Research Proposal & Progress Report

Alexander Hertel *

July 5, 2006

1 Introduction

This paper has two purposes. The first purpose is to provide a progress report describing research and results completed to date. Its second purpose is to describe current open problems, and lay out an area for future research that should yield further interesting results. The completed results are categorized into four papers, all of which fall in the area of proof complexity. These results are tied together chronologically, with the later results having intuitive relationships to the earlier ones.

Our first paper [Her06] describes a Non-Hamiltonicity Proof System (NHPS), together with proofs of soundness, completeness, exponential lower bounds, and simulation results relating it to other proof systems. These results are described below in Section 2. The NHPS came from the analysis of an algorithm for determining non-Hamiltonicity. The overall direction of this research was to better understand graphical proof systems, as well as general limitations of algorithms of this type. Specifically, since the NHPS deals with local configurations in the graph, the goal was to prove lower bounds for graph algorithms which involved ideas such as local search. This motivated the search for a simulation result with Tree Resolution (T-RES), which necessarily required the use of a reduction. The development of such a reduction from the Hamiltonian Cycle problem to SAT led to the next result.

This next result is described in our second paper [HH06], which formalizes the concept of dangerous reductions. We were able to show that slightly different versions of the reduction from Hamiltonian Cycles to SAT give results with drastically different proof complexities. These results are described below in Section 3. However, investigating which translations require easy proofs and which ones require hard proofs needed a more efficient method than drawing out clumsy DPLL trees. The Prover / Delayer Game from [BSIW04] is known to simplify lower bounds on proof sizes, but we discovered that it is also an excellent tool for giving upper bounds which are much simpler than DPLL upper bounds.

This led us to our third paper, [HU06b], in which we explore Prover / Delayer Game upper bounds for T-RES proof size. These results are described below in Section 4.

Our fourth paper [HU06a] is probably the one containing the most significant results. In it we explore the proof complexity of Intuitionistic Propositional Logic. This research was not motivated by any of the previous work, but the result interestingly is highly relevant to the research on dangerous reductions. These results are described below in Section 5.

*This research supported by NSERC and the University of Toronto Department of Computer Science

2 Non-Hamiltonicity Proof System

In this paper [Her06] we describe the NHPS, a proof system for non-Hamiltonicity. This proof system was formalized partly as a means of analyzing an algorithm for non-Hamiltonicity, and partly from the aim of diversifying the field of proof complexity by developing another graphical proof system.

We prove that the NHPS is both sound and complete, and provide families of graphs for which it has exponential size lower bounds, where size is defined to be the number of nodes in the NHPS proof-tree. We call one of these families the $G_{\frac{n}{2},\frac{n}{2}}$ graphs.

Theorem 2.1. *The lengths of tree-like* NHPS *proofs have* $\Omega(3^n)$ *size-lower bounds.*

Using the reduction and definitions from the research on dangerous reductions as well as Prover / Delayer Game upper bound techniques, we show that T-RES effectively p-simulates (see Section 3) the NHPS:

Theorem 2.2. *Under the reduction described in [HH06] which uses all of the clause groups,* T-RES *effectively p-simulates* NHPS*.*

In addition, we again use the reduction from [HH06] as well as an example from [HU06b] to show an effective exponential separation between T-RES and NHPS. The example using the Prover / Delayer Game is given below in Theorem 4.5, which shows that the $H(G_{\frac{n}{2},\frac{n}{2}})_{T,O,1,F}$ formulas described in Section 3 below have polynomial size upper bounds. Since the NHPS has exponential lower bounds for the $G_{\frac{n}{2},\frac{n}{2}}$ graphs (see Theorem 2.1 above), but T-RES has polynomially-bounded proofs for the $H(G_{\frac{n}{2},\frac{n}{2}})_{T,O,1,F}$, we have an effective separation between the two proof systems.

These results provide concrete connections between the NHPS, Dangerous Reductions, and the Prover / Delayer Game.

3 Formalizing Dangerous Reductions

In this paper [HH06] which was written together with Philipp Hertel, we give a theoretical proof confirming empirical results from the Propositional Reasoning community.

The area of SAT-Solving has been very successful. Because of this, research into solving other $\mathcal{NP}$-Complete problems has largely consisted of translating them into SAT, and then applying researchers' considerable experience with SAT-Solvers. However, this strategy is not without its pitfalls. For example, researchers were noticing that different reductions, when applied to the same input, could output formulas with drastically different proof complexities. They similarly found that adding redundant clauses which were not technically required for the correctness of a reduction would sometimes make formulas considerably easier to solve. The problem of exploring these phenomena was listed in [KMS97, KS03] as being one of the most interesting open problems in the area of Propositional Reasoning.

To this end we provide formal examples supporting these empirical results as well as proofs explaining the underlying mechanisms at play. We give two examples of very similar and natural reductions from the Hamiltonian Cycle problem to SAT, which, when applied to a certain family of graphs, produces formulas with drastically different proof complexities. More specifically, prove the two theorems below. Note that H is the reduction being used, the subscripts indicate slight differences between the reductions, and K_n^* is the family of graphs being translated to formulas.

The polynomial upper bounds for T-RES are given by:

Theorem 3.1. T-RES *proofs for the unsatisfiability of* $H(K_n^*)_{T,O,F}$ *formulas have* $O(n^2)$ *size upper bounds, where* n *is the number of distinct variables contained in the formulas.*

The exponential lower bounds for AC^0-Frege systems are given by:

Theorem 3.2. *The size of* AC^0-Frege *proofs for the unsatisfiability of* $H(K_n^*)_{T,1,F}$ *formulas have* $\Omega(2^{\sqrt[d]{n}})$ *lower bounds, where* d *is the depth of the Frege proof, and if there exist size-*N AC^0-Frege *proofs restricted by* $m_{x,n} = 1$ *of* $H(K_n^*)_{T,1,F}$*, then there exist size-*$N + O(n^3)$ *proofs of* $fPHP_{n-2}^n$.

This has an immediate corollary which is relevant to Propositional Reasoning:

Corollary 3.3. *No SAT algorithm based on* AC^0-Frege *nor any weaker proof system can efficiently solve* $H(K_n^*)_{T,1,F}$ *formulas. This includes* DPLL *as well as clause-learning based SAT-solver algorithms.*

Apart from their relevance to the open problems from Propositional Reasoning, these results have proof complexity implications. The first implication is that this result allows us to define the intuitive notion of dangerous and safe reductions. There are three different types of reductions. Let α be a proof system for a language L_1, let β be a proof system for a language L_2, and let $R : L_1 \to L_2$ be a reduction from L_1 to L_2. Informally,

- If R takes easy instances for α, and converts them to hard instances for β, then we say that R is explosive. This corresponds to our idea of a dangerous reduction.
- If R takes instances for α, and converts them to instances for β which have the same complexity, then we say that R is stable. This corresponds to our idea of a neutral reduction.
- If R takes hard instances for α, and converts them to easy instances for β, then we say that R is implosive. This corresponds to our idea of a beneficial reduction.

More formally,

Definition 3.4. *Let* α *be a proof system for a language* L_1*, let* β *be a proof system for a language* L_2*, and let* $R : L_1 \to L_2$ *be a reduction from* L_1 *to* L_2*. If there exists some family of strings* $X = \{x_1, x_2, ...\}$*,* $X \subseteq L_1$ *such that for all* k *and for all* $x_i \in X$ *there exists an* α*-proof* P_1 *of* x_i*, but there exists no* β*-proof* P_2 *of* $R(x_i)$ *such that* $|P_2| \leq |P_1|^k$*, then we say that the reduction* R *is* (α, β)*-Explosive on the set* X*.*

Definition 3.5. *Let* α*,* β*,* L_1*,* L_2*, and* R *be as in Definition 3.4. If there exist constants* k_1 *and* k_2 *and a family of strings* $X = \{x_1, x_2, ...\}$*,* $X \subseteq L_1$ *such that for any* α*-proof* P_1 *of* x_i *there exists a* β*-proof* P_2 *of* $R(x_i)$ *where* $|P_2| \leq |P_1|^{k_1}$ *and* $|P_1| \leq |P_2|^{k_2}$ *then we say that the reduction* R *is* (α, β)*-Stable on the set* X*.*

Definition 3.6. *Let* α*,* β*,* L_1*,* L_2*, and* R *be as in Definition 3.4. If there exists some family of strings* $X = \{x_1, x_2, ...\}$*,* $X \subseteq L_1$ *such that for all* k *and for all* $x_i \in X$ *there exists a* β*-proof* P_2 *of* $R(x_i)$ *but there exists no* α*-proof* P_1 *of* x_i *such that* $|P_1| \leq |P_2|^k$*, then we say that the reduction* R *is* (α, β)*-Implosive on the set* X*.*

We provide a number of examples of explosive, stable, and implosive reductions, many of which are drawn from the three other papers mentioned in the introduction.

This leads to the second implication for proof complexity: When comparing the relative proof complexity of two proof systems over different languages, one necessarily must perform a reduction. However, as we have seen, reductions can either unfairly inject complexity by leaving out critical clauses, or 'cheat' by doing some of the work that the target proof system might not be able to do itself. This makes it impossible to carry out an objective comparison of proof system strength, because it is impossible to tell what role is being played by the reduction. The use of the standard definitions of p-simulation and exponential separations when comparing proof systems over different languages should therefore be avoided in favor of the notions of 'effective p-simulation', and 'effective exponential separation', which are the standard notions relative to a specific reduction.

4 Prover / Delayer Game Upper Bounds for Tree Resolution

In this paper [HU06b] we describe a useful application for the Prover / Delayer Game given in [BSIW04]. The Prover / Delayer Game is an adversarial game between two players, the 'Prover', and the 'Delayer'. The game is played on an unsatisfiable CNF formula F. The point of the game is for the Prover to falsify some initial clause of F, thereby falsifying the formula. Since the formula is unsatisfiable, this is inevitable. Roughly speaking, the Delayer's goal is to delay the falsification of the formula for as long as possible.

The game proceeds in rounds. Each round starts with the Prover choosing a variable, and asking the Delayer what the value of that variable is. The Delayer can give one of three answers:

- True
- False
- You Choose

If the Delayer says 'You Choose', then the Prover gets to decide the value of that variable. In addition, every time 'You Choose' is said, the Delayer wins one point. This is the only way in which points can be scored. The game finishes when some clause has been falsified. The real goal of the game is not actually to prove or delay; rather, the Delayer's aim is to win as many points as possible, while the Prover's aim is to make sure that the Delayer wins as few as possible. This leads us to the following definition:

Definition 4.1. *The Prover / Delayer number of an unsatisfiable formula F, denoted $PD(F)$, be the maximum number of points that the Delayer can score on F with the Prover playing with an optimal strategy.*

The literature contains some interesting uses for this value. For example, in [BSIW04], the authors show that lower bounds on $PD(F)$ translate into lower bounds for the size of T-RES proofs for F. More specifically,

Theorem 4.2. *If the Delayer has a strategy guaranteed to win $> k$ points on F, then every* DPLL *tree for F has size $> 2^k$.*

This result simplifies T-RES lower bounds, since it allows us to make arguments about the game, which are much simpler than arguments about formulas.

Another use for $PD(F)$ is to prove bounds on the amount of clause space required to compute a T-RES proof. Informally, the tree clause space of F, denoted $TCS(F)$ is the minimum number of clauses that must be simultaneously kept in memory at any time while computing a T-RES proof of F. A more precise characterization is given below in Definition 6.4. In [ET03], the authors give both upper and lower bounds for $TCS(F)$ using $PD(F)$:

Theorem 4.3 ([ET03]). *For any unsatisfiable formula F, $TCS(F) = PD(F) + 1$.*

We extend these results by showing that an upper bound on $PD(F)$ also translates into an upper bound for the size of a T-RES proof for F:

Theorem 4.4. *If the Prover has a strategy limiting the Delayer to at most k points playing on formula F which contains $\leq n$ distinct variables, then F has a* DPLL *tree of size $O(n^k)$.*

We conclude this paper by providing some examples of how the Prover / Delayer Game can be used to simplify T-RES upper bound proofs. The first example shows how the Prover / Delayer Game can be used to simplify Theorem 3.1 from the dangerous reductions research above. The second example shows that there is a translation of the $G_{\frac{n}{2},\frac{n}{2}}$ formulas which give the NHPS exponential lower bounds such that the resulting formula has polynomial T-RES upper bounds, thereby proving an effective separation between the two proof systems:

Theorem 4.5. T-RES *proofs for the unsatisfiability of $H(G_{\frac{n}{2},\frac{n}{2}})_{T,O,1,F}$ formulas have $O(n^3)$ size upper bounds, where n is the number of distinct variables contained in the formulas.*

These examples further illustrate the relationship between the different areas of research.

5 Proof Complexity of Intuitionistic Propositional Logic

In this paper [HU06a] we explore the proof complexity of Intuitionistic Logic. Since this is our most important and complicated result, we shall describe it in greater detail than the other three papers. The following is an overview of the paper with proofs and minor details omitted:

5.1 Introduction

Intuitionistic Propositional Logic (IPL) is perhaps the best-studied non-classical logic. It contains all of the standard connectives $\wedge, \vee, \supset, \neg$, but uses non-classical semantics that rely on Kripke models [TD88]. An intuitionistic formula is considered to be intuitionistically valid if it is true in all possible models. A well-known paper by Statman [Sta79] shows via a very natural reduction from QBF that the problem of determining whether a formula is intuitionistically valid is $\mathcal{PSPACE}$-Complete.

IPL was born out of the desire to enforce the requirement of constructivism in mathematics [TD88]. The following classical tautologies are not intuitionistically valid, since it is possible to build Kripke countermodels which falsify them:

- $A \vee \neg A$
- $(A \supset B) \supset (\neg A \vee B)$ (but $(\neg A \vee B) \supset (A \supset B)$ is intuitionistically valid).
- $\neg(A \wedge B) \supset (\neg A \vee \neg B)$ (but all other forms of DeMorgan's Laws are intuitionistically valid).
- $\neg\neg A \supset A$ (but $A \supset \neg\neg A$ is intuitionistically valid).

The last non-tautology, $\neg\neg A \supset A$, gives insight into why IPL is associated with constructivism. Without this inference pattern, the standard proof-by-contradiction technique where we assume $\neg A$, derive a contradiction, conclude $\neg\neg A$, and eliminate the double negation to conclude A becomes impossible. In effect, all IPL proofs must be constructive.

Since IPL is hamstrung by its inability to take advantage of the standard proof-by-contradiction technique available to Classical logic, it is widely believed to be weaker in the sense that there are probably formulas which have short Classical proofs, but no short Intuitionistic proofs. From a complexity-theoretic perspective, IPL is in some regards more tractable than Classical Logic. For example, it has feasible interpolation [Pud99, BP01]. It is consequently reasonable to assume that IPL lower bounds should be easier to prove than for Classical Logic. The main theorem of this paper is a partial result working towards this goal. Rather than showing that IPL is weak in an absolute sense, we show it to be weak relative to Statman's translation. More specifically, we show that unless $\mathcal{NP} = co\mathcal{NP}$, Statman's reduction from QBF to IPL cannot even translate trivial classical instances of the law of excluded middle into intuitionistic formulas which have polynomially-bounded proofs. Since Statman's translation is the obvious and natural reduction to use, this result shows that if a more feasible reduction exists, then it must be complicated. An immediate implication of this result is that unless $\mathcal{NP} = co\mathcal{NP}$, there is a superpolynomial separation between an extended form of Gentzen's system LK and an extended form of his system LJ.

5.2 The System LJ

For the purposes of this paper, we deal with a sequent calculus for IPL in the style of Gentzen [Sza69]. We use capital letters $A, B, C, ...$ to denote complex formulas, lower-case letters $x, y, z, ...$ to denote atomic formulas, and Greek letters $\Gamma, \Delta, \Theta, ...$ to denote sets.

The proof system that we use is essentially the same as Gentzen's system LJ. Our proofs are tree-like, meaning that each sequent can be an input for at most one inference rule. Furthermore, each sequent has singular right side; that is, there is at most one formula on the right-hand side of each sequent. In addition, all axioms are of the form $x \mapsto x$, where x is an atomic sentence letter. LJ is formulated as follows:

axiom:

$$\frac{}{x \mapsto x} \qquad \text{where } x \text{ is atomic}$$

weakening:

$$\textbf{left} \quad \frac{\Gamma \mapsto A}{B, \Gamma \mapsto A} \qquad \text{and} \qquad \textbf{right} \quad \frac{\Gamma \mapsto}{\Gamma \mapsto B}$$

exchange:

$$\textbf{left} \quad \frac{\Gamma_1, A, B, \Gamma_2 \mapsto C}{\Gamma_1, B, A, \Gamma_2 \mapsto C}$$

contraction:

$$\textbf{left} \quad \frac{\Gamma_1, A, A, \Gamma_2 \mapsto B}{\Gamma_1, A, \Gamma_2 \mapsto B}$$

cut:

$$\frac{\Gamma \mapsto A \quad A, \Delta \mapsto B}{\Gamma, \Delta \mapsto B}$$

$\perp$ introduction:

$$\textbf{left}\ \frac{}{\perp \mapsto} \qquad \text{and} \qquad \textbf{right}\ \frac{\Gamma \mapsto}{\Gamma \mapsto \perp}$$

$\neg$ introduction:

$$\textbf{left}\ \frac{\Gamma \mapsto A}{\neg A, \Gamma \mapsto} \qquad \text{and} \qquad \textbf{right}\ \frac{A, \Gamma \mapsto}{\Gamma \mapsto \neg A}$$

$\vee$ introduction:

$$\textbf{left}\ \frac{A, \Gamma \mapsto C \quad B, \Delta \mapsto C}{A \vee B, \Gamma, \Delta \mapsto C} \qquad \text{and} \qquad \textbf{right}\ \frac{\Gamma \mapsto A}{\Gamma \mapsto A \vee B} \ \text{ as well as } \ \frac{\Gamma \mapsto A}{\Gamma \mapsto B \vee A}$$

$\wedge$ introduction:

$$\textbf{left}\ \frac{A, B, \Gamma \mapsto C}{A \wedge B, \Gamma \mapsto C} \qquad \text{and} \qquad \textbf{right}\ \frac{\Gamma \mapsto A \quad \Delta \mapsto B}{\Gamma, \Delta \mapsto A \wedge B}$$

$\supset$ introduction:

$$\textbf{left}\ \frac{\Gamma \mapsto A \quad B, \Delta \mapsto C}{A \supset B, \Gamma, \Delta \mapsto C} \qquad \text{and} \qquad \textbf{right}\ \frac{A, \Gamma \mapsto B}{\Gamma \mapsto A \supset B}$$

One important point of note is that if you remove the requirement of having only a single formula on the right-hand side of a sequent, then the system LJ becomes the system LK for Classical Logic [Sza69]. In other words, every LJ proof is also an LK proof.

5.3 Statman's Translation & LJ$[\vec{E_S}]$

In this section we define an augmented form of LJ that includes extension axioms. We then review Statman's reduction from QBF to IPL. Next we define a specific form of LJ augmented with Statman's extension axioms.

5.3.1 Extended LJ

Definition 5.1. *We shall use* LJ$[\vec{E}]$ *to denote an augmented form of* LJ *where* $\vec{E} = E_1, ..., E_n$*, and each* E_i *contains the pair of sequents* $A \circ B \mapsto y_{A \circ B}$ *and* $y_{A \circ B} \mapsto A \circ B$ *with the restriction that* $A \circ B$ *must be a complex formula, and* $y_{A \circ B}$ *is an atom not appearing in* $E_1, ..., E_{i-1}$*. We allow the sequents in* $\vec{E}$ *to be used as axioms in any* LJ$[\vec{E}]$ *proof. We refer to* LJ$[\vec{E}]$ *as an Extended Sequent Calculus, and the two sequents in each* E_i *are referred to as extension axioms. This same augmentation can be defined for the system* LK*.*

5.3.2 Statman's Translation

In his paper showing that IPL is $\mathcal{PSPACE}$-Complete [Sta79], Statman proves this result by providing a reduction from QBF to IPL. This translation is very important to our main result, and proceeds as follows:

1. As input take a QBF formula $F_{QBF} = Q_n x_n, ..., Q_1 x_1 B_0$, where B_0 is a quantifier-free prenex formula, and each $Q_i = \forall$ or $\exists$.

2. Let $y_0, ..., y_n$ be entirely new variables not appearing in F_{QBF}. These are extension variables that are necessary to keep our translation from growing exponentially.

3. Define a series of $B_i^{\checkmark}$ formulas as follows (note that these are not extension variables, but rather are just shorthand for the purposes of this reduction; similarly, $A \leftrightarrow B$ is shorthand for $(A \supset B) \wedge (B \supset A)$):

   - $B_0^{\checkmark} = \neg\neg B_0 \leftrightarrow y_0$
   - $B_k^{\checkmark} = ((x_k \vee \neg x_k) \supset y_{k-1}) \leftrightarrow y_k$ if $Q_k = \forall$
   - $B_k^{\checkmark} = ((x_k \supset y_{k-1}) \vee (\neg x_k \supset y_{k-1})) \leftrightarrow y_k$ if $Q_k = \exists$

4. Output $F_{IPL} = B_0^{\checkmark} \supset (B_1^{\checkmark} \supset (...(B_{n-1}^{\checkmark} \supset (B_n^{\checkmark} \supset y_n))...))$

5.3.3 Proof of Correctness

A full proof of correctness showing that F_{QBF} is true if and only if F_{IPL} is intuitionistically valid does not add to the understanding of this paper, so we shall refer the interested reader to [Sta79]. However, part of the proof is of interest. Specifically, Statman's proof of correctness implicitly proves that a cut-free tree-like sequent calculus formulation of LJ p-simulates Boolean Truth Trees (BTT), a brute-force tree-like system for QBF.

Boolean Truth Trees

The BTT proof system is defined as follows:

- Every BTT proof is a tree in which every node contains a fully-quantified QBF formula. These formulas may contain constants 0 and 1. The root contains the formula which is to be proven true, and each leaf contains a formula in which all variables have been replaced by constants, and it consequently evaluates to 1, thereby showing that the formula is true. (It is most convenient to picture the proof with the root at the top and the leaves at the bottom).
- Every internal node v in the proof tree has either one or two children. If the outermost quantifier in the formula F at v is $\exists x$, then v has one child containing F', which is F with $\exists x$ removed, and every instance of x replaced by an appropriate constant such that F' is true. If the outermost quantifier in the formula F at v is $\forall x$, then v has two children, one containing F_0, where all instances of x have been replaced by 0, and the other containing F_1, where all instances of x have been replaced by the value 1.

It is easy to see that BTT is sound; if a BTT tree can be built for a QBF formula F, then F is true. It is also easy to see that BTT is complete; every true QBF formula has a BTT proof. This is a brute-force proof system, because it is extremely inefficient; the size of a BTT proof is exponential in the number of $\forall$ quantifiers contained in the formula to be proved true.

BTT Example

Consider the formula $\forall x \exists y \forall z (((\neg x \wedge \neg y) \vee z) \vee ((x \wedge y) \vee z))$. The following BTT proof shows that it is a true QBF formula:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{((1 \wedge 1) \vee 0) \vee ((0 \wedge 0) \vee 0) = 1}[0/z]
\quad
\dfrac{}{((1 \wedge 1) \vee 1) \vee ((0 \wedge 0) \vee 1) = 1}[1/z]
}{\forall z(((1 \wedge 1) \vee z) \vee ((0 \wedge 0) \vee z))}
}{\exists y \forall z(((1 \wedge \neg y) \vee z) \vee ((0 \wedge y) \vee z))}[0/y]
}{\forall x \exists y \forall z(((\neg x \wedge \neg y) \vee z) \vee ((x \wedge y) \vee z))}[0/x]
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{((0 \wedge 0) \vee 0) \vee ((1 \wedge 1) \vee 0) = 1}[0/z]
\quad
\dfrac{}{((0 \wedge 0) \vee 1) \vee ((1 \wedge 1) \vee 1) = 1}[1/z]
}{\forall z(((0 \wedge 0) \vee z) \vee ((1 \wedge 1) \vee z))}
}{\exists y \forall z(((0 \wedge \neg y) \vee z) \vee ((1 \wedge y) \vee z))}[1/y]
}{\forall x \exists y \forall z(((\neg x \wedge \neg y) \vee z) \vee ((x \wedge y) \vee z))}[1/x]
$$

P-Simulation Result

We shall now make explicit the implicit p-simulation result in Statman's proof of correctness; this provides a deeper insight into the mechanisms at play underlying the reduction.

If we forget about the extension variables, then Statman's translation converts the formula F_{QBF} to an intermediate formula A^+ as follows:

- $B_0^+ = \neg\neg B_0$
- $B_k^+ = (x_k \vee \neg x_k) \supset B_{k-1}^+$ if $Q_k = \forall$
- $B_k^+ = (x_k \supset B_{k-1}^+) \vee (\neg x_k \supset B_{k-1}^+)$ if $Q_k = \exists$

Note that $A^+ = B_n^+$. Statman's proof of correctness proceeds in two parts. He first shows that F_{QBF} is true if and only if A^+ is intuitionistically provable. Next he proves that A^+ is intuitionistically provable if and only if F_{IPL} is. We are only interested in the forward direction of the first part; the implicit p-simulation result is given by the proof that F_{QBF} being true implies that A^+ is intuitionistically provable.

The overall idea of the simulation is to take the brute-force BTT proof tree T_{BTT} for F_{QBF}, and by way of very local transformations, build an exactly analogous cut-free tree-like LJ proof T_{LJ}.

Theorem 5.2. *Cut-free tree-like* LJ *p-simulates* BTT.

In effect, the obvious brute-force BTT proof translates into a brute-force cut-free, tree-like LJ proof.

5.3.4 The System $\mathsf{LJ}[\vec{E_S}]$

Given that we have the definition of $\mathsf{LJ}[\vec{E}]$ as well as the details of Statman's translation (the full version, with extension variables), we may now define the system $\mathsf{LJ}[\vec{E_S}]$. This is the system which we will use for our main theorem in Section 5.6.

Definition 5.3. *We define* $\mathsf{LJ}[\vec{E_S}]$ *to be* LJ *augmented with Statman's extension axioms for a formula* $F_{QBF} = Q_n x_n, ..., Q_1 x_1 B_0$*, where* B_0 *is a quantifier-free prenex formula, and each* $Q_i = \forall$ *or* $\exists$*. More precisely,* $\vec{E_S} = E_0, ..., E_n$*, where*

- $E_0 = \{\neg\neg B_0 \mapsto y_0, y_0 \mapsto \neg\neg B_0\}$
- $E_k = \{(x_k \vee \neg x_k) \supset y_{k-1} \mapsto y_k, y_k \mapsto (x_k \vee \neg x_k) \supset y_{k-1}\}$ *if* $Q_k = \forall$*, and*
- $E_k = \{(x_k \supset y_{k-1}) \vee (\neg x_k \supset y_{k-1}) \mapsto y_k, y_k \mapsto (x_k \supset y_{k-1}) \vee (\neg x_k \supset y_{k-1})\}$ *if* $Q_k = \exists$

5.4 Cut-Elimination

In this section we extend the cut-elimination technique developed by Buss and Pudlák in [BP01] (which itself was adapted from [BM99]) so that it holds for any system $\mathsf{LJ}[\vec{E}]$.

Definition 5.4. *The closure of a proof* P*, denoted* $cl(P)$*, is the smallest set of sequents which includes the sequents of* P *and is closed under both weakening and cut.*

Definition 5.5. *A principal cut is a cut in which at least one of its two input sequents is an extension axiom.*

Theorem 5.6 (Cut-Elimination). *For any proof* P *in any system* $\mathsf{LJ}[\vec{E}]$*, it is possible to eliminate all non-principal cuts from* P *to produce a pseudo-cut-free proof* P' *such that* $cl(P') \subseteq cl(P)$*.*

5.5 The Proof Closure Property

Informally, the Proof Closure Property guarantees that any sequent in the closure of an $\mathsf{LJ}[\vec{E_S}]$ proof P can be derived via a polynomial number of cuts and weakenings from the sequents in P. This property is strongly related to Resolution involving Horn clauses.

Lemma 5.7 (Proof Closure Property). *Let P be a size-N $\mathsf{LJ}[\vec{E_S}]$ proof, where N is the number of bits required to encode P. If $\Gamma \mapsto A \in cl(P)$ then there exists a size-$O(N^2)$ tree-like and a size-$O(N)$ DAG-like $\mathsf{LJ}[\vec{E_S}]$-proof of $\Gamma \mapsto A$.*

The Proof Closure Property will be used together with the Cut-Elimination Theorem. The overall idea is to apply a the Cut-Elimination Theorem, which blows up our proof exponentially, but allows us to prove things that we otherwise could not have proved due to the presence of cut. Once we have what we want in the closure of our proof, we apply the Proof Closure property to extract what we showed to be in the closure with a polynomially-bounded proof.

5.6 Main Result

We are now ready to prove our main result, which relies heavily on the Cut-Elimination Theorem as well as the Proof Closure Property.

Theorem 5.8 (Main Theorem). *Let $F_{Prop} = A(x_1, ...x_n)$ be any arbitrary classical propositional tautology containing n distinct variables, and consider the formula $F'_{Prop} = A(x_1, ...x_n) \vee \neg A(x_1, ...x_n)$. Let F_{QBF} be the prenex QBF translation of F'_{Prop} where each quantifier is $\forall$, and let F_{IPL} be Statman's translation of F_{QBF}. If there exists a size-N DAG-like $\mathsf{LJ}[\vec{E_S}]$ proof of F_{IPL}, where N is the number of bits required to encode F_{IPL}, then F_{Prop} has a DAG-like classical $\mathsf{LK}[\vec{E_S}]$ proof of size-$O(N^4)$.*

5.7 Immediate Implications

Our main result implies that if Statman's translation maps trivial instances of the law of excluded middle in QBF to IPL instances which have polynomially-bounded proof complexity, then $\mathcal{NP} = co\mathcal{NP}$. This leads to some immediate corollaries:

5.7.1 Conditional Separation Between Classical & Intuitionistic Logic

The most important implication of our main result is that under the assumption that $\mathcal{NP} \neq co\mathcal{NP}$, there is a superpolynomial separation between $\mathsf{LK}[\vec{E_S}]$ and $\mathsf{LJ}[\vec{E_S}]$:

Corollary 5.9. *The F_{IPL} formulas have size-$O(n)$ $\mathsf{LK}[\vec{E_S}]$ proofs, where n is the number of distinct variables in F_{IPL}, but unless $\mathcal{NP} = co\mathcal{NP}$, there are no $\mathsf{LJ}[\vec{E_S}]$ proofs of F_{IPL} with polynomial upper bounds.*

An interesting point of note is how powerful $\mathsf{LJ}[\vec{E_S}]$ really is. LJ and LK are very similar, and $\mathsf{LJ}[\vec{E_S}]$ is strictly stronger than LJ. But LK with cut is p-equivalent to any Frege system, and $\mathsf{LJ}[\vec{E_S}]$ has cut. All of these things together suggest that it is a fairly powerful system.

5.7.2 Dangerous Reductions

The second implication of our main result is that Statman's reduction is 'dangerous' in the sense that unless $\mathcal{NP} = co\mathcal{NP}$, it translates some trivial formulas from QBF to very difficult formulas in IPL. In order to formalize this notion, we take the formal definitions of what constitute dangerous and safe proof complexity reductions from [HH06].

Corollary 5.10. *Unless $\mathcal{NP} = co\mathcal{NP}$, Statman's reduction is (α,LJ[$\vec{E_S}$])-Explosive for every proof system α for QBF which has polynomially-bounded proofs for every prenex instance of the law of excluded middle.*

6 Proposed Future Research

There are a few different avenues for further research. One potential direction is to continue research in the areas defined by the four papers described above.

6.1 Continuing The Work Already Done

6.1.1 Further Research Involving The NHPS

As already mentioned, one of the motivations for developing the NHPS was to explore lower bounds for graph algorithms that take advantage of local properties. An interesting avenue of research would be to do just that. Is it possible to establish a framework more general than the NHPS which allows us to translate lower bounds for known proof systems into lower bounds for graph algorithms?

6.1.2 Further Research Involving Dangerous Reductions

Although we gave an example of a provably dangerous reduction, we still lack a meaningful way of characterizing such reductions so that researchers can predict which reductions will be dangerous and therefore avoid them. Such a characterization would be very useful and interesting.

6.1.3 Further Research Involving IPL

Our work in Intuitionistic Logic has highlighted a few interesting open problems, all of which appear to be fairly difficult.

The first open problem is to prove superpolynomial LJ lower bounds which do not depend on any assumptions.

This first problem probably subsumes the second open problem, namely showing a superpolynomial or even exponential separation between LK and LJ which does not depend on any assumptions. Another way of phrasing this is as follows: Statman's translation is probably explosive, but perhaps some other safe translation is possible. Does a safe translation exist, or are all reductions from QBF to IPL explosive? Since our conditional separation seems to show that IPL is weaker than QBF, it is reasonable to conjecture that no safe translation exists. However, if a safe translation does exist, then it almost certainly is not as natural or intuitive (no pun intended) as Statman's.

The third open problem has to do with the reduction in the reverse direction: Does a safe translation from IPL to QBF exist? Finding such a translation appears to be difficult, but it is reasonable to conjecture that the answer is yes. One possible way to prove the opposite reduction is to further work with the results from [Lad77], in which the author proves the $\mathcal{PSPACE}$-Completeness of various systems of Modal Logic, all of which use the same Kripke Semantics as Intuitionistic Logic.

6.2 A New Research Area: Resolution Size vs. Width vs. Space

Another potential direction is to work in a new area that is still closely related to the work that we have been doing. There has been a fair amount of work done lately on the relationship between different measures of complexity for RES proofs. The three most studied measures are refutation size, width, and various different forms of space. The definitions of size and width can be formalized using a fairly simple definition of what constitutes a RES proof, whereas the definition of space requires a slightly more complicated formulation.

Definition 6.1 (RES Proof). *If F is an unsatisfiable set of clauses, then the sequence of clauses $\pi = C_1, C_2, ..., C_k$ is a* RES *refutation of F if the clauses in π meet the following requirements:*

1. *$C_k = \emptyset$ (the empty clause).*
2. *Each C_i appears in F (ie. is an input, or initial clause) or follows from two previous clauses in π by the resolution rule.*

If the graph underlying the structure of π is a tree (ie. each clause in π is part of at most one application of the resolution rule), then the proof is said to be a Tree-Like T-RES *refutation. Otherwise it is said to be DAG-Like.*

This allows us to define size and width:

Definition 6.2 (Size & Width). *Let F and π be as above in Definition 6.1. If a* RES *proof π contains k clauses, then it is said to have size k. The width of a clause C refers to how many literals it contains. The width of a* RES *refutation π is equal to the width of its widest clause.*

The definitions of the various different measures of space that we shall be discussing requires an alternative definition of RES proof which requires the notion of configuration:

Definition 6.3 (Configuration-Style RES Proof). *A configuration $\mathbb{C}$ is a set of clauses. If F is an unsatisfiable set of clauses, then the sequence of configurations $\pi = \mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_x$ is a* RES *refutation of F if $\mathbb{C}_1 \subseteq F$, $\emptyset \in \mathbb{C}_x$, and for each $i < x$, $\mathbb{C}_{i+1}$ is obtained from $\mathbb{C}_i$ by one of the following rules:*

1. *deleting one or more of its clauses,*
2. *adding the resolvent of two clauses of $\mathbb{C}_i$,*
3. *adding one or more of the clauses of F (initial clauses).*

In addition, π is said to be Tree-Like if we replace rule 2 with the following:

2. *adding the resolvent of two clauses of $\mathbb{C}_i$* ***and deleting the parent clauses***

This leads us to our different definitions of space. Intuitively, space refers to the amount of memory that is required in order to compute π. Note that for each of these measures, it is important to distinguish between space for RES and clause space for T-RES.

Definition 6.4 (Clause Space). *Let F be an unsatisfiable set of clauses and π be a* RES *(*T-RES*) refutation of F as defined above in Definition 6.3. If each configuration $\mathbb{C}_i$ in π contains at most s clauses, then π is a* RES *(*T-RES*) refutation bounded by (tree) clause space s.*

Definition 6.5 (Variable Space). *Let F be an unsatisfiable set of clauses and π be a* RES *(*T-RES*) refutation of F as defined above in Definition 6.3. If each configuration $\mathbb{C}_i$ in π contains at most s distinct variables, then we say that π is a* RES *(*T-RES*) refutation bounded by (tree) variable space s.*

Definition 6.6 (Total Space). *Let F be an unsatisfiable set of clauses and π be a* RES *(*T-RES*) refutation of F as defined above in Definition 6.3. If for each configuration $\mathbb{C}_i$ in π, $\sum_{C \in \mathbb{C}_i} Width(C) \leq s$, then π is a* RES *(*T-RES*) refutation bounded by (tree) total space s.*

We shall abbreviate clause space, variable space, and total space of a formula F as $CS(F)$, $VS(F)$, and $TS(F)$ respectively. In the Tree-Like cases, we will abbreviate them as $TCS(F)$, $TVS(F)$, and $TTS(F)$, respectively. For each of these space measures, the $(T)CS$ / $(T)VS$ / $(T)TS$ of a formula F refers to the minimal $(T)CS$ / $(T)VS$ / $(T)TS$ of any RES proof of F.

It is easy to see that for all F, $CS(F) \leq TS(F)$, and $VS(F) \leq TS(F)$. The same holds for the Tree-Like cases. $VS(F)$ and $CS(F)$ are incomparable, since each of these quantities can be made larger than the other by choosing an appropriate example for F.

6.3 Major Results

6.3.1 Space & Games

The investigation of RES space has always been associated closely with the well-known Pebbling Game and Pebbling Number of a graph. There are different versions of Pebbling Games, but in its simplest form, the player tries to 'pebble' a DAG $G = (V, E)$ in which every vertex has in-degree at most 2. The leaves of G have in-degree 0 and are referred to as 'Source' nodes. A single vertex in G is referred to as a 'Target' node, and has out-degree 0. All non-target nodes can have arbitrary out-degree, except of course in the case where G is a tree, in which case the maximum out-degree is 1. The rules of the pebbling game are as follows:

- At any point, the player may put a pebble on a source node.
- At any point, the player may remove a pebble.
- If both of a node's children have pebbles on them, then the player may put a pebble on it.
- The game ends once the player puts a pebble on the target node.

Definition 6.7. *The Pebbling Number of G, denoted $Peb(G)$ is the minimum number of pebbles that the player must be given in order for it to be possible to pebble the target node without violating any of the above rules.*

The pebbling game is related closely to RES clause space. Let F be any arbitrary unsatisfiable formula, let π be a RES refutation of F, and let G be the DAG underlying the structure of π. It is not hard to see that the clause space required to compute π is exactly equal to $Peb(G)$. More specifically, the clause space of a formula F is equal to the pebbling number of the DAG with the smallest pebbling number of all DAGs underlying valid RES refutations of F. This shows that both general clause space as well as tree clause space are related closely to the Pebbling Game.

Tree clause space is also related closely to our work with the Prover / Delayer Game. As already mentioned above in Theorem 4.3, the results in [ET03] are the definitive work relating the Prover / Delayer number $PD(F)$ of an unsatisfiable CNF formula to its tree clause space $TCS(F)$ by showing that $TCS(F) = PD(F) + 1$. In other words, the Prover / Delayer Game perfectly captures the notion of clause space for T-RES.

6.3.2 Tree Space & Size

In addition to pioneering the field and proving many other space results, Esteban & Torán also showed that an upper bound on tree clause space gives an upper bound on the size of T-RES proofs:

Theorem 6.8 ([ET01]). *Let F be an unsatisfiable formula on n distinct variables. If F has a* T-RES *refutation requiring tree clause space $s = TCS(F)$, then it has a* T-RES *refutation of size $\binom{n+s}{s}$.*

6.3.3 Space & Width

The definitive paper relating space and width is [AD03]. In it the authors prove that for unsatisfiable k-CNF formulas, space is an upper bound on width:

Theorem 6.9 ([AD03]). *Let F be any unsatisfiable k-CNF formula. Then $CS(F) \geq w(F \vdash \emptyset) - k$, where $CS(F)$ is the minimal clause space of refuting F, and $w(F \vdash \emptyset)$ is the minimal width of refuting F.*

This is a very powerful result, because it can be used to derive space lower bounds for all formulas for which width lower bounds are known.

6.3.4 Width & Size

The relationship between width & size is very important because it shows that lower bounds for width imply lower bounds for size. This simplifies proofs for size lower bounds by allowing us to focus on lower bounds for width. The main result linking width and size was [BSW01], but their results have since been generalized in [Urq06]:

Theorem 6.10 ([BSW01]). *Let Σ be a contradictory set of clauses with an underlying set of variables V, $S(\Sigma)$ the minimum size of a* RES *refutation of Σ, $w(\Sigma)$ the maximum number of literals in a clause in Σ, and $w(\Sigma \vdash \emptyset)$ the minimal width of refuting Σ. Then*

$$S(\Sigma) = exp\left(\Omega\left(\frac{(w(\Sigma \vdash \emptyset) - w(\Sigma))^2}{|V|}\right)\right)$$

6.3.5 Other Relationships

Another important result is [BS02], in which the author proves a number of interesting tradeoff results. For example, he provides families of formulas for which

- there are linear size T-RES proofs and constant width T-RES proofs, but there is no T-RES proof that has both small width and small size.
- there are RES proofs with constant clause space, and RES proofs with constant width, but there is no RES proof that has both small width and small clause space.
- there are linear size RES proofs that also have constant width, but there is no RES proof that has both small clause space and small size.

These results rely on the Pebbling Formulas, unsatisfiable formulas which are based on the Pebbling Game described above.

6.4 Open Problems

This area of research has some interesting open problems. One such problem is to prove a separation between clause space for T-RES and clause space for RES. The current best result seems to be the linear separation in [ET03]. The authors show that for the Pebbling Formulas of the complete binary tree of height h, the general clause space is at most $2h/3 + 3$, but the tree clause space is at least $h - 2$. Surely this separation can be improved. In addition, for the same formulas, the lower bound for general clause space given in [Nor05] is $\Omega(\sqrt{h})$. This also leaves quite a gap for improvement.

Other important open problems come from [Nor05], one of the most interesting recent papers in the area. In it the author investigates the $Peb^d_{T_h}$ formulas, which are simply the pebbling formulas using d colours built on the complete binary tree. The main result is a separation between width and space showing that the $Peb^d_{T_h}$ formulas have RES proofs with constant width, but have no RES proofs with clause space less than $\Omega(\sqrt{h})$. This result leads to the following interesting conjectures / open problems. Note that *BW-Peb* is the Black-White Pebbling Number of a Graph; the Black-White Pebbling game is a simple variant of the Pebbling Game described above.

Conjecture 6.11. *For any constant* $d \geq 2, CS(Peb^d_{T_h} \vdash \emptyset) = \Theta(BW\text{-}Peb(T_h)) = \Theta(h)$

Conjecture 6.12. *For any constant* $d \geq 2$ *and for* G *an arbitrary DAG with a unique target and with all vertices having in-degree* 0 *or* 2, $C(Peb^d_G \vdash \emptyset) = \Omega(BW\text{-}Peb(G))$

7 Acknowledgements

- First and foremost I'd like to thank my supervisor Alasdair for all of the time spent teaching me and guiding my research.
- I'd also like to thank Toni for her meetings with me and input into my research.
- Philipp also deserves credit for our joint-work on Dangerous Reductions.
- Thanks to Tanya for her continued help and support.
- And finally thank you to NSERC and the U of T Dept. of Computer Science for their support.

References

[AD03] A. Atserias and V. Dalmau. A Combinatorial Characterization of Resolution Width. *Proc. of the 18th IEEE Conference on Computational Complexity*, 2003.

[BM99] S. R. Buss and G. Mints. The Complexity of the Disjunction and Existential Properties in Intuitionistic Logic. *Annals of Pure and Applied Logic*, 99:93 – 104, 1999.

[BP01] S. R. Buss and P. Pudlák. On The Computational Content of Intuitionistic Propositional Proofs. *Annals of Pure and Applied Logic*, 109:49 – 64, 2001.

[BS02] E. Ben-Sasson. Size Space Tradeoffs For Resolution. *Proceedings of the 34th ACM Symposium on the Theory of Computing*, pages 457 – 464, 2002.

[BSIW04] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson. Near Optimal Separation of Tree-like and General Resolution. *Combinatorica*, Vol. 24, Issue 4:585 – 604, 2004.

[BSW01] E. Ben-Sasson and A. Wigderson. Short Proofs are Narrow -Resolution Made simple. *Journal of the Association for Computing Machinery*, 48:149–169, 2001. Preliminary version: Proceedings of the 31st Annual ACM Symposium on Theory of Computing, 1999, pp. 517-526.

[ET01] J. Esteban and J. Torán. Space Bounds for Resolution. *Information and Computation*, 171:84 – 97, 2001.

[ET03] J. Esteban and J. Torán. A Combinatorial Characterization of Treelike Resolution Space. *Information Processing Letters*, 87:295–300, 2003.

[Her06] A. Hertel. A Non-Hamiltonicity Proof System. Unpublished Manuscript, 2006.

[HH06] A. Hertel and P. Hertel. Genome Assembly Algorithms for New Sequencing Technologies. http://www.cs.toronto.edu/~ahertel/WebPageFiles/Papers/GenomeSequencingProject.pss, 2006.

[HU06a] A. Hertel and A. Urquhart. Proof Complexity of Intuitionistic Propositional Logic. Unpublished Manuscript, 2006.

[HU06b] A. Hertel and A. Urquhart. Prover / Delayer Game Upper Bounds For Tree Resolution. Unpublished Manuscript, 2006.

[KMS97] H. Kautz, D. McAllester, and B. Selman. Ten Challenges in Propositional Reasoning and Search. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997.

[KS03] H. Kautz and B. Selman. Ten Challenges Redux: Recent Progress in Propositional Reasoning and Search. *Ninth International Conference on Principles and Practice of Constraint Programming*, 2003.

[Lad77] R. Ladner. The Computational Complexity of Provability in Systems of Modal Propositional Logic. *SIAM Journal of Computing*, Vol. 6, No. 3:467 – 480, 1977.

[Nor05] J. Nordström. Narrow Proofs May Be Spacious: Separating Space and Width in Resolution. *Electronic Colloquium on Computational Complexity*, 66, 2005.

[Pud99] P. Pudlák. On The Complexity of Propositional Calculus, Sets and Proofs. In *Logic Colloquium '97*, pages 197 – 218. Cambridge University Press, 1999.

[Sta79] R. Statman. Intuitionistic Propositional Logic is Polynomial-Space Complete. *Theoretical Computer Science*, 9:67 – 72, 1979.

[Sza69] M.E. Szabo. *The Collected Papers of Gerhard Gentzen.* North-Holland Publishing Company, Amsterdam, 1969.

[TD88] A.S. Troelstra and D. Van Dalen. *Constructivism in Mathematics, An Introduction.* Elsevier Science Publishers B.V., Amsterdam, 1988.

[Urq06] A. Urquhart. Width Versus Size in Resolution Proofs. *Proceedings of the 3rd Annual Conference on Theory and Applications of Models of Computation*, pages 79 – 88, 2006.