers. Note that these paths are edge-disjoint (and also vertex-disjoint except at the endpoints, but this is more complicated to see and not important). The picked centers naturally form a $n/T \times n/T$ grid if we interpret the paths between the chosen centers as edges. We proceed to make the correspondence more complete by assigning values to variables.

Pick a random solution to the Tseitin formula with charges 0 at the chosen centers and 1 at other nodes. As the number of chosen centers is odd, by Lemma 2.1, there are many such solutions. For variables not on the chosen paths these are the final values while for variables on the chosen paths we call them *suggested* values.

For each path P between two chosen centers we have a new variable x_P and for each variable x_e on P it is replaced by x_P if the suggested value of x_e is 0 and otherwise it is replaced by $\bar{x}_P$.

We claim that with these substitutions we have reduced the Tseitin problem on an $n \times n$ grid to the same problem on an $n/T \times n/T$ grid. This is true in the sense that we have an induced grid when we interpret paths as new edges and we need to see what happens to the axioms.

Given a formula F we can apply a restriction σ to it in the natural way resulting in a formula denoted by $F\lceil_{\sigma}$. Variables given constant values are replaced by constants while surviving variables are replaced by the appropriate negation of the corresponding path variable. A restriction has a natural effect on the Tseitin contradiction as follows.

- The axioms for nodes not on a chosen paths are all reduced to true as all variables occurring in them are fixed in such a way that the axioms are true.

- The axioms for interior nodes of a chosen path are reduced to tautologies as the axiom is true independent of the value of the involved variable(s) x_P. This is true as flipping a single x_P changes the value of two variables next to any such node.

- The axioms at the chosen centers turn into the axioms of the smaller instance.

These just defined restrictions are called *full restrictions* as they completely reduce a full size problem to a smaller problem. A typical full restriction is denoted by σ. Note that these full restrictions are really "affine restrictions" in the vocabulary of [15] as they do not only assign values to variables but also identify several old variables with the same new variable which might also be negated. For simplicity, however, we keep the simpler term "restrictions".

We construct a full restriction by first making a partial restriction and we turn to defining these next.

4.2 Partial restrictions and pairings

A typical partial restriction is called ρ and as we mostly discuss partial restrictions we simply call them "restrictions" while we use the term "full restrictions" when that is what we have in mind. At the same time as describing partial restrictions we give a probability distribution on such restrictions.

Let k be an odd integer of the form $Cs(n/T)^2$ for a constant C to be determined, where s an upper bound on the depth of the decision tree we are analyzing. The first step of constructing ρ is picking k centers uniformly at random from the set of all $\Delta(n/T)^2$ centers defined in the previous section. These are the *alive* centers. In the future we only consider the case when the number of live centers in each sub-square is between a factor .99 and 1.01 of its expected value Cs. The probability of this being false is $O(n^2 e^{-\Omega(s)})$ and this is simply added to other failure probabilities. We are careful to make sure that $s = \omega(\log n)$.

We define charges that are 0 for all live centers and 1 for dead centers. As the number of live centers is odd we can apply Lemma 2.1 and pick a random solution with these charges to the Tseitin formula. For edges not on paths between live centers these are final values while for variables on such paths we call them *preferred* values.

The choice of the centers together with the fixed and preferred variables is denoted by ρ. The choice of ρ is the main probabilistic event. Note that by Lemma 2.1 the number of possible values for fixed and preferred values is independent of which centers are alive and even of k as long as it is odd.

A partial restriction ρ is, for the analysis, preferable to a full restriction σ as it behaves much more independently. A drawback is, however, that as soon as a live center v is discovered then we have many paths leaving v in ρ and this could result in a deep decision tree if they all corresponded to live variables. In order to avoid this we add a second step, a pairing π, turning a partial restriction into a full restriction.

Choose one center to survive in each sub-square. These are called the *chosen centers* and paths between such centers correspond to the variables that remain and are called *chosen paths.* Centers that were alive through the first part of the process but are not chosen are called *non-chosen.* The centers killed already by ρ are simply called dead.

The simplest way to eliminate the non-chosen centers would be if we were able to pair them up in such a way that the two centers in a pair are in adjacent sub-squares and hence connected by a path. In such a case we could negate the preferred values along this path and after this make the preferred values permanent outside the chosen paths. Note that this makes sure that the parity condition at these non-chosen centers are now satisfied. For variables on the chosen paths we turn the preferred values into suggested values completing the full restriction.

Such a pairing might exist with high probability but, as we do not know how to prove this fact, we allow a more general way of eliminating non-chosen centers. We still call this object a pairing as it is not too far from the truth and gives the right intuition.

Definition 4.2 *A* pairing π *is a graph supported on the non-chosen centers. Each component of* π *is either a single edge or a star of size four with one center and three nodes of degree one. Connected centers are located in adjacent sub-squares.*

Before we study pairings let us establish some notation. As the original grid is also a graph with edges we from now on use the term "grid-edges" to refer to edges in the original grid. The chosen centers form a smaller grid and this also has edges and we call these "new grid-edges". We only consider paths in the original grid and we keep the shorter term "path" for these. Thus from now on an "edge" is a connection between two live centers and corresponds to a path in the grid-graph. A "new grid-edge" corresponds to a chosen path and is thus also an edge in the graph of the live centers.

Some edges are conflicting in that we do not allow them to be present in the graph at the same time. More precisely we allow at most one path in each of the four directions from a center. As picking a path corresponds to changing the variables on this path this is the same as saying that the variables can only change at most once.

Lemma 4.3 *If each sub-square has between* $.99Cs$ *and* $1.01Cs$ *non-chosen centers, a pairing* π *exists.*

Proof: For each pair of neighboring sub-squares we want to determine the number of edges of π to go between these two sub-squares. Let m be the smallest integer greater or equal to $.26Cs$, then the number of edges between any two neighboring sub-squares will be either m or number or $m + 1$. As each non-chosen center is of odd degree in π the parity of the number of edges leaving a fixed sub-square is determined and we need to take this into account. We do this by finding a solution to a solvable Tseitin instance.

For each pair of neighboring sub-squares introduce a variables y_e (these correspond to new grid-edges) and make the constraint that the four such variables leading into a sub-square sum modulo 2 to the parity of the number of non-chosen centers in this sub-square. As the total number of non-chosen centers is even (both k and the number of chosen centers are odd) this is a solvable instance. Take any solution and fix the number of paths between two sub-squares corresponding to new grid-edge e to be $m + y_e$.

Consider any sub-square. Suppose that the number of non-chosen centers in it is a. By the just determined variables we know that we should have $1.04Cs + \delta$ edges leaving the sub-square for the some $\delta \in [0, 7]$. This fixes the number of degree three centers in that sub-square to $(1.04Cs + \delta - a)/2$ and by the construction of the numbers y_e this is an integer and by the assumption $a \in [.99Cs, 1.01Cs]$ it is positive and bounded by $.025Cs + 3$. Choose this number of centers to be of degree 3 and connect these to centers in adjacent sub-squares, making sure to connect each center only once. Once this is done we can pair up the remaining centers respecting the number of edges between any two sub-squares. ∎

We could have a probability distribution on π but this does not seem natural and in fact we work with any π. This choice does not matter greatly and this can be seen as follows. In the end when analyzing the process of creating a decision tree we only use a very local piece of π. In particular when looking for a decision tree of depth s we only analyze what happens to $O(s)$ centers in π. There are only $s^{O(s)}$ alternatives for these centers and factors of this size change very little in our argument.

As stated above π makes it possible to turn ρ into σ. Variables not on live paths take their fixed values. Variables on live paths but not on chosen paths take their preferred values unless they are on a path chosen by π in which case these values are negated. On the chosen paths, the preferred values now becomes suggested and this completes the description of σ.

We use the term "preferred values" as a vast majority of the variables will eventually be fixed to these values as very few variables appear on the paths of π. On the other hand "suggested values" are much less certain as the path variables should be thought of as equally likely to be 0 and 1 and thus these variables are equally likely to take also the non-suggested value.

As an intermediate between ρ and the full restriction σ we have ρ and some information in the form of existence or non-existence of edges. We have the following definition.

Definition 4.4 *A* piece of information *is either in form of an edge (v, w) for two centers v and w or $(v, \delta, \perp)$ where v is a center and δ is a direction (i.e. "left", "right" "up" or "down"). The former says that there is an edge from v to w while the latter says that there is no edge from v in the direction δ.*

We note that, as edges are undirected (v, w) and (w, v) denote the same information. In some situations we are, however, interest in the information starting a v in all four directions and then it useful to use the notation with v in the first component. We use sets of information pieces.

Definition 4.5 *An* information set*, I, is a collection of information pieces. Its* support *is the set of centers mentioned in these pieces. An information set is* consistent *if it does not have two different pieces of information from the same center in one fixed direction. Furthermore, if I has information in all four directions from a center v then it has an odd number of edges touching v.*

Note that here, as opposed to the grid, we do not have a problem of small connected components in the complement of a set of centers. This follows as we only consider information sets of size $O(s)$ and a center has a potential edge to all centers in neighboring sub-squares. Jointly with ρ an information set fixes the values of some more variables as follows.

Definition 4.6 *Let ρ be a restriction and I an information set. A variable x_e is considered* forced by (ρ, I) *iff either its closest endpoint, v, is not live in ρ or if the information of v in the direction of e is contained in I. It is forced to its preferred value unless the relevant information piece states that there is an edge from v in the direction of e that corresponds to a path that passes through e in which case it takes the opposite value.*

There are other situations where the value of a variable might be determined by ρ and I, such as the lack, or scarcity, of live centers in a sub-square but we do not allow the algorithm to use this information. We need the notion of a closed information set.

Definition 4.7 *An information set I is* closed *if it is supported on a set X of centers such that for any $v \in X$ the set I contains the information in all four directions.*

The definition implies that for any $v \in X$, in any direction δ where there is not an element of X, we have a non-edge $(v, \delta, \perp)$. When considered as a graph such an information set is an odd-degree graph (with degrees one and three) on the centers of X.

Note that if we have a closed information set I then if we consider all variables forced by (ρ, I) this can be described by a restriction where the centers in the support of I are killed. We simply negate the values of any preferred variable on any path in I and then forget that the centers in the support of I were alive.

Thus, if we let such a closed information set operate on a restriction ρ we get a restriction with fewer live centers where the number of killed centers is exactly the number of centers in the support of the corresponding graph.

5 Decision trees

We have decision trees where each internal node is marked with a variable and the outgoing edges are marked with 0 and 1. The leaves of a decision tree are labeled by 0 and 1. We allow decision tree of depth 0 which are constants 0 or 1.

All decision trees considered in this paper have a depth that is smaller than the dimension of the grid we are currently considering. For each branch in a decision tree there is partial assignment that forces an input to follow this branch. As the branch is short we call it *consistent* if the corresponding assignment is consistent in the sense of Definition 3.1.

We trim decision trees to maintain the property that all branches of a decision tree are consistent. When a decision tree is created this is not a problem but trimming takes place when we consider what happens under a partial assignment τ or a full restriction σ. In that latter case, the initial decision tree uses the variables x_e while the resulting decision tree uses the new variables x_P. Let us first consider the case of a partial assignment τ.

In each situation when walking down the tree if we encounter a variable such that one of its values would make the branch, jointly with τ, inconsistent we simply erase the sub-tree of the inconsistent value and remove the query for the variable whose value is forced by consistency. That at least one branch is consistent follows from Lemma 3.2.

When considering an full restriction σ the situation is very similar. Many variables are fixed and variables alive are identified with the new variables x_P. The first time such a variable x_e identified with a particular variable x_P is queried this results in a query (unless it is fixed by consistency) while for later variables $x_{e'}$ on the same path P their values are dictated by the found value for x_p and whether $x_{e'}$ is identified with x_P or $\bar{x}_P$ which in its turn depends on the suggested value for $x_{e'}$.

A more static way is to consider all branches of T from the root to a leaf and see which of the corresponding assignments are consistent with σ (or τ). The consistent branches remain and the not consistent branches are erased. It is easy to see that the remaining branches (possibly after some contractions) nicely fit into a decision tree and in fact the decision tree we just defined above by the dynamic process.

If the depth of a decision tree is greater than the size of the remaining grid after σ we could be in a situation that no branch of the tree is consistent with σ. We make sure this does not happen by only considering shallow trees.

We let a *1-tree* be a decision tree where all leaves are labeled 1 and define a *0-tree* analogously. Special cases of such trees are trees of depth 0. Next we turn to a procedure of representing formulas by decision trees of small depth.

6 t-evaluations

We have a supposed proof and we have the set of formulas that appear in the proof. We also have each sub-formula in each of these formulas and this gives a set of formulas Γ. We consider t-evaluations φ, as defined by [18], that map formulas to decision trees of depth at most t. Such mappings will not be total and we are interested in finding t-evaluations defined over as large set of formulas as possible. This is made possible by, at the same time as extending the domain, applying a restriction. Let us define the desired properties required of t-evaluations.

1. The constant true is represented by a constant 1 and the constant 0 is represented by a constant 0.

2. If F is an axiom of the Tseitin contradiction then $\varphi(F)$ is a 1-tree.

3. If $\varphi(F) = T$ then $\varphi(\neg F)$ is a decision tree with the same topology as T but where the value at each leaf is negated.

4. Suppose $F = \vee F_i$. Consider a leaf in $\varphi(F)$ and the assignment, τ leading to this leaf. If the leaf is labeled 0 then for each i $\varphi(F_i)\lceil_\tau$ is a 0-tree and if the leaf is labeled 1 then for some i, $\varphi(F_i)\lceil_\tau$ is a 1-tree.

The intuitive role of $\varphi(F)$ is that it represents the formula F as a function on all assignments that satisfy[3] "the relevant" local Tseitin constraints. As F might depend on all variables this does not make complete sense, but for F that depends on few variables this intuitive notion is literally true. For large formulas the correspondence is not as direct and for $F = \vee F_i$ the representation might depend on the order of the sub-formulas F_i.

As an example let us explicitly give the representation of an axiom and take $(x_{e_1} \vee x_{e_2} \vee x_{e_3} \vee x_{e_4})$ where e_i are the four grid-edges incident to a center v. Naturally each variable is represented by a decision tree of depth one. This clause is represented by a decision tree of depth three with all leaves labeled 1 querying the variables x_{e_1}, x_{e_2}, and x_{e_3} in order. The only leaf that requires a little bit of thought to see that it is labeled 1 is the node where all three variables are zero. In this leaf, x_{e_4} is reduced to a decision tree of depth 0 with label 1 as the only value of x_{e_4} consistent the three 0s is 1.

Note that we cannot represent this formula by a smaller tree as, by rule 4, for each 1-leaf, we must have an assignment that forces one of the decision trees for x_{e_i} to be a 1-tree.

As another example consider the conjunction of all the axioms. As we do not have any $\wedge$-gates, this is represented as the negation of the disjunction of the negations of all axioms. As we just saw, each axiom is represented by a 1-tree of depth 3 and hence its negation is a 0-tree of the same depth. Any disjunction of such trees can be represented by a decision tree of depth zero where the only leaf has label 0 and hence the representation of the negation of such a disjunction is a tree of depth 0 with label 1.

Thus we have constant one as a representation for a formula that, when interpreted in the natural way, evaluates to false on each input. The reason is that each sub-formula looks true in the local sense and the conjunction of any number of sub-formulas that look true is considered true.

For a general set of formulas we cannot hope to have a t-evaluation for a small t and our plan is to proceed as follows for $i = 0, 1, 2 \ldots d$.

[3] This is achieved since we only consider branches in decision trees which are consistent.

- We have a t-evaluation for all formulas of Γ that were originally of depth i.
- Pick a random full restriction σ_i and extend the t-evaluation to all formulas of $\Gamma\lceil_{\sigma_i}$ of original depth at most $i+1$.

At the starting point, $i = 0$, each formula is a literal which is represented by a natural decision tree of depth 1 and we start by proving that t-evaluations are compatible with restrictions.

Lemma 6.1 *Given a set of formulas Γ' and a t-evaluation φ whose domain includes Γ' and let σ be full restriction whose output is a grid of size n. Then, provided that $t < n$, $\varphi(F)\lceil_\sigma$ is a t-evaluation whose domain includes $\Gamma'\lceil_\sigma$.*

Proof: This is an easy consequence of the definitions but let us go over the various possibilities. Hitting a decision tree with a full restriction can never increase the depth of the decision tree and hence all representations are decision trees of depth at most t. Note also that as $t < n$ some branch of the decision tree is consistent with σ. We need to check the properties of a t-evaluation.

The first and second properties are obvious as a restriction does not change the fact that something is 1-tree or a 0-tree.

The third property is also rather obvious. The decision trees for F and $\neg F$ are effected the same way and there is nothing that can change that the corresponding leaves have labels that are the negations of each other.

For the fourth property consider any branch in T that appears in $T\lceil_\sigma$ and the corresponding assignment τ which, by definition of $T\lceil_\sigma$, is consistent with σ. As already τ reduces the T_i in a good way, we need only observe that $T_i\lceil_\sigma\lceil_\tau$ is a non-empty decision tree and hence it is a 1-tree or a 0-tree as desired. ■

Now we eventually come to the key lemma of the entire argument.

Lemma 6.2 *Let s' be an integer and $s = \max(s', t)$, then there is a constant A such that the following holds. Suppose there is a t-evaluation that includes $F_i, 1 \le i \le m$ in its domain and let $F = \vee_{i=1}^m F_i$. Let σ be a random full restriction from the space of restrictions defined in Section 4. Then the probability that $F\lceil_\sigma$ cannot be represented by a decision tree of depth at most s' is at most*

$$(As^{27}t\Delta^{-1})^{s'/108}.$$

We postpone the proof of this lemma to Section 7 and see how to use it. We apply it with $s' = t = s = \frac{1}{2}n^{1/(58(d+1))}$ and $\Delta = s^{29}$ (and hence $T = 4s^{58}$) and let us fix these values.

We start with the original Tseitin contradiction on the $n \times n$ grid. Let $n_i = nT^{-i}$. We are going to choose a sequence of full restrictions σ_i mapping a grid of size n_i to a grid of size n_{i+1} randomly. Let σ_i^* be the composition of σ_0, σ_1, $\ldots \sigma_i$. As stated above, Γ is the set of sub-formulas that appear in an alleged proof and we let

$$\Gamma_i = \{F\lceil_{\sigma_i^*} \mid F \in \Gamma \wedge depth(F) \le i\}.$$

Let f_i be the number of sub-formulas of depth at most i in Γ.

Lemma 6.3 *With probability* $1 - f_i(s/A)^{-s/108}$ *there is a* t*-evaluation* φ_i *whose domain includes* Γ_i.

Proof: This is essentially collecting the pieces. We prove the lemma by induction over i. For $i = 0$ we have the t-evaluation that maps each literal to its natural decision tree of depth 1.

When going from depth i to depth $i+1$ we need to define φ_{i+1} on all formulas originally of depth at most $i + 1$ and consider any such F.

1. For each F of depth i it is, by induction, in the domain of φ_i and we can appeal to Lemma 6.1.
2. If F is of depth i then $\varphi_{i+1}(\neg F)$ is defined from $\varphi_{i+1}(F)$ negating the labels at the leaves.
3. For $F = \vee F_i$ where each F_i is of depth i we apply Lemma 6.2.

The only place where the extension might fail is under step three but, by Lemma 6.2, the probability of failure for any individual formula is at most $(s/A)^{-s/108}$ and we have at most $f_i - f_{i-1}$ formulas the induction is complete. ■

As a final piece we establish that all formulas appearing in a short proof must be represented by 1-trees and as the constant false is represented by a 0-tree we cannot derive the desired contradiction in a short proof. In order to prove this we must go over the derivation rules of our proof system. The details are not important and we choose the same rules as [12] and these are as follows.

- (Excluded middle) $(p \vee \neg p)$
- (Expansion rule) $p \to (p \vee q)$
- (Contraction rule) $(p \vee p) \to p$
- (Association rule) $p \vee (q \vee r) \to (p \vee q) \vee r$
- (Cut rule) $p \vee q, \neg p \vee r \to q \vee r$.

Lemma 6.4 *Suppose we have derivation using the above rules and using the Tseitin conditions in the* $n \times n$ *grid as axioms. Let* Γ *be the set of formulas appearing as sub-formulas of any formula in the given derivation and suppose that we have a* t*-evaluation whose domain includes* Γ *where* $t \leq n/3$*. Then each line in the derivation is mapped to a 1-tree. In particular we do not reach a contradiction.*

Proof: We prove this by induction over the number of lines in the derivation. We constantly make use of the fact that $t \leq n/3$ to conclude that for any decision tree, T, in the domain of the t-evaluation and any assignment τ to at most $2t$ variables we have that $T\lceil_\tau$ is still a non-empty decision tree. By assumption each axiom is represented by a 1-tree and we consider the derivation rules.

Let us first look at excluded middle $F = p \vee \neg p$. Take any leaf in $\varphi(F)$ and let τ be the assignment leading to this leaf. As p and $\neg p$ are represented by trees that only differ in that the labels at the leaves are negated they cannot both be reduced to 0-trees by τ and hence we conclude that the label of the leaf in $\varphi(F)$ must be 1.

For the expansion rule let $F = p \vee q$. Take any leaf in $\varphi(F)$ and let τ be the assignment leading to this leaf. If this leaf has label 0 then, by definition, $\varphi(p)\lceil_\tau$ must be a 0-tree but this contradicts that $\varphi(p)$ is a 1-tree.

Now consider the contraction rule and $F = p$. Take any leaf in $\varphi(F)$ and let τ be the assignment leading to this leaf. If this leaf has label 0 then consider $\varphi(p \vee p)\lceil_\tau$ and take any branch τ_1 in this tree consistent with τ. As $\varphi(p \vee p)$ is a 1-tree this must lead to a label 1 but this contradicts the definitions as both sub-formulas (p and p) cannot be reduced to 1-trees under τ_1 as τ_1 is consistent with τ and $\varphi(p)\lceil_\tau$ is a 0-tree.

The association rule is more or less obvious as our requirements for t-evaluation do not really distinguish the two formulas. On the other hand the two formulas may have different t-evaluations so let us do also this case. We have $F = (p \vee q) \vee r$ and take a supposed leaf with label 0 in $\varphi(F)$ and let τ be the assignment leading to this leaf. By definition, $\varphi(r)\lceil_\tau$ as well $\varphi(p \vee q)\lceil_\tau$ are 0-trees. From the latter statement we conclude that also $\varphi(p)\lceil_\tau$ and $\varphi(q)\lceil_\tau$ are 0-trees. Let us consider $\varphi(p \vee (q \vee r))\lceil_\tau$. There is some branch τ_1 in this tree that is consistent with τ and this leads to a leaf with a label 1 as this is a 1-tree. One of the three sub-formulas is reduced to a 1-tree at this leaf and we reach the usual contradiction.

Let us finally look the cut rule. We have $F = (q \vee r)$ and let us take a supposed leaf with label 0 in $\varphi(F)$ and let τ be the assignment leading to this leaf. We know that $\varphi(q)\lceil_\tau$ and $\varphi(r)\lceil_\tau$ are both 0-trees. Consider any branch in $\varphi(p)\lceil_\tau$ and let τ_1 be the assignment of this branch. Assume this leaf is labeled 0, the other case being similar. Now take any branch in $\varphi(p \vee q)\lceil_{\tau\tau_1}$. As this is a 1-tree the label at this branch must be 1. This contradicts that $\varphi(p)\lceil_{\tau_1}$ as well as $\varphi(q)\lceil_\tau$ are both 0-trees. This concludes the case analysis. ∎

Fixing parameters we get the main theorem of this paper.

Theorem 6.5 *Suppose that $d \leq \frac{\log n}{59 \log \log n}$, then, for sufficiently large n, any depth-d Frege refutation of the Tseitin contradiction on the $n \times n$ grid requires size $exp(\Omega(n^{1/58(d+1)}))$.*

Proof: Suppose we have a refutation of size S and consider the corresponding set of sub-formulas Γ. Remember that $s' = t = s = \frac{1}{2} n^{1/(58(d+1))}$ and $\Delta = s^{29}$.

With the given choice of Δ we have $T \leq n^{1/(d+1)}$ and we have a $nT^{-d} \geq T$ sized grid remaining after σ_d^*. The probability that we fail to have a t-evaluation that includes all formulas of Γ in its domain after σ_d^* is, by Lemma 6.2 bounded by $S(s/A)^{-s/108}$. The probability that we at any stage of the process we do not have between $.99Cs$ and $1.01Cs$ alive centers in a sub-square is bounded by $n^2 e^{-\Omega(s)}$. As $s = \omega(\log n)$, the sum of these two failure probabilities, for sufficiently large n, is smaller than 1 there exists a σ_d^* which makes all subformulas in the proof have a t-evaluation and such that the final restriction gives a grid of size at least T. As $t = o(T)$ we can appeal to Lemma 6.4 and the proof is complete. ■

We have an immediate corollary.

Corollary 6.6 *Any polynomial size Frege refutation of the Tseitin contradiction requires formulas of depth* $\Omega(\frac{\log n}{\log \log n})$.

Finally we turn to the proof of the switching lemma which is the heart of the argument.

7 Proof of the switching lemma

Remember that we have $F = \vee F_i$ and we have a t-evaluation φ that includes each F_i in its domain and let $T_i = \varphi(F_i)$. We create an *extended canonical* decision tree for $F\lceil_\sigma$ by going over the trees T_i one by one. If there is a branch in T_i that leads to a leaf with label 1 that is consistent with the information we have so far we explore the variables of this branch (and some extra variables). Let us proceed.

It is important that the constructed decision tree does not depend on the preferred values along the chosen paths but we may, and indeed we will, let it depend on other parameters and in particular we make use of the knowledge of the identity of the chosen centers and non-chosen centers.

As we go over the T_i's we have a set of centers, S, that will be called *exposed centers* and an information set I that, jointly with ρ, guides the construction of the decision tree. Both S and I start out empty.

For non-chosen centers in S, the set I contains the information pieces corresponding to their component in π and if one center in such a connected component belongs to S then so does the entire component. For chosen centers in S we have, in the decision tree, queried all variables x_P adjacent to these centers and this information is present as information pieces in I. The one-answers are recorded in the form of a path while the zero answers as two non-edges. A typical set of answers given by the decision tree is denoted by τ. These are answers in a decision tree querying new variables x_P. Note that the value of x_P jointly with ρ determines the value of all x_e on the chosen path P.

We go over the decision trees one by one and let us see what happens when we consider T_i. Take the first (in some fixed order) branch in T_i that leads to a leaf labeled 1 (if no such branch exists we move to T_{i+1}). For the variables appearing on this branch we have unique values required to reach this leaf. We let a *forcing information*, J, be an information set that, jointly with I, forces[4] all variables on this branch, from now on called "the forceable branch" to take these unique values. We require the following properties of J.

[4] Please remember, by Definition 4.6 for a variable to be forced we need to know the relevant information at its closest endpoint.

1. If J contains a non-edge from a chosen center it also contains a non-edge in the "reverse direction". As an example if it contains a non-edge going left from a chosen center v then it contains a non-edge going right from the chosen center in the sub-square to the left of v.

2. Neither I nor J contains a path between a chosen center and a non-chosen center.

3. The information sets I and J are consistent and disjoint.

Even given these requirements we might have many different J forcing the same path. Any such forcing information works equally well and any rule for making this rather arbitrary choice is equally good for us.

At any point when forming the extended canonical decision tree, the information I comes from information in π and from queries already done in the decision tree with answers τ. Let us first see that the lack of forcing information implies that T_i is in fact reduced to a 0-tree.

Lemma 7.1 *If there is no forcing information for T_i then $T_i\lceil_{\sigma\tau}$ is a 0-tree.*

Proof: Suppose indeed that there is a branch in T_i that leads to a 1-leaf and is consistent with σ and τ. This implies that we can extend τ to τ_1 such that we reach this leaf. In other words, σ and τ_1 jointly determine a value to each variable on this branch and for any variable x_e on this branch, not already fixed by ρ we have the information of its closest endpoint in its direction either from π or, if its closest endpoint is chosen, by τ_1.

We proceed to construct some forcing information J. Let us consider a variable x_e on the branch. For e whose closest endpoint is not chosen we include the information from π on this closest endpoint in direction of e. If the closest endpoint of e is chosen then it may or may not be on the chosen path in its direction.

If e is on the chosen path then the information τ_1 must contain the value of the corresponding path-variable and we include that information in the form of an edge or two non-edges in J. If e is not on the chosen path then we choose some value to the path-variable in its direction from its closest endpoint that is consistent with τ_1 and choices for previous variable set in the current process. Given the value of this variable we include this in the information set J either as an edge or two non-edges.

This constructed information set J clearly forces the values of the variables on the branch to the values needed to follow the branch and we need to check that it is an allowed information set. The first property is true by construction.

As π only contains paths between two non-chosen centers and τ_1 and its extension only paths between two chosen centers, we cannot have a path between a chosen and non-chosen center in J and we need to check consistency with I.

On the non-chosen centers, I contains some information from π and as the information in J on the non-chosen part is also from π this is consistent (clearly any duplicated information can simply be dropped from J).

On the chosen centers we know that τ_1 is an extension of τ, the information obtained in the decision tree up to this point. As the information in I on the chosen centers is exactly given by τ and the information in J which is from τ_1 is consistent with τ we conclude that J is consistent with I.

We conclude that the constructed J is an allowable forcing information. This is a contradiction to the assumption of the lemma and we conclude is that the assumed 1-branch in T_i does not exist. ■

Given a forcing information set J we continue the construction of the decision tree as follows. We expose all centers in the support of J but also some additional centers as follows.

- For any non-chosen center v in the support of J we expose the centers in its connected component in π.
- We let the chosen centers in the support of J be the nodes supplied by the adversary in the matching game described in Section 3 played on the grid given by the chosen nodes. We apply Lemma 3.3 and expose also the partners of these nodes in the matching provided by P_M.

We note that if the support of the forcing set J is of size r then the number of exposed centers is at most $4r$ as we expose at most 3 more centers for any non-chosen center and at most one extra center for any chosen center.

We now extend the information I by including the connected component from π of the non-chosen exposed centers. For the chosen centers we query all variables adjacent to any exposed center. We record one-answers as an edge in I and zero-answers as two non-edges including the other endpoint of a potential chosen path, i.e. the chosen center in the adjacent sub-square in the given direction.

Given this extended I it is possible to tell whether the forceable branch in T_i is traversed. This follows as for any variable on the branch the closest endpoint is now exposed and for each exposed center we have information pieces in all four directions. If this branch is indeed followed, the process is ended as $T_i\lceil_{\sigma\tau}$ is a 1-tree and the branch of the decision tree can be terminated with label 1.

If the forceable branch is not followed we continue the process by first looking at T_i under this new extended information I and searching for some new forcing information of a different 1-branch and then looking at $T_{i'}$ for $i' > i$.

Finally, if all T_i's have been processed we terminate the branch in the decision tree and label the leaf 0. This ends the description of the creation of the extended canonical decision tree for $F\lceil_{\sigma}$. We observe that we have created a decision tree that is a legitimate choice for $\varphi(F)$. Indeed, at any leaf labeled 1 we have found a T_i that is reduced to a 1-tree and if all T_i have been processed then, by Lemma 7.1, this leaf in the decision tree is correctly labeled 0.

Note that this process depends on ρ and π but not, in a serious way, on the negations of the preferred values along the paths between the chosen centers. As we have no paths between chosen and non-chosen centers the only difference is that variables on chosen paths in one case are forced by the path and in the other case by two non-edges and this does not cause any difference as the supports are identical. As this is of key importance let us record this as a lemma.

Lemma 7.2 *Let σ_1 be obtained from ρ_1 and π and σ_2 from ρ_2 and π where ρ_1 and ρ_2 pick the same set of centers and fixed values. Assume furthermore that the only difference between ρ_1 and ρ_2 is that for each chosen path P there is a bit c_P such that for each grid-edge e on P the preferred values of x_e differ by c_P in ρ_1 and ρ_2. Then the only difference between the extended canonical decision trees of $F\lceil_{\sigma_1}$ and $F\lceil_{\sigma_2}$ is the labeling of the internal edges.*

In the decision tree, at round j, we query all variables touching the chosen centers of the set S. We say that the set of answers is *closed* iff the answer to a query is one iff it corresponds to an edge in the dynamic matching created by P_M. This slightly overloading the notion "closed" but note that a closed branch gives rise to a closed information set and hence we feel that using the term "closed" also in this situation gives the correct intuition. The following lemma is now an immediate consequence of Lemma 7.2.

Lemma 7.3 *If the probability that $F\lceil_\sigma$ needs a decision tree of depth s' is at least q, then the probability that the extended canonical decision tree of $F\lceil_\sigma$ contains a closed branch of length at least s' is at least $2^{-s'}q$.*

In view of this lemma we complete the proof by analyzing the probability of such a closed branch. This analysis is done using the labeling technique of Razborov [14]. In other words we take a ρ that contributes to the above event and create a ρ^* which is also a restriction but with fewer live centers. We then establish that given ρ^* and some extra information it is possible to reconstruct ρ. The proof is finished by establishing the fact that there are many fewer ρ^* than ρ and the extra information can be limited in size.

As the overall structure closely follow the proof of Razborov let recall this proof as it is helpful for reference. Razborov has a restriction that keeps exactly k randomly picked variables undetermined and randomly gives values 0 and 1 to the other variables. He creates a canonical decision tree by the process below where the counter j indicates the stage.

1. Set $j = 1$

2. Find the first possible 1-branch of a decision tree, T_{i_j} that can be traversed given the random restriction ρ and the values queried in the decision tree so far. If no such branch exists in any remaining tree answer 0 and halt.

3. Let S_j be the set of undetermined variables on this branch.

4. Let σ_j be the values of the the variables in S_j that force this 1-branch to be traversed.

5. Query the variables in S_j in the decision tree. Record the answers as τ_j. If $\tau_j = \sigma_j$ answer 1 and halt, otherwise set $j = j + 1$ and goto step 2.

The restriction ρ^* is now defined as ρ with the addition that the variables in S_j are given the values given by σ_j. A good picture to keep in mind is the following.

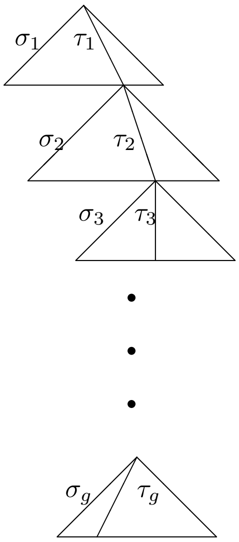

Figure 3: The long path in the decision tree is given by the τ_i following the middle line. In each step there is an assignment σ_i that leads to a 1-leaf.

It is not difficult to see that ρ^* makes the input follow the 1-branch in T_{i_1}. The reconstruction information tells which variable(s) on this branch belong(s) to S_1 and their values in τ_1. It is not difficult to see that this can be done with $(4t)^{|S_1|}$ alternatives. The reason is that once the branch is given the elements in S_1 can be identified by giving their index on the branch.

Given this information the reconstruction algorithm changes that values of the variables in S_1 from σ_1 to τ_1 creating a restriction ρ_1^*. This restriction forces the 1-branch of T_{i_2} and thus it is possible to identify S_2 and τ_2 at a cost $(4t)^{|S_2|}$. We then change the values on S_2 from σ_2 to τ_2 and continue like this until all sets S_j have been identified. Finally ρ is defined as the restriction obtained from ρ^* by changing all elements of $\cup_i S_i$ to undetermined.

If the decision tree needs to query s variables then ρ^* has $k-s$ undermined variables and the information set used by the reconstruction procedure takes at most $(4t)^s$ different values.

There are at most

$$\binom{n}{k-s}2^{n+s-k}$$

possible ρ^* and thus at most

$$(4t)^s \binom{n}{k-s} 2^{n+s-k}$$

different ρ can be reconstructed this way. As there are

$$\binom{n}{k} 2^{n-k}$$

possible ρ the probability that ρ gives a branch of length at most s in the canonical decision tree is at most

$$\frac{(4t)^s \binom{n}{k-s} 2^{n+s-k}}{\binom{n}{k} 2^{n-k}} \approx \left(\frac{8kt}{n}\right)^s$$

and we are done.

We follow the same recipe and the information set J at stage j plays the role of σ_j while discovered information from π and the queries to the decision tree plays the role of τ_j. In Razborov's proof σ_j and τ_j are just different assignments to the same set of variables and thus it is obvious that τ_j is compatible with $\sigma_{j'}$ for $j \neq j'$. This compatibility requires some care in our case. One important step is also to enlarge the given forcing information J to a closed information set. This is useful for at least two reasons. A restriction combined with a closed information set gives values to the same variables as a restriction with fewer live variables. Also closed information sets supported on disjoint set of variables are always consistent. The fact that we are analyzing a closed branch makes also the information set I "almost" closed. The only non-closed part is some non-edges for some chosen but non-exposed centers but these can be handled. After this detour let us return to the main argument and thus we have a ρ giving a closed branch of length at least s' in the extended canonical decision tree and and we proceed to construct ρ^*. We later describe the information needed to invert this mapping.

For technical reason we stop the creation of the extended canonical decision tree once we have exposed at least s' centers and we analyze the probability that we ever reach this point. Suppose this happens after the gth stage, where $g \leq s'$ as we expose at least one center in each stage.

At the end of the process we have a set, S_g, of exposed centers which is of cardinality at least s' and at most $s' + 8t$, as we at each stage expose at most $8t$ centers. This follows as J contains at most $2t$ centers as the length of each branch in T_{i_j} is at most t and we add at most 2 centers for each variable on the branch. We later expose at most three more centers for each element in the support of J.

Let us look at the forcing information in stage j and introduce some notation. The forceable branch appears in T_{i_j} and let J_j be the forcing information set. As we continue processing the same T_i after a stage is completed it might be the case that $T_{i_j} = T_{i_{j+1}}$, but then the forceable branches are different. We want to extend the information set J_j to transform it into a closed set.

Consider any center v in the support J_j. It has information in some of its directions coming from I and J_j. If it has information in all four directions nothing needs to be done. Otherwise, take one direction for which the information is not known. If there are more directions in which there is no information, add a non-edge in any other such direction.

If we already have an odd number of edges next to v we add a non-edge in the final direction and otherwise we add an edge to a fresh center in the suitable sub-square. By a fresh center we mean a non-chosen center that is not an element of S_g and has not been used for an earlier J_j. As we use at most one fresh center for each element in S_g the number of non-fresh centers is at most $2|S_g| \leq 2s' + 16t$. As there are $.99Cs$ non-chosen centers in any sub-square there is always, provided that C is a large enough constant, a fresh center to add. Finally we add non-edges from the fresh center in the other three directions.

When we have processed all centers of J_j we have created a closed graph which extends the information set J_j and which we denote γ_j. This follows as for each even degree center we have added a fresh center that is of degree one. Below we establish that the γ_j have disjoint supports, but let us assume that this is true for the time being. The process is quite similar to the proof of Razborov for the ordinary switching lemma and a picture of it can be seen in Figure 4.

As discussed previously, closed graphs can be used to define restrictions with fewer live centers and we define ρ^* to be the restriction defined by ρ together with the graph $\gamma = \cup_{j=1}^{g} \gamma_j$. This is a standard restriction where all centers in the support of γ are now dead. We call these the *disappearing* centers.

For the curious reader let us point out a subtle point. It is true that any collection of closed information sets with disjoint supports are compatible, but this is only true as long as we forget what centers are chosen as we could have the case that the four chosen neighbors of a chosen center all have a non-edge in its direction. This would not be allowed but once we forget who was chosen there is no problem.

Before we turn to the reconstruction process let us introduce some notation for the information sets of the decision tree process. Let us see what happens at stage j.

On the non-chosen centers there is the information of some connected components of π, namely all the exposed centers and let $I_{j,n}$ denote the union of these components discovered in stage j. For the chosen centers the information is obtained by the decision tree. As the decision tree is closed this is given by a matching on the exposed chosen centers. On top of this we have the information of non-edges of non-exposed chosen centers in the direction of exposed chosen centers. Call this information on the chosen centers $I_{j,c}$ and let I_j be the union of $I_{j,n}$ and $I_{j,c}$. Note in particular that, as all centers in the support of J_j are exposed at round j, all centers in the support of J_j are contained in the closed graph part of I_j. As the only other centers in the support of γ_j are the unique centers added as the final step and we conclude, as claimed above, that the supports of γ_j for different values of j are disjoint.

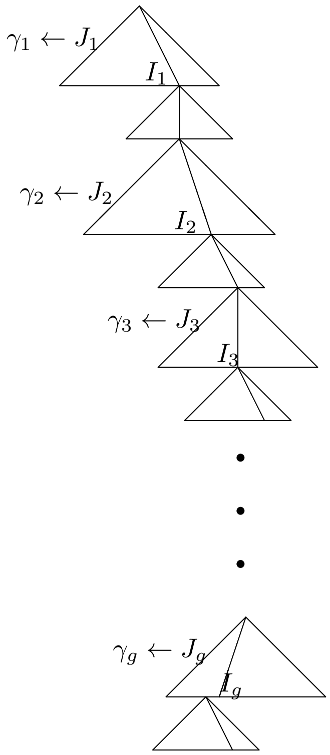
$\gamma_1 \leftarrow J_1$
I_1
$\gamma_2 \leftarrow J_2$
I_2
$\gamma_3 \leftarrow J_3$
I_3
$\gamma_g \leftarrow J_g$
I_g

Figure 4: The long path in the decision tree in our proof. The I_j asks for more information than the 1-forcing information sets J_j and the information comes from π and the answers in the decision tree. The information sets J_j are completed to closed information sets γ_j once the full long path has been found.

We let I_j^* denote $\cup_{i=1}^{j-1} I_i$, the information set gathered during the first $j-1$ rounds. It turns out to be convenient to consider $\cup_{i=j}^{g} \gamma_i$, the graphs added after stage j, and we let γ_j^* denote this graph.

The high level plan is now as follows. As γ_j extends the forcing information J_j we have that $(\rho, I_j^* \cup \gamma_j)$ and hence $(\rho, I_j^* \cup \gamma_j^*)$ forces the input to traverse the jth forceable branch. This branch should enable us to find a good fraction of the elements of γ_j, namely the closest endpoints of all variables on this branch. We then use some external information to find the rest of the elements of γ_j (as well as its graph structure). Finally we then use external information to reconstruct I_j and proceed with stage $j+1$.

As I_1^* is the empty set and $\gamma_1^* = \gamma$ the starting point of the decision process is (ρ, γ) which forces exactly the same variables as ρ^* and thus we know where to start. Although these two objects force the same variables the information content is slightly different in that (ρ, γ) contains the information we are trying to recreate, the identity of the disappeared centers.

We let ρ_j^* be the restriction obtained from applying γ_j^* to ρ and at stage j we will be working with (ρ_j^*, I_j^*) instead of $(\rho, I_j^* \cup \gamma_j^*)$. Again these two objects force the same set of variables but have slightly different information contents.

It is important to identify T_{i_j} and the forceable branch but unfortunately it might not be the first 1-branch traversed by (ρ_j^*, I_j^*). The reason for this is that we might reach a 1-leaf by a branch using variables that would give forcing information that is not allowed. For instance when we make sure that γ_j is closed we add paths between chosen and non-chosen centers and this is not allowed as forcing information. A more subtle problem is that of requiring the other endpoint of non-edges on chosen centers when used as forcing information. It turns out that it is difficult to make sure that the information at the other endpoint is consistent with the rest of the information.

Let I_j^{*-} be the information pieces of I_j^* with any piece supported on γ_j^* removed and let I_j^- be I_j with the same type of pieces taken away. The removed pieces are simple to describe.

Lemma 7.4 *An information piece in I_j^* that is on a center in the support of γ_j^* is in the form of a non-edge from a chosen center in the direction of an exposed chosen center.*

Proof: The information set I_j^* consists of a closed graph jointly with non-edge information on chosen centers of the type allowed in the lemma. Since any information set J_i for $i \geq j$ is disjoint with I_j^* no γ_i with $i \geq j$ can intersect the closed graph part of I_j^*. ■

We get a direct consequence of Lemma 7.4.

Lemma 7.5 *Any variable forced by (ρ, I_j^*) is forced also by (ρ_j^*, I_j^{*-}).*

Proof: The removed pieces of I_j^* are, by Lemma 7.4, on centers that have disappeared in ρ_j^* and hence any variable forced by such a piece is fixed in ρ_j^*. As the piece of information is a non-edge in both I_j^* and γ_j^* it is forced to the same value. ■

As stated above we might have some 1-branch before the forceable branch of stage j. This could, in some vague sense, be good, in that it reveals some element of γ, but as we cannot count on this happening we need to make sure that this is not bad. Thus, we have to be careful to make sure that the reconstruction process is not fooled. Towards this end we introduce the *signature* of any disappearing center, v, as follows.

1. The value of j such that $v \in \gamma_j$. This has at most s possibilities.
2. The information of whether v is a closest endpoint to any variable on the forceable branch and in such a case in which direction(s) it has variables appearing on this branch. This has $O(1)$ possibilities.

On the high level the reconstruction procedure maintains the following information.

1. A counter j of the current stage to be reconstructed. Initially $j = 1$.
2. The restriction ρ_j^*. Initially $\rho_1^* = \rho^*$ and we describe below how to update.
3. The information set I_j^{*-}. Initially this is empty and we describe below how to update.
4. A set E of disappearing centers together with their signatures. Initially E is empty.

In the reconstruction process we need to find the identity of some centers. For intuition let us discuss different contexts where this happens and how much external information is needed. For some disappearing centers we also specify the signature which amounts to $O(s)$ possibilities for each center. We have the following cases.

1. A disappearing center that is the closest endpoint of a variables on a discovered 1-branch. This can be found by giving the distance from the root on the branch at cost t.
2. A disappearing center that is not the closest endpoint of a variable on a branch but we know the sub-square where it is located. This can be specified at cost Δ.
3. A non-disappearing and live center where we know the sub-square. This can be specified at cost $1.01Cs$ as these are the number of live centers in any sub-square.

The two first situations appear when finding centers in γ_j while the last situation appears when finding centers in I_j that are not contained in γ_j^*. Identifying a disappearing center has "profit" (as will be seen in the final calculation of counting the number of ρ^* compared to the number of ρ) of $\Omega(\Delta/s)$ and thus there is a huge profit in the first case and the moderate loss in the second. For the third case there is no associated profit but on other hand only a moderate cost. The key for the final analysis is to bound the number of costly step by a constant times the number of profitable steps of the first kind. Let us now formally define the reconstruction process.

1. Set $j = 1$, $\rho_1^* = \rho^*$ and let I_1^{*-} be the empty set.
2. Find the next 1-branch traversed by the information (ρ_j^*, I_j^{*-}).
3. Locate the closest endpoints of all variables on this branch. If any such center belongs to E and its signature does not match the current branch, go to the next 1-branch. By "not matching" we mean that the stage information is incorrect or that the direction(s) of the edges involved does not exactly match the signature.
4. Read a bit b to determine if there are more disappearing centers to be found as the closest endpoint to variables on this branch.
5. If $b = 1$ read one integer that is at most t to determine a disappearing center that is the closest endpoint of a variable on this branch. Read its signature. If this signature agrees with the current branch repeat step 3 and otherwise include it in E and go to the next branch.
6. If $b = 0$ we have found the forceable branch. We read some external information to determine γ_j and I_j^- (details below). Update ρ_j^* to ρ_{j+1}^* and I_j^{*-} to I_{j+1}^{*-}, drop any disappearing center of stage j from E, $j = j+1$ and repeat from 2.

The are a few details and facts about this reconstruction procedure to sort out. Let us start with establishing that we are indeed correctly identifying the forceable branch.

Lemma 7.6 *If a 1-path is forced by (ρ_j^*, I_j^{*-}) and the signatures of all closest variables on this branch match and it is the first such branch, then this branch is the jth forceable branch.*

Proof: As all variables on the branch are forced we must have the information of their closest endpoints in the correct direction(s). As none of the variables have a closest endpoint of a stage later than j the branch is forced by $(\rho, I_j^{*-} \cup J_j)$ jointly possibly with a non-edge in γ_j contained in I_j^*. This implies that the forcing information J_j is valid for this branch and being the first such branch it must be the jth forceable branch. ■

Let us now see how to reconstruct γ_j. We have already identified all the closest endpoints of variables on the forceable branch and we know, by their signature which directions they need a neighbor. We read the identity of these centers at a cost[5] of at most Δ for each center. This identifies J_j. To finalize

[5] It might be the case that some of these centers were found previously and are part of E or that also the other endpoint is uniquely defined by occurring variable. In either case the cost, including the signature is $O(st)$ which is bounded by Δ.

the description of γ_j we read the identity of the unique fresh centers used to make γ_j closed at a cost of Δ for each such center. Having identified γ_j we turn to I_j^-. We first have a bit for each element in γ_j to indicate whether it is also an element of I_j.

First observe that any center in the support of I_j^- cannot belong to the support of $\gamma_{j'}$ for $j' > j$ and thus any such center is still alive in ρ_j^* and thus can be identified at as cost of at most $1.01Cs$ provided we know the sub-square to which it belongs.

First we reconstruct the non-chosen centers. For each non-chosen center in J_j using $O(1)$ bits we find out the size of the connected component in π and the directions of each edge. Then we identify the other endpoint of each such edge at cost $1.01Cs$.

For the chosen centers we can again discover the graph part with $O(1)$ bits per center for structure and an integer of size $1.01Cs$ for the identity. The non-edges not supported on γ_j^* are also reconstructed at cost $1.01Cs$ for identity and $O(1)$ bits per center for direction.

Finally for any center in γ_j we have 4 bits to describe whether the piece of information in the form of non-edge in any direction(s) should be added in I_{j+1}^{*-}.

This terminates the description of the reconstruction and let us sum up the external information needed. Let a_j be the number of disappearing centers that are discovered through being the closest endpoint of a discovered variable and are part of the jth forceable branch and let b_j the number of additional centers in γ_j. Furthermore let c_j the number of centers needed to be discovered in I_j^- after γ_j was discovered.

Lemma 7.7 *We have $b_j + c_j \leq 25a_j$.*

The fact that there is some constant such that the above lemma is true is fairly obvious but as the constant goes into the exponent of the final result we make a moderate effort to minimize it.

Proof: All centers contributing to b_j and c_j are discovered while processing the jth forceable branch. We start with some centers discovered as closest endpoints and find other centers in γ_j and I_j. Let us see how many centers that can be included based on a single starting point v. Let us first assume that the starting points are at distance at least 7 and begin by looking at the case when v is a chosen center.

Remember that a discovered v is the closest endpoint of a variable on the discovered 1-branch. The information set J_j might contain also the other endpoint(s) of paths starting at v. When forming γ_j we might add additional centers to make it closed. Finally when construction I_j we expose the partners in the matching provided by P_M and then also the neighbor of all chosen exposed centers. There are a number of cases to consider.

The center v might have up to four neighbors in J_j and let us first assume that all four are present. As J_j is consistent, v must have an edge to one of the neighbors but for the other three we might have to add a fresh center as a neighbor to γ_j to make it closed.

In the information set I_j we first expose the partners of v and its neighbors in J_j in the matching provided by P_M. As v needs to be matched to one of it neighbors[6] this is a total of at most 8 centers that can be exposed. The chosen center neighbors of all these centers are members of I_j.

In total v might hence cause us to identify the 4 chosen centers at distance one, the 8 chosen centers at distance 2 and 9 chosen centers at distance 3 (we had at most 3 centers at distance 2 as partners of neighbors and each of these have 3 neighbors not counted before). We also have the 3 fresh centers that might be included in γ_j. This is a total of 24 centers that might be needed to identify all centers of γ_j and I_j. Let us turn to the case when v has information in three directions in J_j.

In this case all the four centers (v and its three neighbors in J_j) might be of degree 0 and need a fresh neighbor when forming γ_j. The set of exposed centers can be the same as in the case of four neighbors of v in J_j. This is true as v might be matched to the missing neighbor. The rest of the argument is the same and thus the difference is that we might have added four centers when forming γ_j as opposed to three, and thus we end up with the bound of 25 added centers in this case.

It is not difficult to see that if v has one or two neighbors in J_j then we add fewer centers. Finally if the starting centers are not well separated then some centers are counted twice and this compensates for some center that becomes degree two and needs a fresh center as a neighbor. We omit the details. We conclude that the estimate holds also in this case. Let us turn to non-chosen centers.

Such a center can only have neighbors in J_j in three directions. This follows as for non-edges at non-chosen centers we do not need the information of the other endpoint of a possible path.

For each of these, the connected component in π might given another three centers to be identified. Thus in this case a single discovered center can only give 12 centers total to be identified and thus the bound for the case of chosen centers gives the bound of the lemma. ∎

Now we are ready to make the final calculation. Letting $a = \sum_{j=1}^{g} a_j$ and defining b and c similarly we can add up the extra information as follows.

- The disappearing centers that are discovered as closest endpoints contribute a factor t^a.
- The other disappearing centers contribute a factor at most Δ^b (or less as discussed in the footnote).
- The signatures contribute at most $(As')^a$ for a constant A as signatures are only needed for disappearing centers discovered as closest endpoints.
- The centers discovered to be part of I contribute a factor $(1.01Cs)^c$.

[6] This need not be to the same neighbor as in J_j, but it is one neighbor.

- The graph structure of γ and I as well as the information which elements of γ_j are included in I_J contributes a factor B^{a+b+c}, for some constant B.
- The bits b contribute $2^{s'+8t+s'}$. This follows as we can have at most $s'+8t$ bits that are 1 (as each time a disappearing variable is discovered) and at most s' bits that are 0 (as each time a stage is ended).

Let $m = \Delta(n/T)^2$ be the total number of centers. The number of ways to choose ρ^* is[7] $2^{r_n}\binom{m}{k-(b+a)}$ where 2^{r_n} is the number of possibilities for the choice of fixed and preferred variables once the choice of centers is fixed. Similarly the number of choices for ρ is $2^{r_n}\binom{m}{k}$. This implies that the probability of having a described closed branch is bounded by

$$\frac{t^a \Delta^b s^a s^c A^{a+b+c} 2^{r_n}\binom{m}{k-(a+b)}}{2^{r_n}\binom{m}{k}} \tag{1}$$

for some (modified) absolute constant A. The quotient of the the binomial coefficients equals

$$\prod_{i=0}^{a+b-1} \frac{k-i}{m+i-k} \leq \left(\frac{k}{m-k}\right)^{a+b} = \left(\frac{Cs}{\Delta - Cs}\right)^{a+b} \leq \Delta^{-(a+b)} s^{a+b} A^{a+b},$$

for some (again different) constant A. We conclude that the probability of the closed branch in the decision tree we are analyzing is at most

$$\Delta^{-a} s^{2a+b+c} t^a A^{a+b+c}, \tag{2}$$

for again a new constant A. Applying Lemma 7.7 and modifying A we have that this is bounded by

$$\Delta^{-a} s^{27a} t^a A^a = (As^{27} t \Delta^{-1})^a. \tag{3}$$

Finally as the number of exposed centers is at most $a+b+c$ and as the numbered of queried variables is at most four times the number of exposed centers we have $a+b+c \geq s'/4$ and hence $a \geq s'/108$ and this concludes that analysis of the probability of a closed branch. Lemma 6.2 now follows from Lemma 7.3 and a final modification of the constant A.

8 Final words

Our lower bound for the Tseitin on the torus gives lower bounds for any graph in which we can embed the torus but as far as we know not on other graphs. In particular it is not clear if the same, or similar, bounds can be obtained for a random graph. It is true however that the result applies to the grid graph

[7] We need also sum this number over possible values of $a+b$ but these sequence is exponentially increasing and thus dominated by the twice the maximal term.

as it is not difficult to see that it is possible to embed the $n \times n$ torus in and $(2n+3) \times (2n+2)$ grid. The wrap around edges are mapped to paths of full length running between the vertices of the torus that are mapped in the natural way to nodes with both coordinates even.

This paper makes proof complexity "catch up" with circuit complexity when it comes to small-depth circuits containing and-gates and or-gates. We have other situation when circuit complexity still has the lead. This included small-depth circuits containing modulo p gates for a prime p and also hierarchy theorems proving that depth d circuits are much more powerful than depth $d-1$ circuits. Almost needless to say, progress on those problems would be highly interesting.

Acknowledgement. Some early ideas of this paper were discussed with Pavel Pudlak and Jakob Nordström. I am also grateful for later discussions with Ilario Bonacina, Susanna F. de Rezende Marc Vinyals, Joseph Swernofsky and Mladen Mikša. I gratefully acknowledge the many comments of the anonymous referees of this paper that hopefully helped me improve its readability.

References

[1] M. Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994.

[2] S. Bellantoni, T. Pitassi, and A. Urquhart. Approximation and small-depth frege proofs. *SIAM J. Comput.*, 21(6):1161–1179, 1992.

[3] E. Ben-Sasson. Hard examples for the bounded depth frege proof system. *Computational Complexity*, 11(3-4):109–136, 2002.

[4] E. Ben-Sasson and A. Wigderson. Short proofs are narrow–resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[5] S. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[6] M. Furst, J.B. Saxe, and M. Sipser. Parity, circuits and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

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

[8] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, STOC '86, pages 6–20, New York, NY, USA, 1986. ACM.

[9] J. Krajícek, P. Pudlák, and A. R. Woods. An exponenetioal lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Struct. Algorithms*, 7(1):15–40, 1995.

[10] J. Mehta. Tree tribes and lower bounds for switching lemmas. *CoRR*, abs/1703.00043, 2017.

[11] T. Pitassi, P. Beame, and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993.

[12] Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Poly-logarithmic frege depth lower bounds via an expander switching lemma. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 644–657, New York, NY, USA, 2016. ACM.

[13] A. Razborov. Bounded-depth formulae over the basis { AND,XOR} and some combintorial problems (in russian). *Problems of Cybernetics. Complexity Theory and Applied Mathematical Logic*, pages 149–166, 1988.

[14] A. A. Razborov. *Bounded Arithmetic and Lower Bounds in Boolean Complexity*, pages 344–386. Birkhäuser Boston, Boston, MA, 1995. Editors Peter Clote and Jeffrey Remmel.

[15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1030–1048, 2015.

[16] M. Sipser. Borel sets and circuit complexity. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, STOC '83, pages 61–69, New York, NY, USA, 1983. ACM.

[17] G. S. Tseitin. On the complexity of derivation in the proposistional calculus. In A. O. Slisenko, editor, *Studies in constructive mathematics and mathematical logic, Part II*, 1968.

[18] A. Urquhart and X. Fu. Simplified lower bounds for propositional proofs. *Notre Dame Journal of Formal Logic*, 37(4):523–544, 1996.

[19] A. C-C. Yao. Separating the polynomial-time hierarchy by oracles. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 1 –10, oct. 1985.