Polynomial-size Frege and Resolution Proofs of st-Connectivity and Hex Tautologies

Samuel R. Buss*

Department of Mathematics

University of California, San Diego

La Jolla, CA 92093-0112

sbuss@math.ucsd.edu

June 6, 2005

Abstract

A grid graph has rectangularly arranged vertices with edges permitted only between orthogonally adjacent vertices. The st-connectivity principle states that it is not possible to have a red path of edges and a green path of edges which connect diagonally opposite corners of the grid graph unless the paths cross somewhere.

We prove that the propositional tautologies which encode the st-connectivity principle have polynomial size Frege proofs and polynomial size TC^0-Frege proofs. For bounded width grid graphs, the st-connectivity tautologies have polynomial size resolution proofs. A key part of the proof is to show that the group with two generators, both of order two, has word problem in alternating logtime (Alogtime) and even in TC^0.

Conversely, we show that constant depth Frege proofs of the st-connectivity tautologies require near-exponential size. The proof uses a reduction from the pigeonhole principle, via tautologies that express a "directed single source" principle SINK, which is related to Papadimitriou's search classes PPAD and PPADS (or, PSK).

The st-connectivity principle is related to Urquhart's propositional Hex tautologies, and we establish the same upper and lower bounds on proof complexity for the Hex tautologies. In addition, the Hex tautology is shown to be equivalent to the SINK tautologies and to the one-to-one onto pigeonhole principle.

*Supported in part by NSF grants DMS-0100589 and DMS-0400848.

1 Introduction

This paper presents upper and lower bounds on proof lengths of propositional tautologies that express st-connectivity properties on grid graphs and of propositional tautologies based on the game of Hex. The st-connectivity tautologies state that two paths that cross each other, must actually cross at some point (somewhat like a generalized intermediate value theorem). Namely, if there are two paths of edges in a rectangular grid graph that begin and end at diagonally opposite edges, then the two paths must intersect somewhere.

The st-connectivity problem is the decision problem of, given a finite graph and two vertices s and t in the graph, determining whether there is a path from s to t. A *grid graph* is a graph in which vertices are rectangularly arranged and in which edges may join only vertices that are vertically or horizontally adjacent. Barrington et al. [2] studied the computational complexity of st-connectivity in constant-width grid graphs; they proved that in graphs of width d, the st-connectivity problem is complete for the circuit class Π_d of unbounded fan-in Boolean circuits of depth d. Since the AC^0-hierarchy is the union of the classes Π_d, these st-connectivity problems give a natural characterization of fragments of AC^0. D. Barrington (in unpublished work) has also investigated the low-level complexity of a number of variations of the st-connectivity problem. He considered, among other things, undirected and directed graphs and graphs in which edges were constrained to go in certain directions.

Our st-connectivity tautologies, called STCONN, will be formulated in terms of an undirected graph with all vertices of degree at most two. This undirected graph consists of two subgraphs, the *green* subgraph and the *red* subgraph; the intuition is that these subgraphs form a path of green edges and a path of red edges, and the st-connectivity tautologies state that the two graphs cannot cross without intersecting. The undirected st-connectivity tautologies are formulated in terms of propositional variables g_e and r_e that indicate the presence or absence of the undirected edge e in the two paths. We also formulate tautologies DSTCONN which express the st-connectivity principle for directed graphs. The DSTCONN tautologies are apparently weaker than the STCONN tautologies.

S. Cook and C. Rackoff [12] earlier considered a different formulation of st-connectivity that assumed that the graph is directed and that furthermore, the vertices along each path are enumerated by a function. They formulated st-connectivity tautologies with variables $g_{v,i}$ and $r_{v,i}$ that indicate whether vertex v is the i-th vertex along the green or red path.

Cook and Rackoff gave polynomial size Frege proofs of these tautologies. The idea of their Frege proofs is based on the concept of winding number. The proofs are proofs by contradiction and work by considering the i-th node along the green path and computing the winding number of the red path around that point. The proof shows that the winding number around the $(i+1)$-st vertex of the path is equal to the winding number around the i-th vertex. Then (brute-force) induction on i is used to argue that the winding number is the same at the first point of the green path as at the last point. From this, a contradiction is reached.

The motivations for the work of the present paper arose from a desire to prove lower bounds on the complexity of propositional proofs. The Σ_d-Frege proof systems are Frege systems restricted to use only Σ_d-formulas. (See the next section for more background on Frege systems.) It has been open for some time whether there are depth two tautologies, or more generally tautologies of constant depth $\leq d$, which superpolynomially separate Σ_d-Frege proof from Σ_{d+1}-Frege systems. N. Segerlind suggested that the st-connectivity problem for width d grid graphs could be good candidate for this, since the st-connectivity principles can be readily expressed as tautologies in disjunctive normal form (see Section 3 below), and since the most obvious proofs of the st-connectivity tautologies are based on expressing st-connectivity in the width d grid graph, which by [2] is known to require Π_d-formulas to express in polynomial size.

At first, we were convinced that this suggestion had some possibility of succeeding, but in the end, the results are negative. Indeed, we prove that the st-connectivity principle tautologies have polynomial size Frege proofs, and even polynomial size TC^0-Frege proofs. Our proofs improve on the above-mentioned proofs of Cook and Rackoff, since we do not need to assume that the graph is directed or that the vertices in the paths are enumerated. Secondly, we prove that, for bounded width grid graphs, there are polynomial size resolution proofs of the st-connectivity principles. As a consequence, there are polynomial size, depth two Frege proofs of the st-connectivity principles for bounded width grid graphs. Thus, the st-connectivity principles cannot be used to give superpolynomial size separations of Σ_d-Frege and Σ_{d+1}-Frege systems.

On the other hand, for general (non-bounded width) grid graphs, we show that bounded depth Frege proofs of the st-connectivity tautologies require exponential size. This is proved in Section 6 via a reduction from the one-to-one, onto pigeonhole tautologies PHP, using the fact that these pigeonhole tautologies require exponential size bounded depth Frege proofs [21, 17].

Urquhart [22] proposed propositional tautologies based on the game of Hex. The game of Hex was independently developed by P. Hein and J. Nash (see Browne [7] for more information about Hex). The Hex tautologies express the fact that a completed Hex game must have a winner. The related Hex decision problem is the problem of deciding who has won the game. Barrington proved that the Hex decision problem is equivalent to several versions of the st-connectivity problems on grid graphs. Section 7 describes the Hex tautologies, and proves they are equivalent to a grid graph tautology, SINK, which states that a directed path cannot have one source and zero sinks. We obtain as a corollary that the Hex tautologies are equivalent to the one-to-one onto PHP tautologies. Consequently, the Hex tautologies can be proved with polynomial size Frege proofs, but require exponential size bounded depth Frege proofs.

The following definition is widely used for the comparing the proof complexity of different families of tautologies.

Definition Let Q and T be families of propositional formulas. Let $\mathcal{F}+T$ denote a Frege system augmented to include all substitution instances of formulas from T. Then, we say $Q \preccurlyeq_{cd\mathcal{F}} T$ holds provided that the formulas from Q have polynomial size, constant depth proofs in the proof system $\mathcal{F}+T$.

We write $Q \equiv_{cd\mathcal{F}} T$ to mean that both $Q \preccurlyeq_{cd\mathcal{F}} T$ and $T \preccurlyeq_{cd\mathcal{F}} Q$.

The following relationships will be established for the tautologies used in this paper:

$$\begin{aligned} &\mathrm{PHP} \equiv_{cd\mathcal{F}} \mathrm{HEX} \equiv_{cd\mathcal{F}} \mathrm{SINK} \equiv_{cd\mathcal{F}} \mathrm{2SINK} \\ &\qquad \preccurlyeq_{cd\mathcal{F}} \mathrm{DSTCONN} \equiv_{cd\mathcal{F}} \mathrm{2DSTCONN} \preccurlyeq_{cd\mathcal{F}} \mathrm{STCONN} \end{aligned} \tag{1}$$

The tautologies 2SINK and 2DSTCONN are variants of SINK and DSTCONN that allow vertices to have in- and out-degrees which are equal and greater than one; they will be described in Section 3.2.

We will also introduce an undirected version of SINK, called USINK. Here we can prove that

$$\mathrm{SINK} \preccurlyeq_{cd\mathcal{F}} \mathrm{Mod}_2 \equiv_{cd\mathcal{F}} \mathrm{USINK} \preccurlyeq_{cd\mathcal{F}} \mathrm{STCONN}.$$

The tautology Mod_2 is the parity principle, or counting mod 2 tautologies. From this, we deduce that $\mathrm{STCONN} \not\preccurlyeq_{cd\mathcal{F}} \mathrm{SINK}$.

2 Preliminaries

This section quickly reviews the propositional proof systems used in this paper. For a more in-depth discussion, see [16].

2.1 Frege systems and TC^0-Frege systems

The first system we use is the *Frege proof systems*, which are the common 'textbook' proof systems for propositional logic based on modus ponens [13]. The lines in a Frege proof consist of propositional formulas built from variables p_i and from the connectives $\neg$, $\land$, $\lor$ and $\rightarrow$. There is a finite set of axiom schemes for Frege systems, for example, $\varphi \land \psi \rightarrow \varphi$ is a possible axiom scheme. The only rule of inference is (w.l.o.g.) modus ponens. Frege systems are sound and implicationally complete,

There are several common restrictions that can be put on Frege systems; for example, bounded depth Frege systems restrict lines to be formulas with negations only on variables and with a bounded number of alternations of $\lor$'s and $\land$'s (and do not permit the connective $\rightarrow$). When the formulas are restricted to be Σ_d, that is, to have d alternating levels of $\lor$'s and $\land$'s (starting with $\lor$'s), then the system is called a Σ_d-Frege system.

Other methods of restricting Frege systems arise naturally from computational complexity. One can work with bounded depth Frege systems over a larger set of connectives, such as parity gates (Mod-2 gates), Mod-k gates, or threshold gates. The TC^0-Frege systems are defined to be bounded depth Frege systems in a language which has the Boolean connectives $\neg$, $\lor$ and $\land$, and the threshold gates $T_k(x_1, \ldots, x_n)$. The T_k predicate is true when at least k of its inputs are true. Two different, but equivalent, formalizations of TC^0-Frege proof systems are given by [9] and [6].

In all the various Frege systems, a proof consists of a sequence of formulas. Each formula must either be an instance of an axiom, or be inferred from earlier formulas by a valid rule of inference. The final line in the proof is the formula proved. The length, or size, of a proof is defined to equal the total number of symbols that occur in the proof. A family of tautologies φ_i is said to have proofs of size $f(n)$ provided each φ_i has a proof of size at most $f(|\varphi_i|)$, where $|\varphi_i|$ denotes the number of symbols in φ_i.

There are several major open problems about the lengths of the propositional proofs, related to open problems such as whether $NP = coNP$. First, there is the question as to whether Frege systems or TC^0-Frege systems have polynomial size proofs of all tautologies. Also open is the question of whether Frege proofs can be superpolynomially shorter than TC^0-Frege proofs; although, it is known that Frege proofs can polynomially simulate TC^0-proofs.

For bounded depth systems, Krajíček [15] defined a notion of Σ-depth d formulas (essentially, Σ_d formulas augmented with an additional bottom level of logarithmic fanin), and he gave a superpolynomial size separation of Σ-depth d LK proofs and Σ-depth $(d+1)$-LK proofs. However, his separation applies only to refuting sequents of Σ-depth d formulas, and it is unknown whether similar results holds for smaller depth formulas.

The corresponding problem for Frege systems is whether there are constants $k \leq d$ such that there is a family of tautologies which are Σ_k-formulas for which the shortest Σ_d-Frege proofs are super-exponentially larger than the shortest Σ_{d+1}-Frege proofs. This open question was the motivation for studying the st-connectivity tautologies, as was discussed in the introduction. The hope was that polynomial size Frege proofs of the st-connectivity tautologies (which can be expressed in as polynomial-size formulas of depth $k=2$) might necessarily involve formulas which express st-connectivity properties, and hence be of complexity Π_d. Somewhat disappointingly, we prove that this is not the case. In fact, when d is constant, there are polynomial size resolution proofs of the st-connectivity principles.

We also prove that the st-connectivity tautologies have polynomial size Frege proofs, as well as polynomial size TC^0-Frege proofs, even if the width d of the graph is not constant.

2.2 Resolution systems

Resolution is a widely used proof system for refuting sets of clauses. Only the propositional fragment of resolution is used in this paper. A *literal* is defined to be either a propositional variable p, or the negation of a propositional variable, $\overline{p}$. A *clause* is a set of literals; the intended meaning of a clause is the disjunction of its literals. We assume, w.l.o.g., that no clause C contains both p and $\overline{p}$ for any variable p. Finally a set of clauses is identified with the conjunction of the clauses.

Resolution is a refutation system, in that it is used to prove the unsatisfiability of a set Γ of clauses. A resolution refutation of Γ is a sequence of clauses ending with the empty clause. Each clause in the refutation must either be from Γ, or must be inferred from two earlier clauses by the resolution rule:

$$\frac{C \cup \{x\} \quad D \cup \{\overline{x}\}}{C \cup D}.$$

The *size* of a resolution refutation is defined to equal the total number of occurrences of literals in the refutation. The *width* of a refutation is the maximum number of literals in any clause in the refutation.

Sometimes a weakening (or subsumption rule) is also permitted:

$$\frac{C}{C \cup D}$$

It is well-known that any resolution refutation R with weakening can converted into a resolution refutation R' without weakening. Furthermore the size (and number of steps) in R' is less than that of R. Therefore, we shall henceforth allow the weakening rule in our resolution proofs.

Resolution is complete, that is, if Γ is an unsatisfiable set of clauses, then there is resolution refutation of Γ. Furthermore, resolution (with weakening) is also *implicationally complete*. Let Γ be a set of clauses and C be a clause. We write $\Gamma \vDash C$ to mean that every truth assignment satisfying Γ also satisfies C. The following well-known theorem (called *Lee's Theorem*) states that resolution is implicationally complete.

Theorem 1 *Suppose $\Gamma \vDash C$. Then there is a resolution derivation of C from Γ (possibly requiring the use of the weakening rule).*

By definition, a resolution derivation of C is the same as a resolution refutation except that it ends with the clause C instead of with the empty clause.

A set Γ of clauses is equivalent to a CNF (conjunctive normal form) formula φ_Γ. Γ is unsatisfiable if and only if $\neg\varphi_\Gamma$ is a tautology. Therefore, a resolution refutation of Γ can be viewed as a proof of $\neg\varphi_\Gamma$. The next section will define a tautology STCONN which expresses the st-connectivity tautologies by first defining a set STCONN^c of clauses that express the negation of the st-connectivity principle. Thus, STCONN^c plays the role of the Γ and STCONN the role of the formula $\neg\varphi_\Gamma$.

3 The st-connectivity tautologies

3.1 Tautologies on undirected graphs

The vertices of a $d \times n$ grid graph are the ordered pairs (i, j) for $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, n$. The vertices are viewed as being in a rectangular array with d rows and n columns, with the vertex $\langle 1, 1\rangle$ as the upper left corner. By convention, grid graphs contain undirected edges. Two kinds of edges are allowed in the grid graph. First, there can be *horizontal* edges, which connect two vertices (i,j) and $(i,j+1)$. Second, there can be *vertical* edges that connect two vertices (i,j) and $(i+1,j)$. Formally, a (potential) edge is an unordered pair $\{u,v\}$ where u and v are vertices which are either vertically or horizontally adjacent. Thus, the (potential) edges in a $d \times n$ grid graph are the edges

$$\begin{aligned} &\{(i,j),(i,j+1)\} && \text{for } i=1,\ldots,d \text{ and } j=1,\ldots,n-1,\\ &\{(i,j),(i+1,j)\} && \text{for } i=1,\ldots,d-1 \text{ and } j=1,\ldots,n. \end{aligned}$$

The set of potential edges is called E. We use variables e, e_1, e_2, etc. to denote members of E.

The st-connectivity principle will be stated in terms of two graphs, G and R. The intuition is that G is a graph of "green" edges that form a path from $(1,1)$ to (d,n), and R is a graph of "red" edges that form a path from $(d,1)$ to $(1,n)$. Variables g_e are used to encode G by letting g_e have value *True* if e is an edge in G. Similarly, the variables r_e encode the red graph.

We shall define sets of clauses that describe the conditions satisfied by the green and red paths, but first we define three methods of constructing sets of clauses.

Definition Let x_1,x_2,x_3,x_4 be variables. The set $\mathit{OneOf}(x_1,x_2)$ is the set of clauses which is satisfied by a truth assignment τ iff τ assigns the value *True* exactly one of x_1 and x_2; namely,

$$\mathit{OneOf}(x_1,x_2) \;=\; \{\;\{x_1,x_2\},\;\{\overline{x_1},\overline{x_2}\}\;\}.$$

The set $\mathit{ZeroOrTwoOf}(x_1,x_2,x_3)$ is the set of clauses that is satisfied by exactly that those truth assignments that assign *True* to an even number of the three variables; namely,

$$\begin{aligned} \mathit{ZeroOrTwoOf}(x_1,x_2,x_3) \;=\; \{\;&\{\overline{x_1},x_2,x_3\},\;\{x_1,\overline{x_2},x_3\},\\ &\{x_1,x_2,\overline{x_3}\},\;\{\overline{x_1},\overline{x_2},\overline{x_3}\}\;\}. \end{aligned}$$

Similarly, we define $\mathit{ZeroOrTwoOf}(x_1,x_2,x_3,x_4)$ to be a set of clauses which is satisfied by exactly the truth assignments that assign *True* to either zero or two of the four variables. Namely,

$$\begin{aligned} \mathit{ZeroOrTwoOf}(x_1,x_2,x_3,x_4) \;=\; \{\;&\{\overline{x_1},x_2,x_3,x_4\},\;\{x_1,\overline{x_2},x_3,x_4\},\\ &\{x_1,x_2,\overline{x_3},x_4\},\;\{x_1,x_2,x_3,\overline{x_4}\},\;\{\overline{x_1},\overline{x_2},\overline{x_3}\},\\ &\{\overline{x_1},\overline{x_2},\overline{x_4}\},\;\{\overline{x_1},\overline{x_3},\overline{x_4}\},\;\{\overline{x_2},\overline{x_3},\overline{x_4}\}\;\}. \end{aligned}$$

The definition of *ZeroOrTwoOf* is "overloaded" as it depends on whether it has three or four arguments. Our notation will further exploit this overloading by writing *ZeroOrTwoOf*(X), where X must be a set containing either three or four literals. The meaning of this notation is clear; in particular, the clauses in *ZeroOrTwoOf*(X) are invariant under permutations of the literals in X.

We now define a set $\mathrm{GC} = \mathrm{GC}(d,n)$ of clauses which describes the conditions on the green edges as represented by the variables g_e. GC is the union of the following sets of clauses:

1. *OneOf*$(\{g_e : (1,1) \in e\})$.
2. *OneOf*$(\{g_e : (d,n) \in e\})$.
3. *ZeroOrTwoOf*$(\{g_e : v \in e\})$, for all vertices v except for $v = (1,1)$ and $v = (d,n)$.

Clearly, a truth assignment to the variables g_e will satisfy the clauses in GC if and only if it defines a graph which contains a simple path from $(1,1)$ to (d,n) as well as zero or more simple cycles, with the path and the cycles (if any) all vertex disjoint. (A path or a cycle is called "simple" if no vertex appears in it twice.) The path will be called the *green path*.

Similarly, the set $RC = RC(d,n)$ of clauses describes the red graph and is the union of the following sets of clauses:

1. *OneOf*$(\{r_e : (d,1) \in e\})$.
2. *OneOf*$(\{r_e : (1,n) \in e\})$.
3. *ZeroOrTwoOf*$(\{r_e : v \in e\})$, for all vertices v except for $v = (d,1)$ and $v = (1,n)$.

The clauses in RC state that the variables r_e define a *red graph* containing a simple path from $(d,1)$ to $(1,n)$ and zero or more simple cycles, with the path and the cycles vertex disjoint.

The set $\mathrm{GRDISJ}(d,n)$ expresses the condition that the red and green graphs are vertex disjoint, and contains the clauses

$$\{\overline{g_e}, \overline{r_f}\},$$

for all edges e, f such that $e \cap f \neq \emptyset$.

The set $\mathrm{STCONN}^c = \mathrm{STCONN}^c(d,n)$ is union of of the sets $\mathrm{GC}(d,n)$, $RC(d,n)$, and $\mathrm{GRDISJ}(d,n)$. This set of clauses expresses (the negation of) the st-connectivity condition for $d \times n$ grid graphs. Indeed, it is easy to verify that STCONN^c is unsatisfiable, since any green path and red path must intersect in at least one vertex. The superscript "c" in the notation stands for either "clauses" or "complement", and indicates that that satisfiability of the set of clauses is equivalent to the failure of the st-connectivity principle.

In order to work with Frege proof systems, we also define a propositional tautology STCONN that expresses the st-connectivity. STCONN^c is a set of clauses, and thus is equivalent to a CNF formula. We define STCONN to be the DNF (disjunctive normal form) formula which is equivalent to the negation of that CNF formula. Then, STCONN expresses the st-connectivity principle directly and therefore is a tautology.

As it will simplify our proofs, we shall work with a slightly modified version of the st-connectivity principle, called STCONN_+. In the modified version we assume that the first (leftmost) edges of the green and red paths are both horizontal, and that there are no other edges incident on any vertex $(i,1)$ from the leftmost column of vertices. Similarly, we assume that the last (rightmost) edges of both paths are also horizontal, and again that there are no other edges incident on any vertex (i,n) in the rightmost column. These assumptions can be made without loss of generality since one could always add additional single columns of vertices at both the left- and right-hand sides; and add horizontal edges outward from the corners of of the original grid graph.

This modified st-connectivity principle is defined as follows: Let STCONN_+^c be the set STCONN^c augmented to include the unit clauses

$$\{\overline{g_e}\} \text{ and } \{\overline{r_e}\} \text{ such that } (i,1) \in e \text{ or } (i,n) \in e \text{ for } 2 \le i < n.$$

The propositional tautologies STCONN_+ are the DNF formulas obtained from the (negation of the) STCONN_+^c clauses.

Note that the size of the formulas STCONN and STCONN_+ is polynomially bounded by d and n. The next three theorems are proved in Sections 4 and 5 and give upper bounds on the propositional proof complexity of st-connectivity.

Theorem 2 *There are polynomial size Frege proofs of the formulas* $\mathrm{STCONN}(d,n)$.

Theorem 3 *There are polynomial size TC^0-Frege proofs of the formulas* $\mathrm{STCONN}(d,n)$.

The polynomial bounds in the above two theorems depend only on the size of the formulas $\mathrm{STCONN}(d,n)$.

Theorem 4 *Let d_0 be a fixed constant. Then there are polynomial size, constant width resolution refutations of* $\mathrm{STCONN}^c(d_0,n)$.

It will suffice to prove these theorems for the STCONN_+ principles instead of the STCONN principles.

Conversely, Section 6 will establish the following theorem, which gives an exponential lower bound on the size of bounded depth Frege proofs of the st-connectivity principle.

Theorem 5 *Let $d \geq 1$. There is a constant ϵ, such that any Σ_d-Frege proof of $STCONN(n,n)$ requires size $\Omega(2^{n^\epsilon})$.*

3.2 The DSTCONN and SINK tautologies

When proving Theorem 5, we will actually establish the same result for directed versions of the st-connectivity principles, and for a "SINK" principle about paths. Since the directed graph principles are defined similarly to the undirected ones, we shall only briefly describe their definitions. A $d \times n$ directed grid graph has vertices (i,j), for $i \in \{1,\ldots,d\}$ and $j \in \{1,\ldots,n\}$. Its potential edges are the *ordered* pairs $\langle u,v\rangle$ for u and v vertices that are horizontally or vertically adjacent. The graph has red and green edges and the variables r_e and g_e, indicate whether the directed edge e is present.

$\mathrm{DSTCONN}^c$ uses clauses $\mathit{EqualCardZeroOrOne}(X,Y)$ where X and Y are sets of at most four variables. This set of clauses is satisfied precisely when either all the members of X and Y are false, or when exactly one variable from each of X and Y is true.

The set $\mathrm{DGC} = \mathrm{DGC}(d,n)$ of clauses expresses the conditions that the green edges must satisfy; they are:

1. $\mathit{OneOf}\{g_e : \mathit{tail}(e) = (1,1)\}$.

2. $\mathit{OneOf}\{g_e : \mathit{head}(e) = (d,n)\}$.

3. $\{\overline{g}_e\}$, where $\mathit{head}(e) = (1,1)$ or $\mathit{tail}(e) = (d,n)$.

3. $\mathit{EqualCardZeroOrOne}(X,Y)$, where $X = \{g_e : \mathit{head}(e) = u\}$ and $Y = \{g_e : \mathit{tail}(e) = u\}$, for every vertex u except $u = (1,1)$ and $u = (d,n)$.

These clauses will be satisfied provided the the green edge variables g_e define a simple path starting at $(1,1)$ and ending at (d,n), plus zero or more simple cycles. The DRC clauses for the red graph are defined similarly.

The set $\mathrm{DSTCONN}^c(d,n)$ of clauses is the set $\mathrm{DGC} \cup \mathrm{DRC} \cup \mathrm{GRDISJ}$. Obviously, $\mathrm{DSTCONN}^c$ is unsatisfiable. We also let DSTCONN be the DNF formula which expresses the negation of the $\mathrm{DSTCONN}^c$ clauses. DSTCONN is of course a tautology expressing the directed st-connectivity principle.

We next define a generalized version of the DSTCONN tautologies that allows a vertex to have in- and out-degrees greater than one, as long as the in- and out-degree are equal. For this, we define the set of clauses $EqualCard(X,Y)$, which is satisfied by an assignment iff it sets equal numbers of the variables in X and Y true. We then define 2DSTCONN exactly like DSTCONN except we replace the clause sets $EqualCardZeroOrOne(X,Y)$ by $EqualCard(X,Y)$. (We use the notation "2DSTCONN" because, after opposing edges are removed, each vertex has in- and out-degree at most two.)

2DSTCONN is probably not of independent interest, but will be convenient later for the proof of Theorem 5. However, it is useful to observe that there is a simple reduction from the 2DSTCONN tautologies to the DSTCONN tautologies that can be formalized with polynomial size Frege proofs. For this, fix some instance of $\mathrm{2DSTCONN}(d,n)$ and some truth assignment that encodes a directed $d \times n$ grid graph G that is purported to falsify the 2DSTCONN formula. We argue informally, with a proof that can be formalized in a bounded depth Frege, that from this we can construct a graph that falsifies an instance of DSTCONN. First, if there are any opposing edges $e_1 = (u,v)$ and $e_2 = (v,u)$ that are both in G, we just remove them from the graph. Now, each vertex in the green subgraph has in- and out-degrees at most two. To reduce the in- and out-degrees to be at most one, we triple the dimensions of the grid graph to be $3d \times 3n$. Each edge in G splits into three sub-edges, and the vertices of in- and out-degree two are transformed according to the construction shown in Figure 1. The result is a grid graph that falsifies the DSTCONN principle.

We have proved:

Theorem 6 *2DSTCONN $\preccurlyeq_{cd\mathcal{F}}$ DSTCONN.*

It is clear that the converse to the theorem holds too, so $\mathrm{2DSTCONN} \equiv_{cd\mathcal{F}} \mathrm{DSTCONN}$. In addition, it is simple to see that $\mathrm{DSTCONN} \preccurlyeq_{cd\mathcal{F}} \mathrm{STCONN}$, since, to reduce DSTCONN to STCONN, one can merely replace the directed edges with undirected edges.

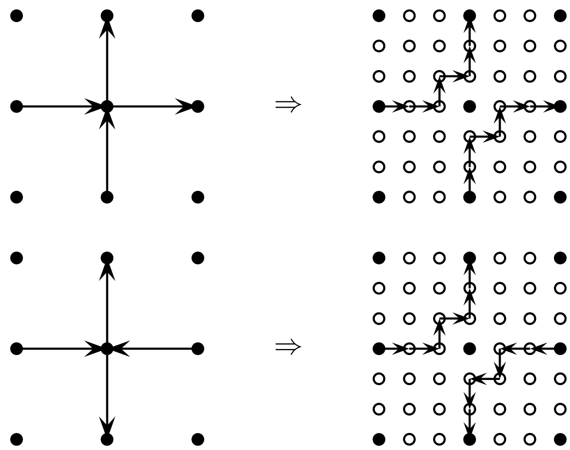

Figure 1: On the left are shown vertices in a directed graph with in- and out-degree equal to two. On the right, the graph's dimensions have been tripled, and the resulting graph has in- and out-degree at most one at every vertex.

We now define tautologies SINK = SINK(d, n) which express the fact that if every vertex in a directed graph has in- and out-degrees bounded by 1 and if there is a source node, then there must be a sink node. These tautologies are formalized with variables x_e, for e any potential edge in a $d \times n$ directed grid graph. The SINK^c clauses state that: (a) vertex $(1, 1)$ has in-degree zero and out-degree one, (b) every other vertex either has no incoming or outgoing edge, or has in- and out-degree both equal to 1. The formulas SINK are the DNF tautologies which express the negation of the conjunction of the SINK^c clauses.

The terminology "SINK" is adopted from [3] who used this name for a decision procedure from the search class PPADS. PPADS is a search class for finding a sink in a directed graph; this class was first defined by Papadimitriou under the name PSK [19, 20]. [3] also defined a search problem called "SOURCE.OR.SINK", in which the problem is to find either a source or sink in a graph other than a given known sink; this latter search problem corresponds to Papadimitriou's class PPAD. Our SINK tautologies are actually closer to the SINK.OR.SOURCE search problem than to the SINK search problem.

The 2SINK tautologies are defined similarly to the SINK tautologies, except that condition (b) is relaxed to: (b$'$) every vertex other than $(1, 1)$ has in-degree equal to out-degree. After removing opposing edges, there are

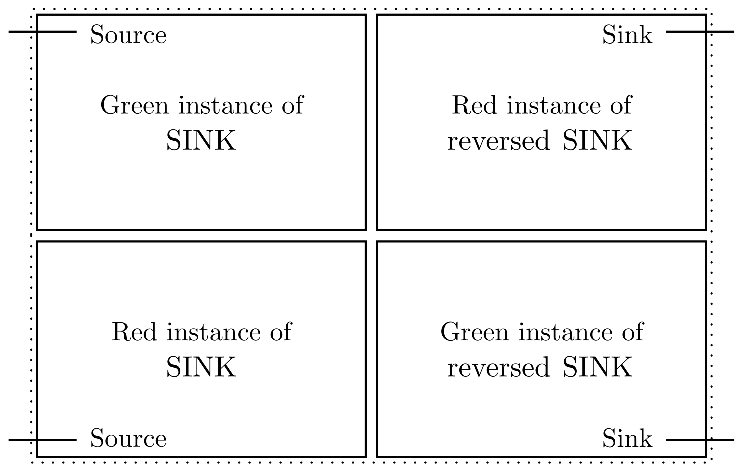

Figure 2: Showing the reduction from SINK to DSTCONN. The four copies of the instance of SINK are oriented so that the node with one edge is at the position indicated.

at most four edges adjacent to any given vertex, so the in- and out-degrees are ≤ 2. Similarly to the argument that 2DSTCONN $\equiv_{cd\mathcal{F}}$ DSTCONN, it can be shown that 2SINK $\equiv_{cd\mathcal{F}}$ SINK.

Theorem 7 SINK $\preccurlyeq_{cd\mathcal{F}}$ DSTCONN.

Proof Assume that there is a truth assignment τ that falsifies the SINK(d, n) tautology. We then define a truth assignment that violates the DSTCONN$(2d, 2n)$ tautology. Namely, we create four copies of the graph defined by the truth assignment τ. In two of the copies, we color the edges green, and in the other two copies, the edges are colored red. Then, in one of the green copies and one of the red copies, the edge directions are reversed, so that those two graphs have a sink, but no source. Then, the four copies are placed as shown in Figure 2 to create a graph that falsifies the DSTCONN$(2d, 2n)$ tautology.

It is easy to verify that this construction can be formalized with constant depth, polynomial size Frege proofs. $\square$

We define PHP = PHP(n) to be the tautology that expresses the pigeonhole principle that there is no *one-to-one and onto* mapping from $[n+1]$ to $[n]$, where $[n] = \{0, 1, \ldots, n-1\}$. As usual, the variables used in

PHP are $x_{i,j}$ expressing the condition that i is mapped to j. Thus, unlike the other tautologies, PHP is a not a grid graph tautology.

Theorem 8 SINK $\preccurlyeq_{cd\mathcal{F}}$ PHP.

Proof We use a construction from [3, §2.4]. Suppose we are given a graph which (purportedly) falsifies the SINK(d,n) tautology. We construct a 1-1, onto mapping f that falsifies the PHP$(d \cdot n - 1)$ tautology. Identifying the $d \cdot n$ vertices with $[nd]$, we define the function f as follows. If there is a directed edge from u to v in the graph, then $f(u) = v$. If there is no edge outgoing from u, then $f(u) = u$. This construction can be carried out in constant-depth Frege. We leave the rest of the details to the reader. $\Box$.

Below, Theorem 9 will establish that PHP $\preccurlyeq_{cd\mathcal{F}}$ SINK and Section 7 will prove that HEX $\equiv_{cd\mathcal{F}}$ SINK. These will suffice to prove the relationships among the various tautologies that are claimed in (1) at the end of the introduction. By (1), the upper bounds on the lengths of Frege proofs of the STCONN tautologies, which are proved in the next section, immediately imply that all of these tautologies have polynomial size Frege proofs. In addition, once Theorem 9 has been proved, the known exponential lower bounds for constant depth Frege proofs of PHP immediately imply similar exponential lower bounds for the other tautologies in (1).

4 The Frege and TC^0-Frege proofs

4.1 Vertical paths and crossing sequences

The general idea of the proofs of the STCONN formulas is as follows. We begin by assuming that STCONN is false, and we have red and green graphs that falsify the STCONN tautology. Then, for each j_0, we consider the j_0-th column of horizontal edges, namely the set of edges $\{(i, j_0), (i, j_0 + 1)\}$ for $i = 1, \dots, d$. Each edge in E is labeled with one of the symbols "g", "r", or "e" depending on the whether the edge is in the green graph, the red graph, or in neither graph. Reading down the column, we form a word w containing the d symbols labeling the d edges in the column. (There is a different w for each column.) The word w contains the symbols g, r, and e, and is called a "crossing sequence" since it lists the order in the which the red and green paths (and cycles) cross the column.

We then consider the following finitely presented group:

$$\mathcal{G} \; = \; \langle g, r;\; g^2 = 1, r^2 = 1 \rangle.$$

This notation means that the group $\mathcal{G}$ has two generators g and r, that satisfy the relations $g^2 = 1$ and $r^2 = 1$, and that no other equalities hold in $\mathcal{G}$ beyond those implied by these two relations (see [18] for more information on finitely presented groups). The elements of $\mathcal{G}$ are represented by strings over the alphabet g and r.[1] The group operation is concatenation, and the empty string ϵ is the identity element 1. However, each group element has multiple representations, for example, ϵ, gg, rr, $rggr$, etc., all represent the identity element. It is well-known that there is a very simple normal form for elements of $\mathcal{G}$: Let v be any string over the alphabet g and r representing an element of $\mathcal{G}$. The normal form of v is obtained by repeatedly removing any substring gg or rr, until no such substring is present. The resulting string is the unique normal form representation for that element of $\mathcal{G}$. (The fact that this process yields a unique normal form can be proved by showing that reduction steps that remove substrings gg and rr satisfy the Church-Rosser property. Alternately, it can be proved from the decision procedure described in Section 4.2.)

The strings w over the alphabet g, r, and e can also be viewed as representations of members of $\mathcal{G}$. The symbol e is identified with the empty string, and then w becomes a string of g's and r's and represents an element in $\mathcal{G}$.

The informal idea of the proof the STCONN tautologies can now be explained as follows. We assume, for sake of contradiction, that STCONN_+ fails. If w is the string from the first column where $j_0 = 1$, then w represents the element $gr \in \mathcal{G}$. If w is the string from the last column with $j = n-1$, then w represents the string $rg \in \mathcal{G}$. In addition, if w and w' are the string from two adjacent columns j_0 and $j_0' = j_0 + 1$, then w and w' represent the same element from $\mathcal{G}$. This is a contradiction, so thus STCONN_+ cannot be false.

The crux of the proof is of course proving that the two crossing sequence words w and w' represent the the same element of $\mathcal{G}$. The intuition for this is shown in Figure 3, which shows two columns j_0 and j_0' and their associated strings w and w'. Going from column j_0 to column j_0', a pair of r's are removed from w, and two pairs of g's are added to w'. Thus w and w' represent the same element of $\mathcal{G}$, namely gr. The intuition is that crossing sequences changes only in this way, namely by adding and/or removing pairs gg or rr.

In order to simplify the proof of the $\mathcal{G}$-equivalence of w and w', we define a more general notion of crossing sequence strings. Instead of dealing with

[1] We do not need to use the symbols g^{-1} and r^{-1} since $g^{-1} = g$ and $r^{-1} = r$.

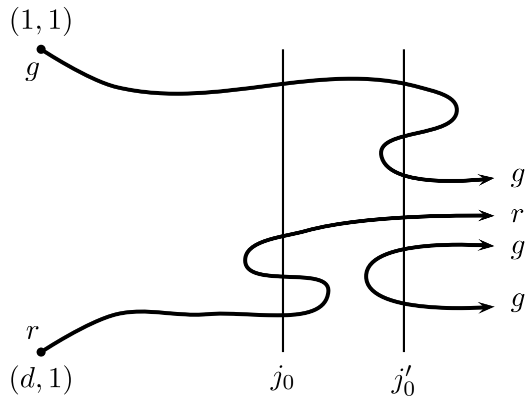

Figure 3: The crossing sequence expressions associated with the columns j_0 and j_0' are $w = grrr$ and $w' = gggrgg$ (reading from top to bottom, omitting the e's). The labels g and r indicate the colors of the paths. We have drawn the paths as curves, but in the grid graph they would actually by composed of straight edges. The arrows do not indicate that the paths are directed, but only that the paths continue on.

only crossing sequences for columns, we also deal with crossing sequences for paths that are nearly vertical, but contain up to one jog to the left. Formally, let $i \in \{0, \ldots, d\}$ and $j \in \{1, \ldots, n\}$. Then the path $\pi_{i,j}$ is as shown in Figure 4: It crosses the edges

$$\left.\begin{array}{c}\{(1,j),(1,j+1)\}\\ \vdots \\ \{(i,j),(i,j+1)\}\end{array}\right\}\ i \text{ horizontal edges}$$
$$\{(i,j),(i+1,j)\} \qquad \text{one vertical edge}$$
$$\left.\begin{array}{c}\{(i+1,j-1),(i+1,j)\}\\ \vdots \\ \{(d,j-1),(d,j)\}\end{array}\right\}\ d-i \text{ horizontal edges.}$$

Note that the cases $i = 0$ and $i = d$ need to be handled separately. When $i = 0$, the first "edge" crossed by the path is $\{(0, j), (1, j)\}$ which is not a true edge. This is instead treated as a extra *virtual* potential edge: of course the virtual potential edge does not appear in either the green or red graph. Likewise, when $i = d$, the last "edge," $\{(d, j), (d + 1, j)\}$, is only a virtual potential edge. The discussion below glosses over the possibility of virtual edges, but of course, these cases need special handling. It should be kept in mind that $\pi_{d,j}$ and $\pi_{0,j+1}$ are essentially the same path.

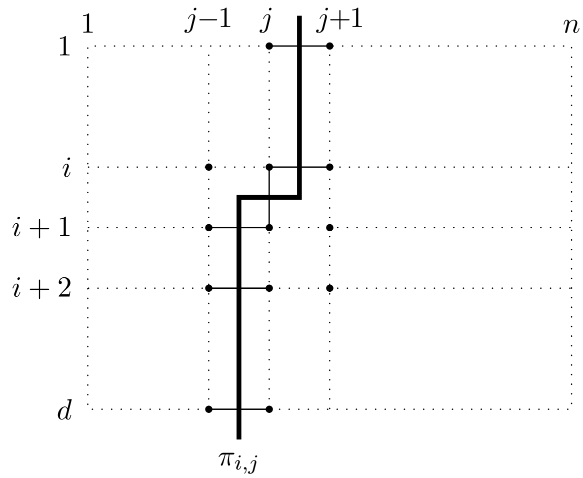

Figure 4: The path $\pi_{i,j}$. Edges crossed by $\pi_{i,j}$ are solid lines, other edges are drawn dotted.

If $i < d$, the path $\pi_{i+1,j}$ is said to *immediately succeed* the path $\pi_{i,j}$. We consider the paths as being sequentially ordered, starting with the leftmost column $\pi_{0,2}$ (or, $\pi_{d,1}$), ending with the rightmost column $\pi_{0,n}$ (or, $\pi_{d,n-1}$), and each path being immediately succeeded by a path that differs only slightly from the previous path (c.f. Figure 5).

Suppose we have a truth assignment that falsifies STCONN_+ (for sake of contradiction). Consider a path $\pi_{i,j}$. It crosses $d+1$ edges, in the order listed above. If e is the k-th such edge, define $\alpha_{i,j,k}$ to be the symbol "g" if g_e is true, to be the symbol "r" if r_e is true, and to be the symbol "e" if neither is true. Then, define $w_{i,j}$ to be the word $\alpha_{i,j,1}\alpha_{i,j,2}\cdots\alpha_{i,j,d+1}$.

We also define a "reduced" crossing sequence, $w^*_{i,j}$; this is obtained from $w_{i,j}$ by removing all occurrences of e. Namely, $w^*_{i,j}$ is a string of g's and r's; its length equals the total number of g's and r's in $w_{i,j}$, and the k-th symbol, $\beta_{i,j,k}$, of $w^*_{i,j}$ is equal to the k-th non-"e" symbol of $w_{i,j}$. Clearly, $w^*_{i,j}$ represents an element, $v_{i,j}$, of $\mathcal{G}$.

4.2 A simpler decision procedure for $\mathcal{G}$

As discussed above, a word v representing an element in $\mathcal{G}$ can be converted to a unique normal form by repeatedly cancelling out pairs "gg" and "rr". Unfortunately, this iterative process is not known to be directly formalizable in weak propositional proof systems such as Frege systems and TC^0-Frege

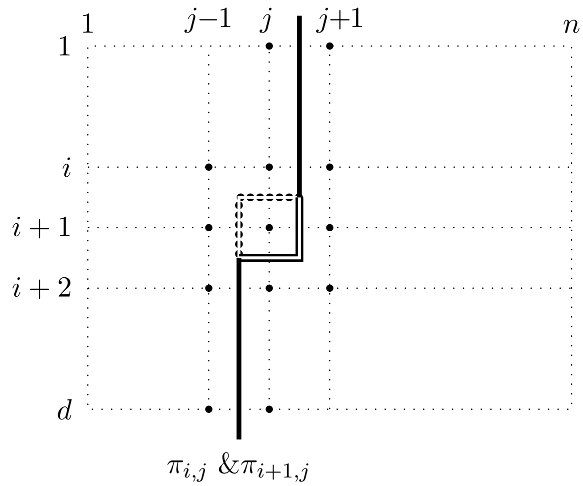

Figure 5: The path $\pi_{i,j}$ and its immediate successor $\pi_{i+1,j}$ differ only in which two of the four edges incident on (i, j) are crossed. The path $\pi_{i,j}$ contains the doubled dotted line segments, and in $\pi_{i+1,j}$ this portion is replaced by the double solid line segments.

systems. Therefore, we must give a simpler method for solving the word problem for G.

For $w \in \mathcal{G}$, the notation w^i means the i-fold multiplication of w with itself. Also, $w^0 = \epsilon$ and $w^{-i} = (w^{-1})^i$. In particular,

$$rg \;=\; (gr)^{-1}, \quad \text{and} \quad rr \;=\; gg \;=\; \epsilon \;=\; (gr)^0, \tag{2}$$

where the equalities denote equality as members of $\mathcal{G}$.

Assume that v is of even length, $v = \beta_0\beta_1\beta_2\cdots\beta_{2n-1}$, with each $\beta_i \in \{g, r\}$. Define c_ℓ for $\ell = 0, \ldots, n-1$ by

$$c_\ell \;=\; \begin{cases} 1 & \text{if } \beta_{2\ell} = g \text{ and } \beta_{2\ell+1} = r \\ -1 & \text{if } \beta_{2\ell} = r \text{ and } \beta_{2\ell+1} = g \\ 0 & \text{if } \beta_{2\ell} = \beta_{2\ell+1}. \end{cases}$$

Then, by (2),

$$v \;=\; (gr)^{c_0}(gr)^{c_1}\cdots(gr)^{c_{n-1}} \;=\; (gr)^{\sum_\ell c_\ell}.$$

In particular, $v = gr$ if and only if v has even length and $\sum_\ell c_\ell = 1$.

This decision procedure for $\mathcal{G}$ can be made even more transparent by defining the quantities d_ℓ by

$$d_\ell \;=\; \begin{cases} 1 & \text{if } \ell \text{ is even and } \beta_\ell = g, \text{ or if } \ell \text{ is odd and } \beta_\ell = r \\ 0 & \text{otherwise.} \end{cases}$$

Then, it is clear by inspection, that $d_{2\ell} + d_{2\ell+1} = 2c_\ell$. Thus, $v = gr$ holds if and only if

$$\sum_\ell d_\ell \ = \ 2. \tag{3}$$

This is the characterization we will use to express the condition that (reduced) crossing sequences represent the element "gr" in $\mathcal{G}$. The condition that $\sum_i d_i = 2$ can be expressed with a polynomial size formula in terms of the values of the d_i's. Simple properties of this summation can be proved in with Frege proofs and TC^0-Frege proofs (by the constructions in [8, 6]).

4.3 A proof formalizable in Frege and (TC^0-)Frege

The Frege and TC^0-Frege proofs of the st-connectivity tautologies proceed as follows:

1. Assume, for sake of a contradiction, that STCONN_+ is false.
2. For each path $\pi_{i,j}$, define the crossing sequence expression $w_{i,j}$.
3. For each $w_{i,j}$, define the reduced crossing sequence expression $w^*_{i,j}$.
4. Prove that $w^*_{d,1}$ is the word "gr" and that $w^*_{0,n}$ is the word "rg".
5. By "brute force induction," prove that each $w^*_{i,j}$ represents the element "gr". The argument starts with $w_{d,1}$, and then proves that if the condition holds for $w^*_{i,j}$, then it holds for the immediately succeeding reduced crossing sequence.
6. Obtain a contradiction, since $w^*_{0,n}$ cannot both equal "rg" and represent "gr".

We wish to argue that each of these six steps can be carried out with Frege or TC^0-Frege proofs.

First, in steps 2 and 3, what "define an expression" means is that propositional formulas are given that define the presence of a symbol in a given position in the word. Thus, the word $w_{i,j}$ is defined by a set of formulas $\varphi_{i,j,k,\alpha}$, for $k = 1, \ldots, d+1$ and $\alpha \in \{g, r, e\}$: the meaning of the formula $\varphi_{i,j,k,\alpha}$ is that the k-th symbol in $w_{i,j}$ is the symbol α. Likewise, $w^*_{i,j}$ is defined with formulas $\psi_{i,j,k,\beta}$. Now the formulas $\varphi_{i,j,k,\alpha}$ are trivial to define in terms of the variables g_e and r_e. The formulas $\psi_{i,j,k,\beta}$ are more complicated, but can be defined from the the fact that the k-th symbol of $w^*_{i,j}$ is the k-th symbol (if any) of $w_{i,j}$ which is not equal to e. Thus $\psi_{i,j,k,\beta}$

would be defined so to say that there is a position k' such that $\varphi_{i,j,k',\beta}$ holds and such that there are exactly $k-1$ values k'' less than k' such that $\varphi_{i,j,k'',\alpha}$ holds with $\alpha \in \{g,r\}$. Both Frege and TC^0-Frege proofs can formalize straightforward facts about counting [8, 6]; in fact, since k is from the range 1 to $d+1$, the threshold gates T_k are sufficiently strong for the counting needed.

Second, in step 5, the brute-force induction step requires arguing about how $w_{i,j}$ can differ from $w_{i+1,j}$. From Figure 5, we see that these two strings differ in at most a single pair of symbols. In fact, letting $\alpha_1\alpha_2$ be the substring in $w_{i,j}$ that is replaced by a substring $\alpha_3\alpha_4$ in $w_{i+1,j}$, we have the following possible cases (since STCONN_+ is assumed to be false):

Value of "$\alpha_1\alpha_2$"	Possible values of "$\alpha_3\alpha_4$"
"ee"	"ee", "gg", or "rr"
"eg" or "ge"	"eg", or "ge"
"er" or "re"	"er", or "re"
"gg"	"ee"
"rr"	"ee"

The argument that if $w^*_{i.j}$ represents "gr", then so does $w^*_{i+1,j}$ splits into the cases as permitted in the table. The cases are all similar, so we shall examine just the case where "ee" has been replaced by "gg". In this case, the reduced word $w^*_{i+1,j}$ differs from $w^*_{i,j}$ in that an extra substring gg has been inserted:

$$w^*_{i,j} \;=\; u_1u_2 \qquad \text{and} \qquad w^*_{i+1,j} \;=\; u_1ggu_2,$$

for strings u_1 and u_2. In the summation (3), one of the new g's is at an even position and the other at an odd position. Hence the values d_ℓ and $d_{\ell+1}$ for the two new g's are opposites, one equals 1 and the other -1. The symbols in u_2 have their positions shifted by two, so the other terms in the summation (3) are unchanged. Thus the summation (3) is unchanged by the insertion of the substring gg. The other cases from the table are proved similarly.

It is well known that the kind of reasoning used above can all be formalized with Frege and TC^0-Frege proofs. Thus, we have completed the proofs of Theorems 2 and 3.

5 Resolution and the constant width case.

We now prove Theorem 4 about the existence of bounded-width resolution refutations of STCONN^c, for constant d. In the constant d case, the resolution refutations are actually conceptually simpler than the Frege proof discussed in the previous section, since we have the luxury of using a proof which has size exponential in d. In fact, the properties of the group presentation $\mathcal{G}$ are no longer important; instead, the resolution proof exploits the fact that the change from $w^*_{i,j}$ to $w^*_{i+1,j}$ is only local.

Recall that each path $\pi_{i,j}$ crosses a set of $d+1$ edges. The $2(d+1)$ variables g_e and r_e for edges e which are crossed by $\pi_{i,j}$ are called the (i,j)*-variables*. The word $w_{i,j}$ was defined from truth values of the (i,j)-variables. A truth assignment to the (i,j)-variables said to be *banned* if the corresponding word $w_{i,j}$ does not represent the element gr in $\mathcal{G}$. For each banned (i,j)-assignment τ, let B_τ be the clause of size $2(d+1)$ that is falsified exactly by τ; that is,

$$
\begin{aligned}
B_\tau \;\; = \;\; & \{g_e : \tau(g_e) = F\} \cup \{\overline{g_e} : \tau(g_e) = T\} \\
& \cup \{r_e : \tau(r_e) = F\} \cup \{\overline{r_e} : \tau(r_e) = T\}.
\end{aligned}
$$

Then, let $\mathcal{B}_{i,j}$ be the set of clauses

$$
\mathcal{B}_{i,j} \;=\; \{B_\tau : \tau \text{ is a banned } (i,j)\text{-assignment}\}.
$$

The resolution refutation of STCONN^c_+ proceeds as follows. It first derives all the clauses in $\mathcal{B}_{(d,1)}$, which is easily done from the unit clauses in STCONN^c_+ (using the weakening rule). Then, having derived all the clauses in $\mathcal{B}_{i.j}$, it then derives all the clauses in $\mathcal{B}_{i+1,j}$, by the method described below. At the end, it uses resolution with unit clauses in STCONN^c_+ to derive a contradiction from a clause in $\mathcal{B}_{0,n}$, namely the clause that contains the literals $\overline{r_{(1,n-1),(1,n)}}$ and $\overline{g_{(d,n-1),(d,n)}}$ and contains the rest of the $(0,n)$-variables unnegated.

The method by which the $\mathcal{B}_{i+1,j}$ clauses are derived from $\mathcal{B}_{i,j}$ deserves more explanation. The path $\pi_{i+1,j}$ differs from the path $\pi_{i,j}$ in only the four edges

$$
\begin{aligned}
e_1 &= \{(i,j),(i+1,j)\} & \qquad e_3 &= \{(i+1,j),(i+1,j+1)\} \\
e_2 &= \{(i+1,j-1),(i+1,j) & \qquad e_4 &= \{(i+1,j),(i+2,j)\}
\end{aligned}
$$

(see Figure 5). Thus, the (i,j)-variables differ from the $(i+1,j)$-variables only in that the former include the four variables g_{e_1}, g_{e_2}, r_{e_1} and r_{e_2}, and that the latter include the four variables g_{e_3}, g_{e_4}, r_{e_3} and r_{e_4}. We let $D_{i,j}$ be the set of eight variables g_{e_i}, r_{e_i}; and let $C_{i,j}$ be the set of variables which are both (i,j)- and $(i+1,j)$-variables. Consider a particular $B_\tau \in \mathcal{B}_{i+1,j}$. We let $\mathcal{B}^{-\tau}$ be the set of clauses B_σ for all banned (i,j)-assignments σ that agree with τ on the variables $C_{i,j}$. $B^{-\tau}$ contains at most sixteen clauses, since there is only sixteen ways to set the values of σ on the four variables g_{e_1}, g_{e_2}, r_{e_1} and r_{e_2}.

Using the reasoning used in the proof in the previous section, we know that

$$\mathcal{B}^{-\tau} \cup \text{STCONN} \models B_\tau.$$

In fact, letting $\text{STCONN}^{i,j}$ be the clauses in STCONN that involve only the variables in $D_{i,j}$,

$$\mathcal{B}^{-\tau} \cup \text{STCONN}^{i,j} \models B_\tau.$$

By the implicational completeness of resolution, it follows that there is a derivation of B_τ from the clauses in $\mathcal{B}^{-\tau}$ and the clauses in $\text{STCONN}^{i,j}$.

Putting all these resolution derivations together gives the derivation of all the $\mathcal{B}_{i+1,j}$ clauses. Also, by inspection, the width of the clauses in the resolution refutation is only $d+O(1)$. Thus, the overall resolution refutation of $\text{STCONN}^c(d,n)$ has polynomial size and uses only clauses with width at most $d+O(1)$.

6 Lower bounds for constant depth Frege proofs

This section establishes the exponential lower bound of Theorem 5. Since similar exponential lower bounds for the pigeonhole principle tautologies PHP have already been proved by [21, 17], it will suffice to will suffice to prove that $PHP \preccurlyeq_{cd\mathcal{F}}$ 2SINK.

Theorem 9 *$PHP \preccurlyeq_{cd\mathcal{F}}$ 2SINK.*

Proof We describe informally a construction that will translate a violation of the $\text{PHP}(n)$ tautology into a a directed grid graph that violates the $\text{2SINK}(2n+1, 2n+2)$ tautology. Our construction is shown in Figure 6 and will be carried out informally. We leave it to the reader to verify that the construction can be defined with bounded depth formulas and that constant depth Frege proofs can prove all the relevant properties of the construction.

Assume that $f : [n+1] \to [n]$ is one-to-one and onto. We use f to construct a $(2n+1) \times (2n+2)$ grid graph G which violates the 2SINK principle. The central column of G contains the vertices $(1, n+1), (2, n+1)$, $\ldots, (2n+1, n+1)$. Our intuition is that we identify $(i, n+1)$ for $i = 1, \ldots, n+1$ with the domain elements of f by letting vertex $(i+1, n+1)$ correspond to $i \in [n+1]$ in the domain of f; in keeping with this intuition, we let $D_i = (i+1, n+1)$. The remaining vertices in the central column can be identified with elements in the range of f by letting $(n+2+i, n+1)$ correspond to $i \in [n]$; we thus let $R_i = (n+2+i, n+1)$. The edges of G are set as follows.

(a) G contains a path with goes horizontally from $(1,1)$ to $D_0 = (1, n+1)$. Namely, it contains the edges $\langle (1,i), (1,i+1) \rangle$ for $i = 1, \ldots, n$. This is the path starting in the upper right corner of Figure 6.

(b) For each $i \in [n]$, G has a directed path from R_i to D_{n-i}. This path starts at $R_i = (n+2+i, n+1)$ and proceeds horizontally leftward to $(n+2+i, n-i)$, it then proceeds vertically up to $(n+1-i, n-i)$, and from there proceeds horizontally rightward to $D_{n-i} = (n+1-i, n+1)$. These are the other paths in the left half of Figure 6.

(c) For each $i \in [n+1]$, G has a path from D_i to R_j, where $j = f(i)$. This path starts at $D_i = (i+1, n+1)$ and proceeds horizontally rightward to $(i+1, n+2+i)$; it then proceeds vertically down to $(n+2+j, n+2+i)$, and finally goes horizontally leftward to $R_j = (n+2+j, n+1)$. These are the paths in the right half of Figure 6.

Examination of Figure 6 shows the correctness of the reduction from PHP to SINK. We claim that the reduction is definable with constant depth polynomial size formulas. For this, we need to find formulas that define the edges' values g_e in terms of the PHP variables $x_{i,j}$. The edges that are not in the lower right quadrant of G are fixed, and independent of the function f, hence, the variables g_e, for e an edge not in the lower right quadrant are just constants *True* or *False*. Consider a variable g_e for an edge in the lower right quadrant. The edge e either is vertical and of the form $e = \langle (n+1+j, n+2+i), (n+2+j, n+2+i) \rangle$, or is horizontal and of the form $e' = \langle (n+2+j, n+2+i), (n+2+j, n+1+i) \rangle$. The vertical edge, e, is present in G if and only if $f(i) \geq j$. Thus the variable g_e of SINK can be defined by

$$g_e \;\Leftrightarrow\; \bigvee_{j \leq k < n} x_{i,k}.$$

Similarly, the horizontal edge is present in G if and only if $f^{-1}(j) \geq i$, so

$$g_{e'} \;\Leftrightarrow\; \bigvee_{i \leq k \leq n} x_{k,j}.$$

Furthermore, polynomial-size, constant depth Frege proofs can prove the correctness of the reduction from the instance of PHP to the instance of SINK^c. □

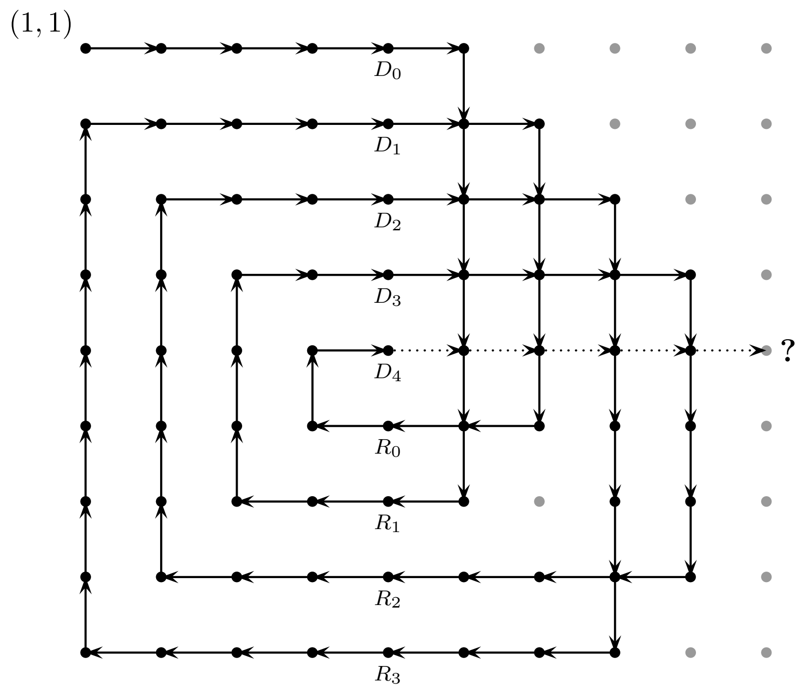

Figure 6: How to build an instance of 2SINK from an instance of PHP. The dotted path from D_4 would connect back up to a R_i point if we had a contradiction to the pigeonhole principle.

Consequently, there are exponential lower bounds on the size of constant depth Frege proofs for all the graph tautologies we have considered.

It is open whether the reductions SINK $\preccurlyeq_{cd\mathcal{F}}$ DSTCONN $\preccurlyeq_{cd\mathcal{F}}$ STCONN are strict. However, we are able to prove that SINK $\not\equiv_{cd\mathcal{F}}$ STCONN by a proof that we only sketch here.

Let Mod_p be the family of tautologies that express the counting modulo p principle [1, 5]. It is well-known that PHP $\preccurlyeq_{cd\mathcal{F}}$ Mod_p, for all p. (As throughout this paper, PHP means the one-to-one, onto version of the pigeonhole principle.) Thus, SINK $\preccurlyeq_{cd\mathcal{F}}$ Mod_p for all p.

Let USINK be the undirected analogue of SINK that states that an undirected grid graph cannot have a single vertex of degree one with the rest of the nodes of degree either zero or two. Clearly, SINK $\preccurlyeq_{cd\mathcal{F}}$ USINK. Also, similarly to the proof of Theorem 7, it can be shown that USINK $\preccurlyeq_{cd\mathcal{F}}$

STCONN. In addition, by using a construction similar to the proof of Theorem 9, $\mathrm{Mod}_2 \preccurlyeq_{cd\mathcal{F}}$ USINK. Then, if USINK $\equiv_{cd\mathcal{F}}$ SINK were valid, we would have $\mathrm{Mod}_2 \preccurlyeq_{cd\mathcal{F}} \mathrm{Mod}_p$ for all p. But this has been shown to be false by [4, 11, 10]. Thus we have USINK $\not\preccurlyeq_{cd\mathcal{F}}$ SINK, and hence STCONN $\not\preccurlyeq_{cd\mathcal{F}}$ PHP.

It also can be shown that USINK $\preccurlyeq_{cd\mathcal{F}}$ Mod_2, and thus USINK $\equiv_{cd\mathcal{F}}$ Mod_2. The reduction USINK $\preccurlyeq_{cd\mathcal{F}}$ Mod_2 can be proved with the construction used by [3, §2.5] to prove that the search problem LEAF is many-one reducible to the search problem LONELY. They also proved the equivalence of LEAF and LONELY, and these two search problems can be viewed as analogues of the USINK and Mod_2 tautologies, but with the important difference that USINK is formulated in terms of a grid graph.

7 The Hex tautologies

Urquhart [22] proposed tautologies based on the game of Hex, which express the fact that any end configuration of a game of Hex must have a winner. We will very briefly review the game of Hex; for more information, consult Browne [7]. The game of Hex is played on an $m \times n$ parallelogram tiled with hexagons, of the type shown in Figure 7. Two players alternate placing stones into hexagons; one player places red stones, the other blue stones. Each hexagon can hold only one stone. The player with red stones (resp., blue stones) wins if he builds a path of red (resp. blue) stones that connects a hexagon in the top row with a hexagon in the bottom row (resp, the left column and the right column).

The Hex tautologies express the fact that, once the board has been completely filled, one of the two players has won. In the spirit of Urquhart's suggestion,[2] the propositional variables for the Hex game are R_h, B_h, M_h, and C_h, for each hexagon h. The names R, B, M, C are mnemonics for "red,", "blue," "magenta," and "cyan." The intent is that the difference between red and magenta hexagons is that red ones are connected to the top row, and the magenta ones to the bottom row. Likewise, the intent is that the blue hexagons are connected to the left border and the cyan ones to the right border. The general idea of the Hex tautologies is that it is impossible that there is both no red hexagon adjacent to a magenta hexagon and no blue hexagon adjacent to a cyan one.

To formally define the HEX = HEX(n) tautologies, we use a slightly larger Hex game board which is size $(n+2) \times (n+2)$. (See Figure 8.)

[2] Our formulation is similar to Urquhart's and is equivalent in the sense of $\equiv_{cd\mathcal{F}}$.

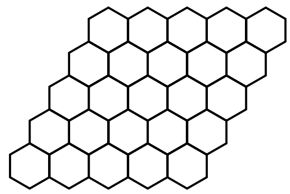

Figure 7: An empty 5×5 Hex board.

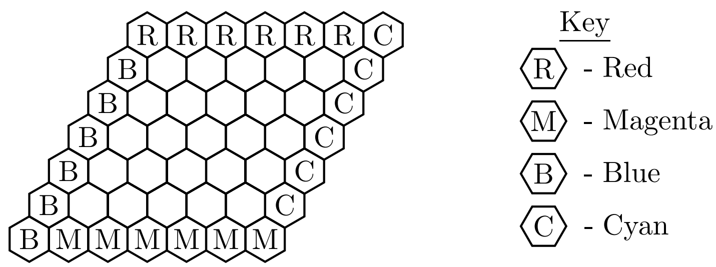

Figure 8: A 7×7 Hex board with border colors filled in.

On this augmented board, the upper row is forced to be red, the lower row magenta, the left column blue, and the right column cyan. The rest of the board is the usual $n \times n$ Hex game. The HEX^c clauses include:

1. Unit clauses expressing the conditions that each border hexagon has its correct color. For example, the unit clauses $\{R_h\}$ for each red hexagon h along the top border.

2. Clauses $\mathit{OneOf}(R_h, M_h, B_h, C_h)$ stating that exactly one of the four variables in true, i.e., that each hexagon h has exactly one color.

3. Clauses saying that no red and magenta hexagons are adjacent, and no blue and cyan hexagons are adjacent. These are $\{\overline{R}_h, \overline{M}_{h'}\}$ and $\{\overline{B}_h, \overline{C}_{h'}\}$, for each pair of h, h' of adjacent hexagons.

As usual, the tautology HEX is the DNF formula which is equivalent to the negation of the HEX^c clauses.

It is, of course, a well-known fact that a completely filled in Hex game has a winner; thus the formula HEX is indeed a tautology. The simplest proof that Hex always has a winner involves a reduction to the SINK principle.

Theorem 10 *HEX $\preccurlyeq_{cd\mathcal{F}}$ SINK.*

Proof The proof of Gale [14] (see also Browne [7, Appendix D]) can be formalized as a reduction to SINK. Suppose we are given a truth assignment that falsifies the HEX(n) formula. We construct a directed graph G which is a counterexample to the SINK principle. The potential edges in G are the edges of the hexagons in the augmented Hex board. The edges that are actually present are the edges between blue (B) and magenta (M), oriented so that blue is on the left side of the edge and magenta on the right. This graph has a source at the lower left corner of the augmented Hex board. In addition, we claim every other node has in-degree equal to out-degree. To verify this, note that if a vertex v is the head of an edge e, then v is of course adjacent to the blue and magenta hexagons on the sides of e. Since the HEX tautology fails, the third hexagon cannot be red or cyan, and hence must be blue or magenta. In either case, there is an outgoing edge from v.

Now, G is not a grid graph, but it can be mapped to one by discretizing to a sufficiently fine rectangular grid. From this, we arrive at an assignment that falsifies SINK.

We leave it the reader to verify that this can be formalized with polynomial size, constant depth Frege proofs. □

The converse to Theorem 10 holds too.

Theorem 11 SINK $\preccurlyeq_{cd\mathcal{F}}$ HEX.

Proof We give only a sketch of a proof, which we claim can be formalized as a polynomial size, constant depth Frege proof. Suppose we are given a truth assignment that falsifies SINK. That is, there is a $d \times n$ directed grid graph G with a source at $(1,1)$ that has no sink. The rest of the vertices of G all have in- and out-degree both equal to zero or to one. By refining G to have dimensions $(4d) \times (4n)$, we can convert the paths (or cycles) in G into bundles of four paths (or cycles). We do this by splitting each edge into four parallel edges, and then hooking up the edges where the path makes a turn, so that there are four paths. The four paths are, of course, completely disjoint. Note that this construction can all be done locally.

We now color these four paths and their vertices. In order from left to right, they are assigned colors red, blue, magenta, and cyan. It is clear that blue vertices lie adjacent only to red, magenta, and blue vertices. Likewise, magenta vertices lie adjacent only to blue, cyan, and magenta vertices. The rest of the vertices in the graph can be colored red, and then all the vertices are colored, with no red and magenta vertices adjacent and no blue and cyan vertices adjacent. The colors of the vertices on the boundary of the graph are all either cyan or red, with the exception of the four vertices at the source of the four paths.

We claim that this grid graph with four colored paths is topologically equivalent to a Hex game with no winner. For this, the rectangular grid graph of colored vertices is mapped into a parallelogram of hexagons. The parallelogram is picked to have resolution somewhat finer than the grid graph (three or four times finer, say), and the grid graph is mapped over the hexagon by an affine transformation and then the hexagons are colored with the color of the closest grid graph vertex. Finally, the four paths are extended to wrap around the outside of the parallelogram. The red path is extended from its source, to wrap clockwise along the top of the parallelogram. The cyan path is extended to wrap counterclockwise around to go down the left side, along the bottom and then up the right boundary. After that, the magenta path is extended counterclockwise around to cover the left boundary and the bottom boundary. Finally, the blue path is extended counterclockwise to cover the left boundary. This results in a parallogram of colored hexagons that is a Hex game with no winner; that is, that violates the HEX tautology. This is the desired contradiction to the HEX tautologies. □

As the SINK principle has already been proved equivalent to the one-one onto pigeonhole principle, we also have

Corollary 12 HEX $\equiv_{cd\mathcal{F}}$ PHP.

Gale [14] also discusses the equivalence of the Hex principle that every completed game has a winner with the Brower fixed point theorem. In addition, he mentions that the principle that a Hex game has a single winner is equivalent to the Jordan curve theorem. However, we do not know any good way to formulate the principle that a Hex game has only one winner as a set of clauses or as a simple polynomial-size propositional tautology. The only methods we know for formulating polynomial-size tautologies that state that every Hex game has a winner involve introducing extra variables (say, to indicate hexagons on a path); however, these do not give particularly elegant formulations of the Hex game winner principle. Papadimitriou [19, 20] discusses the complexity of a number of principles, including the Brower fixed point theorem and Sperner's lemma, and related search problems are defined by Beame et al. [3]. Possibly these ideas can lead to further interesting propositional tautologies for proof complexity.

Acknowledgements. I wish to thank A. Beckmann, D. Barrington, S. Cook, J. Krajíček, T. Pitassi, N. Segerlind, and A. Urquhart for helpful discussions and correspondence. This paper had its genesis in a precursor of Theorem 4 which was discovered during a conversation with N. Segerlind.

References

[1] M. Ajtai, *The independence of the modulo p counting principles*, in Proceedings of the 26th Annual ACM Symposiom on Theory of Computing, Association for Computing Machinery, 1994, pp. 402–411.

[2] D. A. M. Barrington, C.-J. Lu, P. B. Miltersen, and S. Skyum, *Searching constant width mazes captures the AC^0 hierarchy*, in Proceedings, 15th Annual Symposium on Theoretical Aspects of Computer Science (STACS'98), Lecture Notes in Computer Science #1373, Berlin, 1998, Springer-Verlag, pp. 73–83.

[3] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi, *The relative complexity of NP search problems*, Journal of Computer and System Sciences, 57 (1998), pp. 3–19.

[4] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák, *Lower bounds on Hilbert's Nullstellensatz and propositional proofs*, Proceedings of the London Mathematical Society, 73 (1996), pp. 1–26.

[5] P. Beame and T. Pitassi, *An exponential separation between the parity principle and the pigeonhole principle*, Annals of Pure and Applied Logic, 80 (1996), pp. 195–228.

[6] M. L. Bonet, T. Pitassi, and R. Raz, *On interpolation and automatization for Frege systems*, SIAM Journal on Computing, 29 (2000), pp. 1939–1967.

[7] C. Browne, *Hex Strategy: Making the Right Connections*, AK Peters, Natick, MA, 2000.

[8] S. R. Buss, *Polynomial size proofs of the propositional pigeonhole principle*, Journal of Symbolic Logic, 52 (1987), pp. 916–927.

[9] S. R. Buss and P. Clote, *Cutting planes, connectivity and threshold logic*, Archive for Mathematical Logic, 35 (1996), pp. 33–62.

[10] S. R. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi, *Linear gaps between degrees for the polynomial calculus modulo distinct primes*, Journal of Computer and System Sciences, 62 (2001), pp. 267–289.

[11] S. R. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. A. Razborov, and J. Sgall, *Proof complexity in algebraic systems and bounded depth Frege systems with modular counting*, Computational Complexity, 6 (1996/1997), pp. 256–298.

[12] S. A. Cook and C. Rackoff. Unpublished course notes, 1997.

[13] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic, 44 (1979), pp. 36–50.

[14] D. Gale, *The game of Hex and the Brower fixed-point theorem*, American Mathematical Monthly, 86 (1979), pp. 818–827.

[15] J. Krajíček, *Lower bounds to the size of constant-depth propositional proofs*, Journal of Symbolic Logic, 59 (1994), pp. 73–85.

[16] ——, *Bounded Arithmetic, Propositional Calculus and Complexity Theory*, Cambridge University Press, Heidelberg, 1995.

[17] J. Krajíček, P. Pudlák, and A. Woods, *Exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle*, Random Structures and Algorithms, 7 (1995), pp. 15–39.

[18] W. Magnus, A. Karrass, and D. Solitar, *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*, Dover, New York, 2nd ed., 1976.

[19] C. H. Papadimitriou, *On graph-theoretic lemmata and complexity classes (extended abstract)*, in Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (Volume II), IEEE Computer Society, 1990, pp. 794–801.

[20] ——, *On the complexity of the parity argument and other inefficient proofs of existence*, Journal of Computer and System Sciences, 48 (1994), pp. 498–532.

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

[22] A. Urquhart, *Hex example.* Email correspondence, Toronto theory group, April 2001.