Proof Complexity of Circumscription
Leroy Chew
MSc Mathematics and Computer Science
2012-2013

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)________________________
0.1 Summary

In this project I have used known propositional calculi such as $LK$, $CIRC$ and $MLK$ as defined in the literature. I have investigated the sequent calculus $CIRC$, which is a complete calculus for propositional circumscription, and have attempted either to relate its proof complexity to the classical sequent calculus $LK$ or to find an exponential lower bound. After proving a lower bound on the proof complexity of $CIRC$, I have completed a preliminary investigation into alternative calculi including $MLK$ in order to find advantages in terms of proof complexity.

In the process of this project, I have found new results presented as theorems 23, 35 and 37 in this report and have given my own mathematical proofs on each theorem. Some of this work is to be presented at the British Logic Colloquium Postgraduate Day in September 2013. As there is much potential for expanding on the work written in this report, I have given a description for what should be done for future work in this area.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Methodology</td>
<td>1</td>
</tr>
<tr>
<td>Mathematical Objects and Notation</td>
<td>2</td>
</tr>
<tr>
<td>Formal Classical Logic</td>
<td>3</td>
</tr>
<tr>
<td>Propositional Circumscription</td>
<td>4</td>
</tr>
<tr>
<td>Complexity</td>
<td>6</td>
</tr>
<tr>
<td>The Antisequent Calculus</td>
<td>8</td>
</tr>
<tr>
<td>A Sequent Calculus for Circumscription</td>
<td>9</td>
</tr>
<tr>
<td>Lower Bound for Circumscription</td>
<td>9</td>
</tr>
<tr>
<td>Comparisons to Other Logics</td>
<td>11</td>
</tr>
<tr>
<td>Comparisons to MLK</td>
<td>12</td>
</tr>
<tr>
<td>Comparison to Niemelä’s Tableau Calculus</td>
<td>15</td>
</tr>
<tr>
<td>Quantified Propositional Formulae</td>
<td>17</td>
</tr>
<tr>
<td>Conclusions</td>
<td>18</td>
</tr>
<tr>
<td>Evaluation of Results</td>
<td>19</td>
</tr>
<tr>
<td>Further Works</td>
<td>19</td>
</tr>
<tr>
<td>Appendix: Undertaking the Project</td>
<td>21</td>
</tr>
<tr>
<td>Appendix: Using the Work and Help of Others</td>
<td>21</td>
</tr>
<tr>
<td>Appendix: Ethical Issues</td>
<td>22</td>
</tr>
<tr>
<td>Appendix: Extra Proofs</td>
<td>22</td>
</tr>
</tbody>
</table>
1 Introduction

Classical logic allows inference of new propositions from known propositions, but it has a property known as monotonicity that distinguishes it from human reasoning. This is believed to be a drawback for its application to artificial intelligence. Monotonicity is the property that once a proposition has been accepted, no new knowledge can change that acceptance. Monotonicity is very useful for fields like mathematics, but it fails to capture the non-monotonicity of everyday reasoning, where a person may change their mind.

Several non-classical non-monotonic logics have been developed as mathematical formal logics. One of them, circumscription, is a popular non-monotonic logic that works by minimising propositional atoms over the language. The precise meanings of these terms are defined in the main report. The semantic definition was developed by McCarthy in 1980 [20]. A sound and adequate calculi $CIRC$ for circumscription has been developed by Bonatti and Olivetti [6].

This project concerns itself with proof complexity, a branch of theoretical computer science, which is the study of the lengths of proofs by number of characters. There has been recent work on the proof complexity of the other non-monotonic logics, specifically default logic and autoepistemic logic [2] [3]. The findings were that each logic could be split into two cases; credulous and sceptical. The credulous cases were found to be related in terms of proof complexity to the classical calculi, and the sceptical cases were found to have exponential lower bounds for their proof complexities [2] [3]. Circumscription is not typically broken down into sceptical and credulous cases, and there has been no investigation into the proof complexity of $CIRC$.

It is known that propositional circumscription is $\Pi^P_2$-complete (we give a definition of this in definition 15, in the section 6), so we would not expect the shortest proofs to be polynomial length in general, unless the polynomial hierarchy collapses to the first level. Overall bounds for the calculus $CIRC$ by Bonatti and Olivetti had not yet been rigorously ascertained before this project was undertaken. The objective of this project is to prove that it has a general lower bound, and then to compare that lower bound to other results. An extended aim is to investigate the alternatives to $CIRC$ for circumscription theorem proving, in particular the calculus earlier developed by Olivetti; $MLK$ and the Tableau Calculus by Niemela. [22] [6] [21]

Sections 3 to 8 give necessary background information from the literature. Sections 9 and 11 are mainly focused the new results of this project and sections 10 and 12 mainly involve comparisons to other works. The sections 3 to 12 are written using a plural first person subject. The sections 2, 13-15 and appendices A through to C are mainly written in the singular first person although the plural first person subject may be used to refer to myself and my supervisor. Appendix D once again refers to the plural first person.

The minimal objectives of this project were to:

1. Understand and analyse the proof complexity of circumscription.
2. Compare the complexity of theorem proving for circumscription to the complexity of theorem proving in classical logic and in other non-monotonic logics.

Once the minimal objectives were completed, I was given the option of possible enhancement that could be achieved through tasks. The meanings of some of these will be explained in the main report:

1. Find short proofs for the hard examples in $MLK$.
2. Find short proofs for the hard examples using the QPF translation.
3. Investigate whether $MLK$ simulates $CIRC$ for minimal entailment.
4. Investigate whether $MLK$ p-simulates $CIRC$ for minimal entailment.
5. Investigate whether $MLK$ simulates $CIRC$ in general.
6. Investigate whether $MLK$ p-simulates $CIRC$ in general.
7. Investigate the proof complexity of $MLK$.
8. Investigate the proof complexity of the tableau calculus.

2 Methodology

Research is mainly theoretical; I have relied mainly on 3 resources of research.

1. Self directed mathematical research; I have studied mathematics at the university level for four years now and am capable of using mathematical techniques to formulate mathematical arguments and proofs, without the need for external resources. This methodology was followed as verified in VLE blog [8].
2. Meetings with the supervisor; Dr. Beyersdorff has much better knowledge and understanding of complexity, proof complexity and formal logic than I, he has previously proved very similar results to the results I am interested in. He also has a better intuition for possible approaches to take and it is important that I take that advice. The meetings also help generate new shared ideas on the research problems. These meetings can be confirmed by my supervisor.

3. Research papers; my work is concerned with the \textit{CIRC} calculus defined in a 2002 paper [6], which itself is built on work demonstrated in many other papers. Definitions need to be clarified, and it is possible to find papers that give results that give greater insight to the problems, or that lead on to obvious unanswered questions. The evidence that these papers were used can be found in this report where citations are used.

The results that have been looked at mainly are upper bounds and lower bounds. To prove a lower bound an ordered scheme of examples with growing proof size can be found. In order to prove an upper bound, particularly in terms of another proving system, polynomial transformations of the proofs of one system to perhaps another proof in another system or the algorithmic solution of a different problem, can be used. These are the primary strategies in my investigation.

In undertaking the project, the minimal objectives were undertaken first.

3 Mathematical Objects and Notation

Näively, a set is an unordered collection of unrepeatable elements; which are the objects that it contains. Two sets are equal if they contain exactly the same elements. Normally we write sets using "{}" parentheses. The set \{a, b\} is the set that contains element \(a\) and element \(b\). The set \{\} contains no element and we usually denote that by \(\emptyset\). The shorthand for \(a\) is an element of set \(A\)" is denoted by \(a \in A\). We can define sets like this, for example \(\{a \in A\mid P(a)\}\) means the set containing every element in set \(A\) such that \(P(a)\) (which is some statement about element \(a\)) is true for each element \(a\).

Some well known sets are; \(\mathbb{N}\) the set of natural numbers \(\{0,1,2,...\}\), \(\mathbb{Z}\) the set of integers which includes the natural numbers and their negative counterparts, \(\mathbb{Q}\) the set of fractions of the integers and \(\mathbb{R}\) the set real numbers which contains all numbers in between as well.

Sets themselves can be treated as mathematical objects and operations can be defined on them.

\textbf{Definition 1.} – The union of sets \(A\) and \(B\) is the set \(A \cup B = \{a \in A \text{ or } a \in B\}\). The union of a set \(C\) which has elements that are sets is \(\bigcup C = \{z\mid\text{there exists } y \in C \text{ with } z \in y\}\).

– The intersection of sets \(A\) and \(B\) is the set \(A \cap B = \{a \in A \mid a \in B\}\). The intersection of a set \(C\) which has elements that are sets is \(\bigcap C = \{z\mid\text{for every } y \in C \text{, } z \in y\}\).

– When sets \(A\) and \(B\) are pairwise disjoint \(A \cap B = \emptyset\) we can find a disjoint union \(A \cup B = A \cup B\), when we use this notation it indicates the sets are pairwise disjoint.

– If \(A \subseteq B\) (that is \(A\) is a subset of \(B\); \(A\) only contains elements \(B\) contains), then \(B \setminus A = \{b \in B \mid b \notin A\}\).

\textbf{Definition 2.} A (total) function \(f\) from set \(X\) to set \(Y\) (written \(f : X \to Y\)), gives every element \(x\) in \(X\) a single element \(f(x)\) in \(Y\). A partial function \(g\) from set \(X\) to set \(Y\) (written \(f : X \to Y\)) gives some elements \(\text{(although possibly none)}\) \(x\) in \(X\) a single element \(f(x)\) in \(Y\).

\textbf{Definition 3.} Here we define a sequence, in two different ways, an infinite sequence, \((a_i)_{i \geq 0}\), in set \(A\), is a function \(f\) from \(\mathbb{N}\) to \(A\) with \(f(i) = a_i\). A finite sequence (of length \(n+1\)), \((a_i)_{0 \leq i \leq n}\), in set \(A\), is a function \(f\) from \(\{i \in \mathbb{N} \mid |i \leq n\}\) to \(A\) with \(f(i) = a_i\). A pair can be regarded as a finite sequence of length 2.

\textbf{Definition 4.} An alphabet is a finite set with elements that are characters. A string is a finite sequence of characters. If \(\Sigma\) is an alphabet then we mean \(\Sigma^*\) to mean the set of all strings that can be made using only characters in \(\Sigma\).

\textbf{Example 5.} \(A \cup \emptyset = A\)

\textbf{Definition 6.} A directed graph is a pair \((V,E)\), where \(V\) is a set of vertices and \(E\) is a set of ordered pairs of \(V\) known as directed edges. A closed path is a finite sequence \((v_i)_{0 \leq i \leq n}\) of vertices where for every \(i : 1 \leq i \leq n\), either \((v_i,v_{i-1}) \in E\) or \((v_{i-1},v_i) \in E\) and that \(v_0 = v_n\), but all other pairs of vertices in
the sequence are distinct. A directed u-v path is is a finite sequence \((v_i)_{0 \leq i \leq n}\) of vertices where for every \(i: 1 \leq i \leq n, (v_{i-1}, v_i) \in E\) and that \(u = v_0, v = v_n\) and they may be equal, but all other pairs of vertices in the sequence are distinct. A directed tree is a directed graph \((V, E)\) that contains no closed paths with strictly more than two different vertices and that there is some vertex \(u\) such that for all vertices \(v \in V\) there is a directed u-v path.

4 Formal Classical Logic

Preliminarily we want to develop a strict formal way of expressing logical entailment in the logics we care about.

Propositions are seen as meaningful statements that may be true or false. For example “the sky is blue” would be a proposition. Linguistically, we are used to building larger propositions from smaller ones, “if it is midday then the sky is blue” is built up of the statements “it is midday” and “the sky is blue” using a connective of “if... then”. It is clear that we need logical connectives in order for this building process but the other ingredient is a list of propositions that cannot be decomposed known as atomic formulae.

In order to specify a propositional formal language a set of all the atomic formulae \(\Sigma_{Prop}\) is needed. We also use a set \(\Sigma_{Conn}\) to contain the logical connectives. Here \(\Sigma_{Conn} = \{\bot, \top, \neg, \land, \lor, (, )\}\). ‘\(\bot\)’ is the false symbol while ‘\(\top\)’ which is a ‘not’ operation, \(\neg \phi\) denotes “not \(\phi\)”. For our binary operations we use the convention to place the operator between the formula. \(\phi \lor \chi\) means \(\phi\) or \(\chi\) or both, \(\phi \land \chi\) means both \(\phi\) and \(\chi\) and \(\phi \rightarrow \chi\) means if \(\phi\) then \(\chi\). The brackets are used to specify the order of operations.

A language is a subset of the set of strings over an alphabet. We use our two ingredients to build an alphabet \(\Sigma = \Sigma_{Prop} \cup \Sigma_{Conn}\) where “\(\cup\)” denotes the disjoint union as detailed in definition 1.

**Definition 7.** A finite string \(\phi\) in \(\Sigma^*\) is a well formed formula (w.f.f) when

1. \(\phi = \bot\).
2. \(\phi = \top\).
3. \(\phi \in \Sigma_{Prop}\).
4. \(\phi = (\neg \chi)\), and \(\chi\) is a well formed formula.
5. \(\phi = (\chi \rightarrow \psi)\), and \(\chi\) and \(\psi\) are well formed formulae.
6. \(\phi = (\chi \lor \psi)\), and \(\chi\) and \(\psi\) are well formed formulae.
7. \(\phi = (\chi \land \psi)\), and \(\chi\) and \(\psi\) are well formed formulae.

If \(\Sigma\) is a finite set of well formed formulae, we can use \((\land \Sigma)\) to mean a conjunction of all elements \(a_i\) in \(\Sigma\), namely, \((\ldots((a_1 \land a_2) \land a_3) \land a_4) \ldots \land a_n)\). In this case, we can also uses the notation \((\land_{i=1}^n a_i)\) here to mean the same thing. A similar notation exists for \(\lor\).

Well formed formulae are not necessarily formulae that are true, but those formulae that have the correct grammatical structure.

**Example 8.** suppose \(a, b, c, d \in \Sigma_{Prop}\) but \(z \notin \Sigma_{Prop}\)

<table>
<thead>
<tr>
<th>String</th>
<th>Well formed formula?</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>yes</td>
<td>(a) is atomic</td>
</tr>
<tr>
<td>(\neg a)</td>
<td>yes</td>
<td>(a) is a well formed formula</td>
</tr>
<tr>
<td>(z \land a)</td>
<td>no</td>
<td>(z) is not in our alphabet.</td>
</tr>
<tr>
<td>(\neg a \lor b)</td>
<td>no</td>
<td>‘(, )’ have not been placed.</td>
</tr>
<tr>
<td>((ab\land))</td>
<td>no</td>
<td>connective not placed in correct position.</td>
</tr>
<tr>
<td>((a \land (\neg a)))</td>
<td>yes</td>
<td>(a) and ((\neg a)) are well formed formulae.</td>
</tr>
</tbody>
</table>

A model is a subset of \(\Sigma_{Prop}\).

**Definition 9.** For a propositional language and a given model \(M\), the idea of satisfiability with the symbol \(\models\) can be defined inductively, for models acting on w.f.f or sets of w.f.f:
1. If $\phi \in \Sigma_{\text{Prop}}$, then $M \models \phi$ if and only if $\phi \in M$.
2. If $\phi = \top$, then $M$ does not satisfy $\phi$.
3. If $\phi = \bot$, then $M \models \phi$.
4. If $\phi = (\chi \rightarrow \psi)$, then $M \models \phi$ if and only if either $M \models \psi$ or $M$ does not satisfy $\chi$.
5. If $\phi = (\neg \chi)$, then $M \models \phi$ if and only if $M \models (\chi \rightarrow \bot)$.
6. If $\phi = (\chi \lor \psi)$, then $M \models \phi$ if and only if $M \models ((\neg \chi) \rightarrow \psi)$.
7. If $\phi = (\chi \land \psi)$, then $M \models \phi$ if and only if $M \models ((\neg \chi) \lor (\neg \psi))$.
8. If $\Delta = \{\phi_i : 1 \leq i \leq k\}$, then $M \models \Delta$ if and only if for all $i : 1 \leq i \leq k; M \models \phi_i$.

**Definition 10.** Semantic entailment is defined using the symbol: `\models` in the way that $\Gamma \models \phi$ means that for set $\Gamma$ of well formed formulae and the well formed formula $\phi$, for every model $M$ such that $M \models \Gamma$, then $M \models \phi$.

In order to simplify notation for sets of w.f.f, there is a relaxation on expressing a singleton set by its element. Additionally ‘,’ is to be used to replace the union ‘∪’ symbol. For example for set of w.f.f; $\Gamma$ and w.f.f $a$ and $b$; $\Gamma, a \models b$ means $\Gamma \cup \{a\} \models b$.

**Theorem 11.** *(The Deduction Theorem)* Let $\Gamma$ be a set of w.f.f and let $a$ and $b$ be w.f.f then,

$$\Gamma \models (a \rightarrow b) \iff \Gamma, a \models b$$

**Proof:** Suppose $\Gamma \models (a \rightarrow b)$, we now look at all the models that satisfy $\Gamma, a$. If $M \models \Gamma, a$ it must satisfy all formulae in $\Gamma$, so $M \models \Gamma$. We can then use our premise $\Gamma \models (a \rightarrow b)$, which by definition 10 means that as $M \models \Gamma$ it must satisfy $(a \rightarrow b)$. Using definition 7, either $M \models b$ or $M$ does not satisfy $a$. As $M \models a$ the latter cannot be true; it must then satisfy the former; $b$. Since $M$ was just a general model of $\Gamma, a$ with no additional information, it is true that $M \models b$ for all models $M$ that satisfy $\Gamma, b$. Hence, by definition 10; $\Gamma, a \models b$.

Conversely, suppose that $\Gamma, a \models b$. We look at all models that satisfy $\Gamma$. If $M \models \Gamma$, either it satisfies $a$ or it does not. In the case that it does then, by definition 7, it must satisfy $\Gamma, a$. We can then use our premise $\Gamma, a \models b$. It will follow from definition 10 that $M \models b$. Using $M \models b$ and definition 7, it is the case that $M \models (a \rightarrow b)$. In the case that $M$ does not model $a$, by definition 7, it then follows that $M \models (a \rightarrow b)$, we have shown that in either case, for a general model $M$ that satisfies $\Gamma; M \models (a \rightarrow b)$. From definition 10; $\Gamma \models (a \rightarrow b)$.

We have shown that by assuming one case we can deduce the other. \(\square\)

**Definition 12.** Let $\Sigma$ be a set of well formed formulae Var($\Sigma$) is the set of all atoms that occur as subformulae (that are substrings that are also well formed formulae in themselves) of elements of $\Sigma$.

## 5 Propositional Circumscription

Circumscription is a form of non-monotonic logic that looks at finding the ‘minimal’ situation that can occur, given our assumptions.

McCarthy first developed circumscription to give an explanation and formalism to the way we understand the “missionaries and cannibals” problem. The problem is that there are three cannibals and three missionaries on one side of a river with a single canoe that can seat two, to solve the problem one has to transport the people across the river without ever letting the cannibals outnumber the missionaries on a single side. While the problem has a known shortest solution, in informal classical logic it is impossible to prove that solution is in fact a solution; we cannot deduce with the given information that the canoe has a functioning oar or that there are no cannibals already on the other side. It is also impossible to prove that ‘trivial’ solutions do not exist, there may be a bridge a mile downstream, or that every missionary is also a cannibal, as the contrary is never stated. The way McCarthy addressed this, was to design a system of first order circumscription, which allowed one to assume some predicates to be false unless otherwise stated. While circumscription can be defined over a first order language, we are primarily interested in the propositional case. [20]
For circumscription, the propositional atoms of a propositional language need to be separated into 3 distinct categories; those atoms that are minimised over, those atoms that are fixed and those atoms which can vary from the minimisation. $P$ is the set of all atoms that are to be minimised over and $R$ is the set of fixed atoms. $P$ and $R$ are disjoint sets.

Using the sets, $P$ and $R$, a pre-order $\leq_{P,R}$ can be defined on the models $I$, $J$ as follows.

$$I \leq_{P,R} J \iff (I \cap P \subseteq J \cap P) \land (I \cap R = J \cap R)$$

This translates to saying that $I$ assigns atom $p$ in $P$ to be true only if $J$ does while $I$ and $J$ give the same sub-model over $R$. Note that it does not necessarily have to be a partial order. Suppose that our set of propositional atoms $\Sigma_{prop} = \{a,b,c\}$, $P = \{a\}$ and $R = \{b\}$, let $I$ and $J$ be two distinct models, with $I = \{b\}$, $J = \{b,c\}$. It then follows that $(I \cap P = J \cap P)$ and $(I \cap R = J \cap R)$ so $I \leq_{P,R} J$ and $J \leq_{P,R} I$ but $I \neq J$. Below is another example.

**Example 13.**

$$\{a,c,d,e\} \leq_{\{a,b,c\};\{d\}} \{a,b,c,d\}$$

In this example, $P = \{a,b,c\}$, $R = \{d\}$

$$\{a,c,d,e\} \cap P = \{a,c\} \subseteq \{a,b,c\} = \{a,b,c,d\} \cap P$$

$$\{a,c,d,e\} \cap R = \{d\} = \{a,b,c,d\} \cap R$$

So the two conditions are satisfied.

$\leq_{P,R}$ is still a transitive relation and minimality can be defined as such for models. Let $I \models \Gamma$. We say that $I$ is a $P;R$-minimal model of $\Gamma$ (and denote it by $I \models_{P;R} \Gamma$) if and only if for any model $J$: $(J \models \Gamma) \Rightarrow ((J \leq_{P,R} I) \Rightarrow (I \leq_{P,R} J))$.

If $\phi$ is a formula in our language then $\Gamma \models_{P,R} \phi$ means that $\phi$ is satisfied by all $P;R$-minimal models of the set $\Gamma$. This is the notion of semantic entailment in circumscription. A few special cases can be noted. When $P = \emptyset$ then $\models_{P,R}$ is synonymous with $\models$, the classical entailment. When $P = \Sigma_{prop}$ then entailment is known as minimal entailment and we denote it with the symbol $\models_M$.

**Example 14.**

$$(a \rightarrow b), (a \lor c) \models_{b,c,a} (b \rightarrow a)$$

Firstly we can look at all possible models as in Fig. 1.

<table>
<thead>
<tr>
<th>$a \in M$</th>
<th>$b \in M$</th>
<th>$c \in M$</th>
<th>$M$ satisfies $(a \rightarrow b), (a \lor c)$?</th>
<th>$M$ minimal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

**Fig. 1.** Models of $(a \rightarrow b), (a \lor c)$

This example is true because the minimal models are $\{c\}$ and $\{a,b\}$. If we then look at the definition for satisfiability for $\rightarrow$, the left hand side is satisfied by both $\{c\}$ and $\{a,b\}$. This example also demonstrates the difference between circumscription and classical logic as the expression $(a \rightarrow b), (a \lor c) \models_{b,c,a} (b \rightarrow a)$ is not true, since model $\{b,c\}$ falsifies it.
As a remark one can look at whether the deduction theorem holds in circumscription. In classical logic by taking $b$ as a premise we should be able to infer $a$. To see if this holds in circumscription we find the minimal models of $b, (a \rightarrow b), (a \lor c)$ as in Fig. 2.

Noteworthy of this example is that even though the right hand side is an implication, bizarrely taking $b$ as a premise does not derive $a$, namely $b, (a \rightarrow b), (a \lor c) \not\models_{b,c,a} a$. This is because $\{b, c\}$ and $\{a, b\}$ are now the only minimal models, each with a different truth value of $a$. So the deduction theorem (Theorem 11) does not translate in circumscription. Furthermore $b, (a \rightarrow b), (a \lor c) \not\models_{b,c,a} (b \rightarrow a)$ as $\{b, c\}$ contradicts it, hence there is not monotonicity.

6 Complexity

In this report a function $f$ is in the class $O(g(n))$ if there is some constant $c$ such that $(\exists m \in \mathbb{N})(\forall n \geq m)(f(n) \leq c \cdot g(n))$. A function $f$ is in the class $\Omega(g(n))$ if there is some constant $c$ such that $(\forall m \in \mathbb{N})(\exists n \geq m)(f(n) \geq c \cdot g(n))$. [14]

The complexity class $P$ is the class of languages with a decision algorithm with polynomially bounded minimum time on a deterministic Turing Machine. Below is a definition of a scheme of classes that contain $P$.

Definition 15. [14]

- $\Pi^P_0 = \Sigma^P_0 = P$.
- For $n \geq 1$, language $L$ is in class $\Pi^P_n$ when it has a polynomial $p$, and some $H \in \Sigma^P_{n-1}$, such that the indicator function of $L$ on input string $x$ is equal to the function that takes $x$ to the truth value (0 if false, 1 if true) of $\forall y((|y| < p(|x|)) \rightarrow \phi(x,y))$, where $\phi$ is the function that maps strings to well formed formulae such that the function $g$ that maps input string $z$ to the truth value of $\phi(z)$ is the indicator function of language $H$.
- For $n \geq 1$, language $L$ is in class $\Sigma^P_n$ when it has a polynomial $p$, and some $H \in \Pi^P_{n-1}$, such that the indicator function of $L$ on input string $x$ is equal to the function that takes $x$ to the truth value of $\exists y((|y| < p(|x|)) \land \phi(x,y))$, where $\phi$ is the function that maps strings to well formed formulae such that the function $g$ that maps input string $z$ to the truth value of $\phi(z)$ is the indicator function of language $H$.

While definition 15 is longwinded and formal the basic idea is that we alternate quantifiers over polynomially bounded variables and if we begin with the universal quantifier we are in a $\Pi$ class and if we begin with an existential quantifier we are in a $\Sigma$ class. Figure 3 shows the classes ordered by $\subset$, all the classes contained in the $\text{PSPACE}$ class which is the class of languages with a decision algorithm that uses a polynomially bounded number of cells in a deterministic Turing Machine.

An example of one of our classes is the complexity class $\Pi^P_2$, which is a subset of the polynomial hierarchy. A language $L$ is $\Pi^P_2$-hard when for every language $K$ in $\Pi^P_2$ there is a polynomial time function $f$ that maps $L$ onto $K$. A language $L$ is $\Pi^P_2$-complete when it is $\Pi^P_2$-hard and in $\Pi^P_2$. Eiter and Gottlob showed [11] that the language of true entailment formula in propositional circumscription is $\Pi^P_2$-complete. [14]

<table>
<thead>
<tr>
<th>$a \in M$</th>
<th>$b \in M$</th>
<th>$c \in M$</th>
<th>$M$ satisfies $b, (a \rightarrow b), (a \lor c)$?</th>
<th>$M$ minimal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Fig. 3. Sketch of the polynomial hierarchy on a Hasse diagram ordered by $\subset$

Definition 16. (Cook, Reckhow [9]) A proof system for an arbitrary language $L$ over alphabet $\Gamma$ is a polynomial-time computable partial function $f : \Gamma^* \rightarrow \Gamma^*$ with \{ $y \in \Gamma^*(\exists x \in \Gamma^*)(f(x) = y)$ \} = $L$. A proof of string $y$ in system $f$ is a string $x$ such that $f(x) = y$.

From this definition we can start defining proof length. For $f$ a proving system of language $L$ and string $x \in L$ we can define $s_f(x) = \min(|w| : f(w) = x)$. $s_f$ is the partial function that tells us the minimum proof length of a theorem. We can overload the notation for $n \in \mathbb{N}$ where $s_f(n) = \max(s_f(x) : |x| \leq n)$. We can define what it means for $f$ to be $t$-bounded by a function $t : \mathbb{N} \rightarrow \mathbb{N}$ which is when $(\forall n \in \mathbb{N})(s_f(n) \leq t(n))$.

A sequent is a pair $(\Gamma, \Delta)$ with $\Gamma$ and $\Delta$ finite sets of well-formed formulae. A sequent is usually written in the form $\Gamma \vdash \Delta$ for classical logic. A sequent $\Gamma \vdash \Sigma$ is true if and only if every model $M$ that models $(\bigvee \Sigma)$ also models the formula $(\bigwedge \Gamma)$, where the disjunction of the empty set is taken as $\bot$ and the conjunction as $\top$. The notation $A,B$ means the union of $A$ and $B$ if they are both sets of formulae, but replaces each with their singletons in the case that they are formulae.

The proof system $LK$ details the derivability of a sequent although there are several different, but basically equivalent ways to define it. Gentzen’s $LK$ is given a definition in Fig. 4 [12]. It should be read as follows: The sequents on the top are called premises and the sequent on the bottom is known as a conclusion. The proof of $\Gamma \vdash \Delta$ is a sequence (as a proof, given in definition 16 is a string we need to express this sequence as a string, by introducing a separator symbol in the language) $(t_i)_{1 \leq i \leq k}$ of well-formed formula such that for all $t_i$ there is a rule of $LK$, with $t_i$ a conclusion and every premise appearing as some $t_j$ for $j < i$. $t_k$ must be the sequent we aim to prove. Notice that the rules here do not contain structural rules; contraction or exchange, because our structures come for free with the use of sets.

The completeness theorem is important, as it means that $LK$ really is an appropriate calculus to use for classical logic. Note, in particular, the soundness of rule $(\bullet \vdash)$, which gives us monotonicity of classical propositional logic.

Theorem 17. The Completeness Theorem (Gentzen [12], Godel [13]) A sequent is true if and only if it is derivable in $LK$.

Example 18. We want to derive $\vdash (p \lor (\neg p))$ (the law of excluded middle). While a proof is formally defined as a sequence we use a prooftree in our examples to show where the premises are used and which rules are used.
7 The Antisequent Calculus

A useful ingredient for working towards a calculus for non-monotonic logics is the notion of antisequent with symbol $\not\models$. $\Gamma \not\models \varphi$ is an expression for “there is a model $M$ that satisfies all formula in $\Gamma$ but for which $(\neg \varphi)$ holds”.

The antisequent $\Gamma \not\models \Sigma$ is true if and only if there is some model $M \models \Gamma$ so that for all $\varphi$ in $\Sigma$ $M \models (\neg \varphi)$. This is equivalent to saying that we cannot derive $\Gamma \vdash \Sigma$.

A proof is used for an antisequent calculus with the symbol: $\not\models$ its rules are given in Fig. 5. The completeness theorem is also important and was proven by Bonatti.

Theorem 19. (Bonatti [5]) An antisequent is true if and only if it is derivable in the antisequent calculus.

Example 20. We want to derive the well known antisequent result $(a \rightarrow b) \not\models (b \rightarrow a)$. It is given in the proof tree below.

\[
\frac{p \not\models \neg p \quad \neg\models \neg p \quad \not\models \neg p \\ \neg\models \neg p \quad \not\models \neg p \quad \neg\models \neg p}{a \rightarrow b} \not\models (b \rightarrow a)
\]
Remark 21. The truth of an antisequent tells us of the existence of a model that satisfies the left hand side but contradicts the right hand side. However it does not point immediately to the model itself. Given a derivation of an antisequent in AC we only have unary rules (rules that require only one premise) and a single axiom (requires no premises) so any sequent results from unary rules applied to an axiom (\(\vdash\)). The axiom (\(\vdash\)) gives us a trivial model that satisfies its RHS but not LHS; we take the RHS atoms to be true and all other atoms to be false. Next we can observe that in every application of the unary rules, the model that makes the premise true also makes the conclusion true. For example if \(\Gamma \nvdash \Sigma, \alpha, \beta\) is a true antisequent, then there is some model \(M\) that satisfies \(\Gamma\), but not any of \(\Sigma\), \(\alpha\) or \(\beta\). So it must not satisfy \((\alpha \lor \beta)\). Thus \(M\) allows the conclusion sequent of rule \((\lor \vdash)\); \(\Gamma \nvdash \Sigma, (\alpha \lor \beta)\) to be true.

8 A Sequent Calculus for Circumscription

In 1992 Olivetti created a sequent calculus MLK that worked primarily for minimal entailment. In 2002 Bonatti and Olivetti found sequent calculi for several different non-monotonic logics, among them was circumscription in a sequent calculi referred to as CIRC. A new constraint has been added to the sequents; \(\Sigma\) which is a set of atoms disjoint from \(R\), so the new sequents are of form \(\Sigma; \Gamma \vdash P; R \Delta\) (which may be regarded as a 5-tuple). As defined by Bonatti and Olivetti [6], the sequent \(\Sigma; \Gamma \vdash P; R \Delta\) is true when: “In every \(\leq_{P,R:R}\)-minimal model of \(\Gamma\) that satisfies \(\Sigma\) there is a formula \(\phi \in \Delta\) that holds.”

It is immediate that propositional circumscription sequents coincide with the formula where \(\Sigma\) is empty. The rules of the syntactic notion are given in Fig. 6 and Bonatti and Olivetti proved their soundness and adequacy as stated in the following theorem [6].

Theorem 22. (Bonatti, Olivetti [6]) A sequent \(\Sigma; \Gamma \vdash P; R \Delta\) is true if and only if it is derivable in CIRC.

9 Lower Bound for Circumscription

One of the aims of this project is to give a bound to the proof complexity of CIRC.

Theorem 23.

\[ s_{CIRC}(n) \in \Omega(2^n) \]
\[
\begin{align*}
\frac{\varphi \in \Sigma}{\Sigma, \Gamma \vdash P} \quad (C1) & \quad \frac{\Gamma \vdash \Delta}{\Sigma; \Gamma \vdash P, \Delta} \quad (C2) \\
\frac{\varphi \in \Sigma}{\Sigma, \Gamma \vdash P} \quad (C3) & \quad \frac{\Gamma \vdash \Delta}{\Sigma; \Gamma \vdash P, \Delta} \quad (C4)
\end{align*}
\]

In all rules \( q \) is atomic and not in \( P \) or \( R \) and \( \neg P = \{ \neg p, p \in P \} \)

\textbf{Fig. 6.} Inference rules of the Circumscription calculus \textit{CIRC} proposed by Bonatti and Olivetti [6]

\[
\begin{align*}
\frac{(a \lor b), b \not\in a}{\alpha; (a \lor b), b \vdash \Delta} & \quad (C1) & \frac{(a \lor b), (\neg a) \vdash (b \rightarrow \neg a)}{(a \lor b), b \vdash \Delta} & \quad (C2) & \frac{(a \lor b), (\neg b) \vdash (b \rightarrow \neg a)}{(a \lor b), (\neg b) \vdash \Delta} & \quad (C3) \\
\frac{(a \lor b), \neg (a \rightarrow \neg a)}{(a \lor b), b \vdash \Delta} & \quad (C2) & \frac{(a \lor b), \neg (b \rightarrow \neg a)}{(a \lor b), b \vdash \Delta} & \quad (C3) & \frac{(a \lor b), (\neg b) \vdash (b \rightarrow \neg a)}{(a \lor b), (\neg b) \vdash \Delta} & \quad (C4)
\end{align*}
\]

\textbf{Fig. 7.} Example of a derivation of \((a \lor b) \vdash_{a,b} (b \rightarrow \neg a)\) in \textit{CIRC}

\begin{proof}
The idea is to produce a linear sized class of formulae in \( n \), but whose proof lengths grow exponentially.

We first begin by defining a scheme of propositional variables \( P_n \) a natural number \( n \). These will be used in the sequents.

\[
P_n = \{ p_i, q_i : 1 \leq i \leq n \}
\]

Then we define some sets of well formed formulae; for all \( n \geq 1 \):

\[
\Gamma_n := \{ (p_i \lor q_i) : 1 \leq i \leq n \},
\]

\[
\Delta_n := \{ (\land_{1 \leq i \leq n}((p_i \land (\neg q_i)) \lor (q_i \land (\neg p_i)))) \}
\]

The class of sequents in \( n \) are:

\[
\emptyset; \Gamma_n \vdash_{P, \emptyset} \Delta_n
\]

Semantically this is true, because every minimal model of one of these sequents will have \( p_i \) or \( q_i \) but cannot have both as these models are not minimal. It is quite clear why these sequents are linear length.

Now for the argument that this must have exponential proof. We build a proof by induction by considering a partition of \( P_n \) into three disjoint sets; \( P^0, P^+, P^- \) and let

\[
k = n - |P^- \cup P^+|
\]

\textit{Induction Hypothesis on} \( k \) for \( k \leq n \) : \( P^+; \Gamma_n, \neg P^- \vdash_{P^0, \emptyset} \Delta_n \) needs at least \( 2^k \) lines of the form \( B; \Gamma_n, \neg A \vdash_{C; \emptyset} \Delta_n \) where \( A, B, C \) are sets of atoms, with \( P^+ \subseteq B, P^- \subseteq A \), and with \( B, A \) disjoint in any line.

The base case is when \( k = 0 \), it is obvious that a single line is needed to at least state the end result, this line is \( P^+; \Gamma_n, \neg P^- \vdash_{P^0, \emptyset} \Delta_n \), and the rest of the induction hypothesis is trivial.

In the inductive case the aim is to show that if \( 1 \leq k \leq n \) then

\[
P^+; \Gamma_n, \neg P^- \vdash_{P^0, \emptyset} \Delta_n
\]

can only be inferred in \textit{CIRC} by using (C3) in the form of:
Lemma 25. For all disjoints sets of atoms $\Sigma^+, \Sigma^-$ such that $\text{Var}(\Gamma \cup \Delta) = \Sigma^+ \sqcup \Sigma^-$. There is a polynomial $p$ such that for all $\Sigma^+, \Sigma^-, \Gamma, \Delta$, the proof length, $s_{LK}(\Sigma^+; \Sigma^-, \Gamma \vdash \Delta)$ is bounded above by $p(\Sigma^+, \Sigma^-, \Gamma \vdash \Delta)$.

Proof. See Appendix D.2

Lemma 24. For a well formed formula $s_{LK}(\phi \vdash \phi) \leq 4|\phi|^2 + 3$.

In actual fact the exact upper bound is not important, but that this can be done in polynomial length.

Proof. See Appendix D.1

Theorem 26. There is a polynomial function $q$ and constant $c$ such that

$$s_{LK}(n) \in O(q(n) \exp(cn))$$

Proof. We wish to prove $\Gamma \vdash \Delta$ which is a sequent of length $n$. We can obtain for every combination of $\Sigma^+$ and $\Sigma^-$ for $\text{Var}(\Gamma \cup \Delta) = \Sigma^+ \sqcup \Sigma^-$ some proof of $\Sigma^+; \Sigma^-, \Gamma \vdash \Delta$ bounded above by a polynomial $p(n)$. We have exponentially many of these combinations for $\Sigma^+$ and $\Sigma^-$. We take $q(n) = 14n + 4 + p(n)$ and $c \geq \ln(3)$.

Induction Hypothesis on $k$: For all combinations of disjoint sets of atoms $A^+, A^-$ such that $A^+ \sqcup A^- = Q$ where $Q \subseteq \text{Var}(\Gamma \cup \Delta)$ and $|\text{Var}(\Gamma \cup \Delta)\setminus Q| = k$ we have $s_{LK}(A^+; A^-, \Gamma \vdash \Delta) \leq q(n) \exp(pk)$.

Base case: when $k = 0$, exp(0) = 1 so $q(w) \exp(ck) \geq q(w) \geq p(w) \geq s_{LK}(A^+; A^-, \Gamma \vdash \Delta)$.

Inductive case: to increment $k$ by 1 we choose any variable $r$ to be eliminated from $Q$ to get set $Q' = Q \setminus \{r\}$. For every combination of disjoint sets of atoms $A^+, A^-$ such that $A^+ \sqcup A^- = Q'$. Using our induction hypothesis we have proofs of $r, A^+, A^-, \Gamma \vdash \Delta$ and $(-r), A^+, A^-, \Gamma \vdash \Delta$ in length shorter than $q(n) \exp(ck)$. We extend as follows:
Therefore since 14w+3 ≤ q(n) exp(ck). We have that \( s_{LK}(A^+, -A^-, \Gamma \vdash \Delta) \leq 3q(n) \exp(ck) \leq \exp(c)q(n) \exp(ck) \leq q(n) \exp(c(k+1)) \). Hence the inductive case is true.

We have \( k = \var{\Gamma \cup \Delta} \) the induction shows that \( s_{LK}(\Gamma \vdash \Delta) \leq q(w) \exp(ck) \).

Bonatti and Olivetti developed several calculi for several different non-monotonic logic. Default logic is a logic that allows default rules, where a result can be inferred unless another result is proven, it allows two types of reasoning, credulous reasoning and sceptical reasoning, Bonatti and Olivetti developed calculi \( BO^{CRED} \) and \( BO^{SKEP} \) to represent the logics respectively. Another type of non-monotone logic, Autoepistemic Logic, again with credulous and sceptical variants has been given respective calculi; \( CAEL \) and \( SAEL \) by Bonatti and Olivetti [6].

**Theorem 27.** (Beyersdorff et al. [2]) There is a constant \( c \) and a polynomial \( p \) such that \( s_{LK}(n) \leq s_{BO^{CRE}}(n) \leq p(n)s_{LK}(cn) \)

**Theorem 28.** (Beyersdorff [3]) There is a polynomial \( p \) such that \( s_{LK}(n) \leq s_{CAEL}(n) \leq p(n)s_{LK}(n) \)

**Theorem 29.** (Beyersdorff et al. [2])

\[ s_{BO^{SKEP}}(n) \in \Omega(2^n) \]

**Theorem 30.** (Beyersdorff [3])

\[ s_{SAEL}(n) \in \Omega(2^n) \]

As seen in theorems 23, 29 and 30 it seems as though the calculus \( CIRC \) is similar in its complexity to the sceptical calculi by Bonatti and Olivetti for default logic and autoepistemic logic. But because of the lower bounds we would not hope to do as well as the credulous logics as in theorems 27 and 28 without finding a lower bound for \( LK \). Ideally we would want to look at a calculus which does not have the pitfalls of \( CIRC \) before any attempt to look for an upper bound similar to theorems 27 and 28 is viable.

**Theorem 31.** The following statements are true and result from theorems 26, 27 and 28.

- There exists polynomial \( p \) and constant \( c \) such that there are infinitely many \( n \in \mathbb{N} \) with \( s_{LK} \leq p(n)s_{CIRC}(cn) \).
- There exists polynomial \( p \) and constant \( c \) such that there are infinitely many \( n \in \mathbb{N} \) with \( s_{BO^{CRE}} \leq p(n)s_{CIRC}(cn) \).
- There exists polynomial \( p \) and constant \( c \) such that there are infinitely many \( n \in \mathbb{N} \) with \( s_{SAEL} \leq p(n)s_{CIRC}(cn) \).

11 Comparison to MLK

Additionally in contrast to the \( CIRC \) calculus by Olivetti is the \( MLK \) calculus defined by Olivetti [22]. Here we use the notation \( \vdash_M \), in the way that a sequent \( \Gamma \vdash_M \Delta \) is true if and only if \( \vdash \Delta \) is true in all \( \Sigma_{\text{Prop};\emptyset} \) minimal models of \( \Gamma \).

**Definition 32.** [22] To introduce derivability we use the property of a positive atom in a formula.

- Atom \( p \) is positive in formula \( p \).
- Atom \( p \) is positive in formula \( \phi \) if and only if it is negative in \( \neg \phi \).
- If atom \( p \) is positive in formula \( \phi \) or \( \chi \), it is positive in \( \phi \land \chi \) and \( \phi \lor \chi \).
- If atom \( p \) is negative formula in \( \phi \) or positive in \( \chi \) then it is positive in \( \phi \rightarrow \chi \).

The \( MLK \) calculus is detailed in Fig. 8.
but we have shown that they cannot, so no simulation can occur.

23 that the best proofs are at least exponential, which means it cannot be polynomially bounded above.

Theorem 33. (Olivetti [22]) A sequent \( \Gamma \vdash_M \Delta \) is true if and only if it is derivable in MLK.

Definition 34. (Krajíček, Pudlák [16], Cook, Reckhow [9])

- We say proof system \( f \) (polynomially) simulates proof system \( g \) when for all \( g \)-proofs \( x \) there is a \( f \)-proof \( y \), with \( |y| \leq p(|x|) \), and \( f(y) = g(x) \).
- We say proof system \( f \) \( p \)-simulates proof system \( g \) when for all \( g \)-proofs \( x \) there is a polynomial-time computable function \( h \) such that \( h(x) \) is a \( f \)-proof and \( g(x) = f(h(x)) \).

It follows immediately that a \( p \)-simulation implies a simulation.

Theorem 35. CIRC does not polynomially simulate MLK for minimal entailment.

Proof. We use the hard examples in theorem 23 and show that they can be proved in MLK in polynomial length. This means that if a simulation exists then they can also be proved in polynomial length in CIRC, but we have shown that they cannot, so no simulation can occur.

Using the same notation as in the proof of theorem 23, but now for fixed \( n \) define \( \Gamma_n \) as \( \Gamma_n \setminus \{(p_i \lor q_i)\} \):

\[
\begin{align*}
\Gamma_n \vdash_M (p_i \lor q_i) & \quad \left( \neg \right) \\
\Gamma_n, p_i \vdash_M p_i & \quad \left( \bullet \vdash \right) \\
\Gamma_n, p_i \vdash_M p_i & \quad \left( \vdash_M \right) \\
\Gamma_n, p_i \vdash_M (p_i \land \neg q_i) & \quad \left( \vdash_M \land \right) \\
\Gamma_n, p_i \vdash_M ((p_i \land \neg q_i) \lor (q_i \land \neg p_i)) & \quad \left( \vdash_M \lor \right) \\
\Gamma_n \vdash_M ((p_i \land (\neg q_i)) \lor (q_i \land (\neg p_i))) & \quad \left( \vdash_M \lor (\vdash_M) \right)
\end{align*}
\]

The proof tree above shows for all \( n \), \( \Gamma_n \vdash_M ((p_i \land (\neg q_i)) \lor (q_i \land (\neg p_i))) \) can be proved in linear length. By repeated use of rule \( (\vdash_M \land) \) to build the big conjunction at most a linear number of times we show \( \Gamma_n \vdash_M \Delta_n \) is provable in a polynomial number of characters.

If CIRC were to polynomially simulate these proofs then there would be a scheme of proofs with a polynomial bounded number of characters in CIRC but it has been demonstrated in the proof of theorem 23 that the best proofs are at least exponential, which means it cannot be polynomially bounded above. \( \square \)
Lemma 36. Let $\Sigma_{\text{prop}} = P^0 \sqcup P^+ \sqcup P^-$ then $P^+; \Gamma, \neg P^- \vdash_{p,0} \Delta$ is true if and only if $\Gamma, \neg P^- \vDash_M \Delta, \neg P^+$ is true.

Proof. if $P^+; \Gamma, \neg P^- \vdash_{p,0} \Delta$ then let $N$ be a $P^+ \sqcup P^0; \emptyset$-minimal model of $\Gamma, \neg P^-$. $N$ is also $\Sigma_{\text{prop}}$-minimal as it cannot satisfy any $p \in P^-$. Either $N$ satisfies $P^+$ in which case it must satisfy $\Delta$ by our premise, or it must satisfy the disjunction of $\neg P^+$ hence $\Gamma, \neg P^- \vDash_M \Delta, \neg P^+$.

If $\Gamma, \neg P^- \vDash_M \Delta, \neg P^+$, then let $N$ be a $P^+ \sqcup P^0; \emptyset$-minimal model of $\Gamma, \neg P^-$. $N$ must not satisfy any $p \in P^-$, hence it is $\Sigma_{\text{prop}}$-minimal, which means if it satisfies $P^+$ it must satisfy $\Delta$. Hence $P^+; \Gamma, \neg P^- \vdash_{p,0} \Delta$. \qed

Theorem 37. MLK simulates CIRC for minimal entailment.

Proof. To prove this theorem we have to show that for any conclusion from our four rules, we can infer the conclusion from the premises in a proof length that is polynomially bounded by the original use of the rule.

The induction argument forms from taking each line of the proof in CIRC and translating it via lemma 36 and showing it can be inferred in polynomial time in the MLK calculus.

The problem with rule (C1) is that it uses the antisequent calculus, and that MLK does not use the antisequent calculus in any rule. We do not necessarily have to use a premise that corresponds with the antisequent calculus instead we prove the conclusion in polynomial length.

Here we introduce the notation $A[x/y]$, this is the formula of $A$ with every occurrence of formula $y$ replaced by formula $x$. Alternatively if we write $A(x)$ instead of $A$ then the notation $A(y)$ tells us to replace $x$ with $y$ and is the same as $A[y/x]$. We can do the same for sets of formulae.

Suppose that $q, \Sigma; \Gamma \vdash_{P,0} \Delta$ is inferred via (C1) in our CIRC calculus and is used in a proof of some minimal entailment sequent, then it is true that $\Gamma, \neg P \not\vdash q$, so there is some model $N$ such that $\Gamma, \neg P$ holds and $(\neg q)$ holds. Moreover, since we have the proof in the antisequent calculus we can recover this $N$. Let $\Sigma^+$ be all atoms that occur in $\Gamma$ that are true in $N$ and let $\Sigma^-$ be all atoms that occur in $\Gamma$ but not in $q, P$ or $\Sigma^+$ (all other atoms that are false in $N$). For atomic sets $A = \{a_i\}, B = \{b_i\}$ let us define $\gamma(A, B) = (\bigland \Gamma[\bot/a1][\bot/a2]|...[\top/b1][\top/b2]|...$. This notation allows us to replace the variables with their assigned value.

It then follows from lemma 25 that $(\neg q), \Sigma^+ \vdash \gamma(\Sigma^- \cup P, \emptyset)$ in a cubic size proof. We do more below to obtain $\Sigma^+ \vdash_M \gamma(\Sigma^- \cup P, \emptyset)$, which will be the base case in our induction,

$$\frac{(\neg q), \Sigma^+ \vdash \gamma(\Sigma^- \cup P, \emptyset)}{(\neg q), \Sigma^+ \vDash_M \gamma(\Sigma^- \cup P, \emptyset)} \quad \frac{\vdash_{M}}{(\neg M)}$$

Suppose now inductively on $|Q|$ that $\Sigma^+ \vDash_M \gamma(U, \emptyset)$ where $U \cup Q = \Sigma^- \cup P$ has a proof size bounded above by a quartic. The idea is that we replace $\bot$ with atoms in $\gamma$ one by one and because of minimisation we can still infer the conclusion.

Using lemma 25 we can infer $\Sigma^+, \gamma(U \cup \{p\}, \emptyset), (\neg p) \vdash \gamma(U, \emptyset)$ in cubic length.

$$\frac{\Sigma^+, \gamma(U \cup \{p\}, \emptyset), (\neg p) \vdash \gamma(U, \emptyset)}{\Sigma^+, \gamma(U \cup \{p\}, \emptyset) \vDash_M \gamma(U, \emptyset)} \quad \frac{\vdash_{M}}{(\neg M)}$$

Now we have the inductive and base case, and all can be proved in quartic length because it is a linear bounded number of iterations of a cubic length extension for each. So we have for the case that $Q = \Sigma^- \cup P$, $\Sigma^+ \vDash_M \gamma(U, \emptyset)$, then we proceed extending the proof without changing the fact we have a quartic upper bound;

$$\frac{\Sigma^+ \vdash_M \gamma(U, \emptyset)}{\Sigma^+ \vdash_M \gamma(U \setminus \{p\}, \emptyset) \vdash_M \gamma(U, \emptyset)} \quad \frac{\neg p \vdash \gamma(U, \emptyset)}{(\neg M)}$$
It now follows $\Sigma^+, (\bigwedge \Gamma) \vdash M (\neg q)$, we now proceed by induction on the number of elements of $\Gamma'$ to change $(\bigwedge \Gamma)$ into $\Gamma$. The induction hypothesis is that we can obtain $\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash (\neg q)$ in quartic length.

For the base case what we already have is sufficient; up to this point we have not exceeded a quartic upper bound.

For the inductive case below is the extension we get the proof for the iterative step on the number of elements of $\Gamma'$, we use an additional assumption that we can prove $\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash (\neg q)$ in quartic length for $\phi \in \Gamma'$ this works by using our short proof of $\phi \vdash \phi$ and building the big conjunction using $(\bullet \vdash \bullet)$ and $(\bigwedge \vdash \vdash)$ rules. Then weakening with $(\bullet \vdash)$ to get the sequent.

$$\frac{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)}{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)} \frac{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)}{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)} \frac{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)}{\Sigma^+, (\bigwedge \Gamma), \Gamma' \vdash_M (\neg q)}$$

Again the extension’s cubic length is clear to see, so overall we still retain the quartic bound.

Now after the induction, we have $\Sigma^+, (\bigwedge \Gamma), \Gamma \vdash_M (\neg q)$ we need to cut out the $(\bigwedge \Gamma)$. We can prove $\Sigma^+, \Gamma \vdash (\bigwedge \Gamma)$ and in cubic length using lemma 25. We can also use weakening $(\bullet \vdash)$ and $(\vdash \bullet)$ with on the proof of $(\neg q) \vdash (\neg q)$ to get $\Sigma^+, \Gamma, (\neg q) \vdash (\neg q), \neg \Sigma^-, \Delta$.

$$\frac{\Sigma^+, \Gamma \vdash (\bigwedge \Gamma)}{\Sigma^+, \Gamma \vdash_M (\bigwedge \Gamma)} \frac{\Sigma^+, \Gamma \vdash_M (\bigwedge \Gamma)}{\Sigma^+, \Gamma \vdash_M (\bigwedge \Gamma)} \frac{\Sigma^+, \Gamma \vdash_M (\bigwedge \Gamma)}{\Sigma^+, \Gamma \vdash_M (\bigwedge \Gamma)}$$

Repeated use of axiom $(\vdash_M \neg)$ on formulae gets us the sequent $\Gamma \vdash_M \Delta, \neg q, \neg \Sigma$ which is equivalent to our conclusion in (C1). Overall we have a linear use of a cubic length proofs hence our proof length is quartic.

For (C2) we can infer $\Sigma, \Gamma \vdash_M \Delta$ then negate $\Sigma$ on the RHS. The quartic bound still holds.

For (C3) we only care about the proof that end with a minimal entailment formula (or specifically translations of such), so our premises must be $\Gamma, \neg P^− \vdash_M \Delta, \neg P^+, \neg p$ and $\Gamma, \neg P^−, \neg p \vdash_M \Delta, \neg P^+$.

$$\frac{\Gamma, \neg P^− \vdash_M \Delta, \neg P^+, \neg p}{\Gamma, \neg P^− \vdash_M \Delta, \neg P^+} \frac{\Gamma, \neg P^−, \neg p \vdash_M \Delta, \neg P^+}{\Gamma, \neg P^− \vdash_M \Delta, \neg P^+}$$

So in some sense we have shown an inductive step of the proof where we translate using lemma 36 to apply the rules. The quartic bound still holds. As required using lemma 36’s translation. Since we have no fixed elements (C4) can be ignored.

\[
\square
\]

Remark 38. It should be easy to extend this to a p-simulation (that is where we explicitly have a polynomial time algorithm for changing a proof in CIRC to a proof in MLK). A major ingredient for this would be an polynomial time algorithm that gives us a proof of the sequents in lemma 25, another major ingredient would be to utilise remark 21 to give us the exact $\Sigma^+$ and $\Sigma^−$ in our model for the antsequent, then it would be a simple matter of following the many steps outlined in the proof above.

### 12 Comparison to Niemelä’s Tableau Calculus

A literal is a well formed formula $a$ or $(\neg a)$ where $a$ is an atom. A clause is a set of literals that represent a disjunction, i.e. the clause $(a, b, (\neg b))$ represents the disjunction $((a \lor b) \lor (\neg b))$. A clausal theory is a set of clauses; the conjunctive normal form of a well formed formula $\phi$ is a clausal theory whose conjunction of the individual disjunctions of its elements is logically equivalent to $\phi$.

**Example 39.** Let $a, b, c$ be atoms. The conjunctive normal form of $((a \rightarrow b) \land c)$ is $\{\{\neg a, b\}, \{c\}\}$. 

A tableau, $T$, is a directed tree using the symbol $\rightarrow$ to denote a directed edge ($u \rightarrow v$ means there is a directed edge from node $u$ to node $v$). Formally, a branch is a sequence of nodes $u_0, u_1, ..., u_k$ such that for $1 \leq i \leq k$, $u_{i-1} \rightarrow u_i$ (it is a directed path) and there is no node $v$ such that $v \rightarrow u_0$ or $u_n \rightarrow v$ (that is maximal).

For clausal theory $\Gamma$ and well-formed formula $\phi$, a $\Gamma, \phi$-tableau is a tableau defined as follows. We start with a sequence $(C_i)_{0 \leq i \leq k}$ of all the clauses of $\Gamma \cup \Delta$, where $\Delta$ is ($\neg \phi$) expressed in CNF (conjunctive normal form) which are nodes of $T$, and we start a branch so that for $1 \leq i \leq n$, $C_{i-1} \rightarrow C_i$.

There are two rules for extending a branch, where the premises must occur earlier in the branch. Figure 9 gives these two rules where those clauses above the line indicate the premises needed to use the rule, and the clauses below indicate the extensions. Note \{a_j\}|{\neg a_j}\} indicates that the branch must divide, with $u_m \rightarrow \{a_j\}$ and $u_m \rightarrow \{\neg a_j\}$.

\[
\begin{array}{l}
\{a_1, a_2, ..., a_m, \neg b_1, \neg b_2, ..., \neg b_n\}, \{b_1\}, \ldots, \{b_n\}, \{\neg a_j\}\} \\
\{\neg a_1\}, \ldots, \{\neg a_{j-1}\}, \{\neg a_{j+1}\}, \ldots, \{\neg a_m\}\} (N1) \\
\{a_1, a_2, ..., a_m, \neg b_1, \neg b_2, ..., \neg b_n\}, \{b_1\}, \ldots, \{b_n\}, \{a_j\}|{\neg a_j}\} (N2)
\end{array}
\]

**Fig. 9. Rules of Niemelä’s Tableau [21]**

**Definition 40. (Niemelä [21])**
- We say a branch $B$ is closed when for some atoms $b_i$, clauses $\{\neg b_1\}, \ldots, \{\neg b_n\}, \{b_1\}, \ldots, \{b_n\}$, occur in the same branch.
- $N_T(B) = \{\neg c\} | c \text{ is an atom, } \{c\} \text{ does not occur in } B, \exists C \in \Gamma \text{ s.t. } c \in C\}$.
- We say a branch $B$ is ungrounded when $B$ contains a singleton clause of a positive atom $\{a\}$, for which $N_T(B), \Gamma \not\models a$.
- We say a branch is MM-closed if is either ungrounded or closed.

**Theorem 41. (Niemelä [21])** There is a $\Gamma, \phi$-tableau with all its branches MM-closed if and only if $\Gamma \models_M \phi$.

**Theorem 42. (Bonatti, Olivetti [6])** CIRC polynomially simulates Niemelä’s Tableau Calculus.

**Corollary 43.** MLK polynomially simulates Niemelä’s Tableau Calculus.

**Proof.** We use the transitivity of simulation on theorems 37 and 42. \qed

**Corollary 44.** Niemelä’s Tableau Calculus does not polynomially simulate MLK.

**Proof.** Suppose it does, then by transitivity CIRC polynomially simulates MLK, which contradicts theorem 35. \qed

**Corollary 45.** Let us denote the tableau calculus by the name TAB. Then

$$s_{TAB}(n) \in \Omega(2^n)$$

**Proof.** We use the same hard examples as in theorem 23. If these had shorter length, then they would be proved shorter than exponential length in the CIRC calculus using the translation in 42, but theorem 23 shows that this is not the case. Hence they are exponentially bounded below. \qed
13 Quantified Propositional Formulae

This sections looks at relating the circumscription problem to an extension of classical logic and in doing so we can build a new calculus to potentially replace CIRC.

In quantified propositional formula we change our logical language to include quantifiers. Now $\Sigma_{\text{Conn}} = \{\bot, \top, \neg, \to, \lor, \land, \forall, \exists\}$ and our alphabet is defined as usual $\Sigma = \Sigma_{\text{Prop}} \cup \Sigma_{\text{Conn}}$.

**Definition 46.** A finite string $\phi$ in $\Sigma^*$ is a quantified propositional formula when

1. $\phi = \bot$.
2. $\phi = \top$.
3. $\phi \in \Sigma_{\text{Prop}}$.
4. $\phi = (\neg \chi)$, and $\chi$ is a quantified propositional formula.
5. $\phi = (\chi \to \psi)$, and $\chi$ and $\psi$ are quantified propositional formulae.
6. $\phi = (\chi \lor \psi)$, and $\chi$ and $\psi$ are quantified propositional formulae.
7. $\phi = (\chi \land \psi)$, and $\chi$ and $\psi$ are quantified propositional formulae.
8. $\phi = (\forall x(\chi))$, and $\chi$ is a quantified propositional formula, $x \in \Sigma_{\text{Prop}}$, and the strings $\forall x$ or $\exists x$ are not substrings of $\chi$.
9. $\phi = (\exists x(\chi))$, and $\chi$ is a quantified propositional formula, $x \in \Sigma_{\text{Prop}}$, and the strings $\forall x$ or $\exists x$ are not substrings of $\chi$.

1. If $\phi \in \Sigma_{\text{Prop}}$, then $M \models \phi$ if and only if $\phi \in M$.
2. If $\phi = \bot$, then $M$ does not model $\phi$.
3. If $\phi = \top$, then $M \models \phi$.
4. If $\phi = (\chi \to \psi)$, then $M \models \phi$ if and only if either $M \models \psi$ or $M$ does not model $\chi$.
5. If $\phi = (\neg \chi)$, then $M \models \phi$ if and only if $M \models (\chi \to \bot)$.
6. If $\phi = (\chi \lor \psi)$, then $M \models \phi$ if and only if $M \models (\neg(\neg \chi \land \neg \psi))$.
7. If $\phi = (\chi \land \psi)$, then $M \models \phi$ if and only if either $M \models (\chi(\bot) \land \chi(T))$.
8. If $\phi = (\exists x(\chi(x)))$, then $M \models \phi$ if and only if $M \models (\chi(\bot) \lor \chi(T))$.
9. If $\phi = (\forall x(\chi(x)))$, then $M \models \phi$ if and only if $M \models (\chi(\bot) \land \chi(T))$.
10. If $\Delta = \{\phi_i : 1 \leq i \leq k\}$, then $M \models \Delta$ if and only if for all $i : 1 \leq i \leq k$; $M \models \phi_i$.

**Fig. 10.** Inference rules of the extension of the $\text{LK}$ calculus to Quantified Propositional Formulae [17]

**Theorem 47.** Let $P = \{p_i | 1 \leq i \leq n_P\}$ and $P = \{r_i | 1 \leq i \leq n_R\}$

then $\Gamma \models p_i$ if and only if

$\top \models (\forall p_1 \ldots (\forall p_{n_P} (\forall r_1 \ldots (\forall r_{n_R} (\exists p'_1 \ldots (\forall r'_1 ((\land \Gamma \to p_i) \lor (((\land_{i=1}^{n_P} (p'_i \to p_i)) \land (\land_{i=1}^{n_R} (r'_i \to r_i))))) \land (\land_{i=1}^{n_P} (p'_i \to p_i)))) \land (\land_{i=1}^{n_R} (r'_i \to r_i)))))) \lor ((\land_{i=1}^{n_P} (p'_i \to p_i)) \land (\land_{i=1}^{n_R} (r'_i \to r_i))))$
Proof. Suppose \( \Gamma \models_{P,R} \phi \), when we universally quantify with \( \forall \) over the variables \( p_i,r_i \), what is being done is that we are checking every model \( M \), in the case that it does not satisfy \( \Gamma \), then \( (\forall \Gamma) \rightarrow \phi \) is true. In the case it is minimal (and satisfies \( \Gamma \)) then \( (\forall \Gamma) \rightarrow \phi \) is true. In the case that it satisfies \( \Gamma \) but is not minimal we know that there is some model \( N \) with \( N \models_{P,R} M \), \( N \models \Gamma \) and \( M \not\models_{P,R} N \). We assume our sequence of existential quantifiers finds values of our variables \( p_i',r_i' \) that agree with model \( N \). Since \( N \models_{P,R} M \), the models musts agree on the variables in \( R \), hence for every \( r \in R \), \((r_i \land r_i') \lor (\neg (r_i \lor r_i'))\) so we know the conjunction \( (\bigwedge_{i=1}^{n} (r_i \land r_i')) \lor (\bigwedge_{i=1}^{n} (\neg (r_i \lor r_i'))) \) is true. We know \((p_i' \rightarrow p_i)\) as this is given by \( N \cap P \subseteq M \cap P \), this allows us to have the conjunction \( (\bigwedge_{i=1}^{n} (p_i' \rightarrow p_i)) \). \((\forall \Gamma[p_i'/p_i][p_{n'}' / p_{n'}][r_i' / r_i'][r_{n'}' / r_{n'}]) \) is true since \( N \) must satisfy \( \Gamma \). Because \( M \not\models_{P,R} N \) then there is some \( p \in P \) such that \( p \) is true in \( M \) and \( p \) is false in \( N \), this is expressed as \((\neg p_i') \land p_i)\) we do not recognise which variable it is but the disjunction; \((\bigvee_{i=1}^{n} (\neg p_i') \land p_i)\) is true regardless. Hence our quantified formula is always true. Because there are no additional restrictions on the existentially quantified variables it will be the case that they indeed form our model \( N \).

Suppose instead \( \Gamma \not\models_{P,R} \phi \), that means some minimal model \( M \) of \( \Gamma \) does not satisfy \( \phi \), hence \( (\forall \Gamma) \rightarrow \phi \) is false. Given an assignment of the variables in neither \( P \) nor \( R \) the existential variables satisfying \((\forall \Gamma[p_i'/p_i][p_{n'}' / p_{n'}][r_i' / r_i'][r_{n'}' / r_{n'}]) \) is equivalent to some model \( N \) they produce satisfying \( \Gamma \). If there were to exist some model \( N \) of \( \Gamma \) with \((\bigwedge_{i=1}^{n} (p_i' \rightarrow p_i)) \land (\bigwedge_{i=1}^{n} (r_i \land r_i') \lor (\neg (r_i \lor r_i'))) \) then \( N \models_{P,R} M \), which would require that \( M \models_{P,R} N \), hence falsifying \((\bigvee_{i=1}^{n} (\neg p_i') \land p_i)) \). So the quantified statement has to be false.

This translation of sequents for circumscription to QPF, allows us to develop a new calculus. The proof of the above translation (which is polynomially bounded) in our extended \( LK \) is a sound and complete calculus. One thing to note is that the quantifier alternation \( \forall \) then \( \exists \) is similar to the quantifiers used in the definition of \( H_2^M \) from definition 15, which is appropriate as circumscription is a problem which is in \( H_2^M \).

14 Conclusions

The minimal objectives of this project were to:

1. Understand and analyse the proof complexity of circumscription.
2. Compare the complexity of theorem proving for circumscription to the complexity of theorem proving in classical logic and in other non-monotonic logics.

Theorem 23 concludes objective 1, with a lower bound. Objective 2 was satisfied in the theorems and discussions of section 10.

The project was extended to begin a preliminary investigation into alternatives in \( CIRC \), this progressed to mainly finding a simulation of \( CIRC \) by \( MLK \) for minimal entailment in theorem 37, and finding a new strategy, using QPF, for a complete calculus for circumscription in theorem 47.

Once the minimal objectives were completed, I was given the option of possible enhancement that could be achieved through tasks:

1. Find short proofs for the hard examples in \( MLK \).
2. Find short proofs for the hard examples using the QPF translation.
3. Investigate whether \( MLK \) simulates \( CIRC \) for minimal entailment.
4. Investigate whether \( MLK \) \( p \)-simulates \( CIRC \) for minimal entailment.
5. Investigate whether \( MLK \) simulates \( CIRC \) in general.
6. Investigate whether \( MLK \) \( p \)-simulates \( CIRC \) in general.
7. Investigate the proof complexity of \( MLK \).
8. Investigate the proof complexity of the calculus resulting from a translation to QPF formulæ.
9. Investigate the proof complexity of the tableau calculus.

Tasks 1 is resolved with theorem 35. Task 3 is resolved with theorem 37. Task 9 is resolved in section 12. The remaining tasks remain unresolved. Significant progress has been made on task 4 as seen in remark 38.

18
15 Evaluation of Results

A large advantage of evaluating these results is that we have used mathematical proofs to gain and verify our results, if a proof is sound and has the desired conclusion then the result is indisputably true. While it is not the case that proofs can be used in every project, the reason it can be used here is that we are dealing with formally defined theoretical objects, in fact because we had the option of using mathematical proofs, no other options for evaluating the correctness of the results were used, as each other method would have a degree of uncertainty which would be avoided by using proofs.

Several new results have been introduced in this report, theorems 23, 35, 37 and 47 in particular. While many of the proofs given here are subject to review, this evaluation will not comment on the soundness of these proofs as if there is any indication that they are unsound or inadequate it should be addressed in the proof itself by adding rigour. There is some question on whether my results are truly falsifiable; consider the case that I make an invalid proof on a theorem, yet neither the theorem nor its negation is provable, then it is literally not falsifiable. In reality the best way to avoid this is to produce valid proofs, which I have made an effort to do. What is definitely true about my work is that it is verifiable; the proofs can be checked and verified or rejected with some ease. Ideally we would have a polynomial time program that could check my work, but given the state of my work it would have to deal with natural language.

I have avoided using heavy formalism in my own proofs as to distinguish my proofs from the proofs in our calculi that we define and investigate as mathematical objects. As such, I have not verified my work using any proof checking software.

In this evaluation section we can look at the profoundness of each results. Theorem 23 is not surprising, given the small number of rules of \textit{CIRC} and the exponential branching algorithm used in the proof of theorem 22 given in Bonatti and Olivetti’s 2002 paper [6]. We have not placed any upper bound on \textit{CIRC} as \textit{CIRC} requires the use of \textit{LK} and it is unknown whether \textit{LK} has a polynomial upper bound or not. It is also unknown whether \textit{CIRC} necessarily requires an exponential number of instances of (C2) in some schemes of proofs as we can use the antecedent calculus and the rule (C1) to infer circumscription sequents as well. Furthermore it isn’t yet known that if we need to use (C2) on an exponential number of classical sequents whether we could prove most of the classical sequents using many of the same lines of proofs.

Due to this exponential branching we should also not be surprised that \textit{CIRC} does not simulate \textit{MLK}. \textit{CIRC} makes heavy use of the branching and the hard examples found were not particularly complicated and made limited use of the connectives. It is no surprise that a calculus that can deal with conjunctions specifically using rule (M) can quickly prove our examples. It is of note that \textit{MLK} was specifically designed to deal with minimal entailment (it can be extended to deal with circumscription properly), whereas \textit{CIRC} deals with the full circumscription. While we have shown that \textit{MLK} polynomially simulates \textit{CIRC} for minimal entailment, we have not shown it true in the general case, so it may be that \textit{MLK} has some pitfall when dealing with fixed and variable atoms.

The idea behind the QPF translation was to experiment with alternative calculi to \textit{CIRC} for circumscription. However, I have not proved any results on its proof complexity. It would interesting to know whether it can simulate or be simulated by \textit{MLK}.

Due to time constraints and supervisor availability I was limited in the number of investigations outside of the minimal aims that I could complete. We have not yet proved an \textit{MLK} simulation of \textit{CIRC} for general circumscription. Additionally, we proved a lower bound for \textit{CIRC}, but we did not yet find a calculus for circumscription that had an upper bound (in terms of \textit{LK}), but we have begun the process of studying the proof complexity of \textit{MLK}.

16 Further Work

I fully aim to continue my work on circumscription and possibly broaden to other areas of proof complexity and nonmonotonic logic as a PhD student. Before then, I will present some of the contents of this projects in September as a presentation for the British Logic Colloquium Postgraduate Day.

The first, most basic extension would be to find a specific algorithm for finding the p-simulation of \textit{CIRC} by \textit{MLK} for minimal entailment.

I aim to look for calculi that could simulate \textit{CIRC} in general. We already have found the QPF translation. The QPF calculus in general has been around for a few decades, and to the best of my knowledge there still
is work to do on its proof complexity. So one option is to look into the proof complexity of calculi of QPF. Another option is to look at the extended MLK that allows us to deal with fixed and variable atoms and see if that simulates CIRC. If it does not then introducing the rules of the extended MLK to CIRC would allow the new calculus to simulate both.

One thing in particular would be to find an upper bound in terms of LK of some circumscription calculus, either the extended MLK, the QPF calculus for translated sequents, or some other calculi that can be thought up. Comparing it to the work of Beyersdorff, where the sceptical calculi were found to have lower bounds but not upper bounds in terms of LK [1] [3], it may a much more difficult task, and until proven it may be impossible.

References

A  Undertaking the Project

During the course of this project I have for the first time proven new results, particularly the results that we were initially interested in. However there are many more questions to be answered and there was always more potential, but it was decided by myself and the supervisor that after the progress meeting we should stop looking for new results and focus on writing them up, this would mean that a few of the tasks that had been discussed in the interim report were postponed until after the submission of this project report.

I have used this project to once again familiarise myself with formal logic, particular learning more about it from a theoretical computer science/complexity point of view, and sensibly bring together material from different lecturers in a consistent manner.

In order to satisfy minimal requirements I undertook the task of finding and proving theorem 23. I chose to first look for a lower bound for two reasons; firstly because the proof of the completeness theorem of CIRC explicitly used an exponential branching algorithm for results, secondly, it was the case that finding a lower bound allowed me to work with examples in CIRC, which at this early stage was helpful for me to understand CIRC even if a result was not proved. Since I did a surplus of modules in semester 1, I had a lot more spare time in semester 2, I used this to tackle the problem as quickly as possible, and it was a strategy that was effective, as I had a mathematical proof long before I had to worry about the final deadline for any changes to the minimum requirements.

For theorem 47, I initially thought of using an algorithm to decide a circumscription sequent and then apply Cook’s theorem to get a propositional formula, however Dr. Beyersdorff guided me to find a more natural formula, and I did find one; although it is still long in some respects, but it does follow from our definition of circumscription.

I used the proof of the completeness theorem of MLK [22] to realise that it worked by explicitly finding a model, I took a similar idea in the simulation of CIRC, using the model explicitly and assuming that in simple schemes of examples LK has polynomial time. I later proved the lemmas needed for the simulation proof.

One of the most difficult parts in the project, was in fact choosing a title as I spent a considerable amount of time seeking an attempt to do a different project on computational homology, negotiating with various potential supervisors, I found myself needing to compromise in order to be able to undertake the homology project. When looking at the opportunity to do this project, which was a supervisor defined project, I found myself able to have more flexibility than I anticipated as long as I satisfied agreed upon criteria. It also gave me an opportunity to prove a new result, something that would have been impossible if I had defined my own project. I advise future students not to underestimate the freedom given by doing a supervisor defined project.

The feedback using my interim report from both my supervisor and assessor was very helpful, I had written up my mathematical proofs in the same style as I would write the answers to an example sheet for a mathematics module. However, even though the proofs were sound they were difficult to understand. The problem I believe is that while that style may be appropriate for an example sheet, example sheet assessors usually have a mark scheme or model answers which would help them read and understand my work. When conveying original results, there are limited additional resources to help understand the solutions, so my proof needed more explanation. I also understand that when presenting a proof, one has multiple options and if clarity comes at no cost to soundness, then it only makes sense to optimise clarity.

A valuable skill that I have learned is now being able to write reports using LaTeX which is an improvement over using Microsoft Office applications. I did not however, make use of BiBTeX, instead manually converting the references; this is something I aim to learn in future reports. I would recommend LaTeX to future students, as the abilities to control contents pages, appendices and the page numbering system required for this report is very implementable in LaTeX.

B  Using the Work and Help of Others

Figure B details how I used each specific item in the references section.

For section 4 I have mainly developed my own notation based on the lecture courses that I have attended on logic. I have not stuck firmly to one particular course as all of the courses had limitations. Logic and Set Theory use posets and lattices to semantically define truth (an extraneous definition I did not need),
whereas Advanced Logic did not include constant symbols, for algorithm design we were mainly focused on clausal theories.

Many of the definitions of the calculi, in the figures 4, 5, 6, 8 and 10 are based on similar figures in their referenced papers, although edited to make sense in the context of my own definitions.

I used LaTeX to generate my report. I also used a LaTeX software package from reference [18], which was used for building a Gantt chart for my interim report. As this is the first time I have used LaTeX, I had a lot of help from Dr. Beyersdorff who sent me the llncs.cls class file, autoepistemic.sty file and autoepistemic.tex file which I modified to produce my paper. With his consent I used the references section, for which I used many of the papers we shared and used them as a starting point for the building of my project.

C Ethical Issues

I have not used participants or medical data in my project. As this project is very theoretical there is not much to mention in relation to ethical issues.

Since I am studying complexity and the polynomial hierarchy there is always a risk that if I prove an unexpected result the polynomial hierarchy may collapse, it being done in a constructive manner may cause problems with financial security and privacy. However I have not proven any results like this, and I would not expect to, if I did find any result like this I would increase the security of my document and not share it with anyone other than my supervisor (who would most likely just explain why it is false).

D Extra Proofs

D.1 Proof of Lemma 24

Proof. We use induction on $|\phi|$

Induction Hypothesis $s_{LK}(\phi \vdash \phi) \leq 4|\phi|^2 + 3$
Base Case: If $|\phi| = 1$ then either $\phi$ is atomic, $\phi = \bot$ or $\phi = \top$.
Because $|\phi| = 1$ we are interested in bounding above by 7. If $\phi$ is atomic, the proof is just an instance of $(\vdash)$.

$$\phi \vdash \phi \quad (\vdash)$$

So $s_{LK}(\phi \vdash \phi) = 3$. This is bounded above by 7.
If $\phi = \bot$ or $\phi = \top$ then $s_{LK}(\phi \vdash \phi)$ is constant.

$$\bot \vdash \bot \quad (\vdash \bot)$$
$$\bot \vdash \bot \quad (\bot \vdash \bot)$$
$$\bot \vdash \bot \quad (\bot \bot)$$

So $s_{LK}(\phi \vdash \phi) \leq 7$ in these cases. This is bounded above by 7, our inductive hypothesis holds for the base case.

Inductive Case:
We consider two cases, first is the fact that we may have infinitely many atoms, but since we can only have finitely many characters in our alphabet, we need to start expressing our atoms as strings that have length greater than one.
If $|\phi| > 1$ then if $\phi$ is atomic we have a proof:

$$\phi \vdash \phi \quad (\vdash)$$

$s_{LK}(\phi \vdash \phi) \leq 2 + 2|\phi|$
this satisfies our inductive hypothesis as for $|\phi| > 1$, $|\phi| \leq |\phi|^2$. Hence,

$$s_{LK}(\phi \vdash \phi) \leq 2 + 2|\phi| \leq 4|\phi|^2 + 3.$$ 

Now we observe the main inductive case where connectives are used, to build larger formulae. The aim here is to show building with any connective is bounded above by $s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5$. Where $\chi$ and $\psi$ are formulas to be used in the connectives (they may not appear in all the connectives but still provide an upper bound as we take their lengths and proof length to be non negative).
If $\phi = (\neg \chi)$;
Then we have a proof of $\chi \vdash \chi$ which gives us length $s_{LK}(\chi \vdash \chi)$.
Then to finish the proof

$$\chi \vdash \chi \quad (\vdash \neg)$$

$$\neg \chi \vdash \chi \quad (\neg \bot)$$

This proof is of length $s_{LK}(\chi \vdash \chi) + |\chi| + 3|\phi| + 2 + 1 + 2$.
And note for later that it is bounded above by $s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5$.
If $\phi = (\chi \lor \psi)$;
Then we have a proof of $\chi \vdash \chi$ which gives us length $s_{LK}(\chi \vdash \chi)$.
And we have a proof of $\psi \vdash \psi$ which gives us length $s_{LK}(\psi \vdash \psi)$.

$$\chi \vdash \chi \quad (\lor \bot)$$

$$\psi \vdash \psi \quad (\bot \bot)$$

This proof is of length $s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + |\chi| + |\psi| + 4|\phi| + 6$.
We simplify to $s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 5|\phi| + 3$.
And note for later that it is bounded above by $s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5$.
Similarly, If $\phi = (\chi \land \psi)$;
Then we have a proof of $\chi \vdash \chi$ which gives us length $s_{LK}(\chi \vdash \chi)$.
And we have a proof of $\psi \vdash \psi$ which gives us length $s_{LK}(\psi \vdash \psi)$. 

23
This proof is of length \( s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + |\chi| + |\psi| + 4|\phi| + 6 \).
And note for later that it is bounded above by \( s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5 \).
Finally, if \( \phi = (\chi \rightarrow \psi) \);
Then we have a proof of \( \chi \vdash \chi \) which gives us length \( s_{LK}(\chi \vdash \chi) \).
And we have a proof of \( \psi \vdash \psi \) which gives us length \( s_{LK}(\psi \vdash \psi) \).

This proof is of length \( s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 4|\chi| + 4|\psi| + 3|\phi| + 3 + 8 \).
We simplify to \( s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5 \).
Firstly we note that in the inductive connective case \( |\phi| \geq 4 \).
In every inductive case using a connective we form a universal upper bound.
\[ s_{LK}(\phi \vdash \phi) \leq s_{LK}(\chi \vdash \chi) + s_{LK}(\psi \vdash \psi) + 7|\phi| + 5. \]
\[ \leq 4|\chi|^2 + 3 + 4(|\phi| - |\chi| - 3)^2 + 3 + 7|\phi| + 5. \]
\[ = 4|\phi|^2 - 24|\phi| + 36 - 8(|\chi|)(|\phi| - |\chi| - 3)| + 7|\phi| + 11 \]
\[ \leq 4|\phi|^2 - 17|\phi| + 47 \leq 4|\phi|^2 - 5|\phi| \leq 4|\phi|^2 + 3 \]
So true in the inductive case.
Hence true for all well formed formulae \( \phi \). \( \square \)

### D.2 Proof of Lemma 25

**Proof.** The inductive hypothesis in the number of connectives of \( \phi \) will be that the proof length (now denoted as \( S(\phi) \)) of either \( \Sigma^+ \), \( \Sigma^- \vdash \phi \) or \( \Sigma^+ \), \( \Sigma^- \vdash (\neg \phi) \) is bounded above by polynomial \( w^3 + 7 \) with \( w = |\Sigma| + |\phi| \).
Firstly let \( \Sigma = \Sigma^+ \cup \neg \Sigma^- \).
Firstly let us consider individual formula, here \( \phi \) will be a well formed formula in \( \Gamma \cup \Delta \), because we have only one model either \( \Sigma' \vdash \phi \) or \( \Sigma' \vdash (\neg \phi) \).
If \( \phi = \top \) then \( \Sigma' \vdash \phi \). We use the proof
\[
\frac{\top \vdash \neg \top}{\Sigma \vdash \neg \top} \text{ repeated (linearly bounded) use of } (\bullet \vdash)
\]

We can use this one as an example for how to count the number of characters, although we will be a little crude in our upper bounds. We have a proof that uses linearly bounded repetition. In general we have a sequence of terms of the form \( |\Sigma' \vdash \top| \). The first term \( a_1 \) is \( |\vdash \top| \) which is (including the separator) \( 3 \).
Then we consider the difference \( d = 2 \) because each term increases by at most \( 5 \) characters (proposition, two parentheses connectives and a comma). We are not concerned if a new formula requires more propositions as we will encompass every size as long as we repeat this \( w \) number of times. So using the arithmetic series formula is \( \left\lceil \frac{w}{2} \right\rceil (2a_1 + (w - 1)5) \).
The proof length is bounded above by \( 3 + 3w^2 \), which for \( w \geq 1 \) is bounded above by \( 9w^3 + 7 \).
If \( \phi = \bot \) it can be proved false
\[
\frac{(\bot \vdash \emptyset)}{(\bot \vdash \emptyset) \text{ repeated (linearly bounded) use of } (\bullet \vdash)}
\]
Using known results on arithmetic series, the proof length is bounded above by $13 + 3w^2$, which for $w \geq 1$ is bounded above by $9w^3 + 7$.

If $\phi$ is atomic and true then $\phi \in \Sigma^+$, $\vdash \phi$ is obtained via axiom ($\vdash$), the remaining LHS is obtained via ($\bullet \vdash$).

Using known results on arithmetic series, the proof length is bounded above by $\frac{w}{2}(5w + 3)$, which for $w \geq 1$ is bounded above by $9w^3 + 7$.

If $\phi$ is atomic and false then $\phi \in \Sigma^-$,

$$
\frac{\phi \vdash \phi}{\emptyset \vdash \phi, (\neg \phi)} (\vdash \neg) \\
\frac{\emptyset \vdash \phi, (\neg \phi)}{(\neg \phi) \vdash (\neg \neg) (\neg \vdash)}
$$

and then by weakening with ($\bullet \vdash$) the right hand side.

Using known results on arithmetic series, the proof length is bounded above by $12 + \frac{w}{2}(5w + 3)$, which for $w \geq 1$ bounded above by our universal bound $9w^3 + 7$.

Hence our inductive hypothesis is true for $|\phi| = 1$.

Inductively on number of characters of $\phi$, we seek to prove a universal bound that can be found for all the inductive cases $S(\chi) + S(\phi) + 33 + 15w + 9w^2$.

- If $\phi = (\neg \chi)$ then if $\phi$ is true in the model then $\chi$ is false so it is already done by the proof of the negation of $\chi$.
  So it is bounded above by $S(\chi)$, which for $w \geq 1$ bounded above by our universal bound $S(\chi) + S(\psi) + 33 + 15w + 9w^2$.

- If $\phi = (\neg \chi)$ and if $\phi$ is false in the model then $\chi$ is true so $\Sigma \vdash \chi$ (repeated (linearly bounded) use of ($\bullet \vdash$)) and we can simply add this to the end of the proof.
  So it is bounded above by $S(\chi) + 2w + 5$ which is bounded above by our universal bound $S(\chi) + S(\psi) + 33 + 15w + 9w^2$.

- If $\phi = (\chi \lor \psi)$ and if $\phi$ is true in the model then either $\Sigma \vdash \chi$ or $\Sigma \vdash \psi$. In the case that $\Sigma \vdash \chi$ we can use our inductive hypothesis. Then proceed as follows;

$$
\frac{\Sigma \vdash \chi}{\Sigma \vdash \phi} (\vdash \lor \bullet)
$$

In the case that $\Sigma \vdash \psi$ we can use our inductive hypothesis. Then proceed as follows;

$$
\frac{\Sigma \vdash \psi}{\Sigma \vdash \phi} (\vdash \bullet \lor)
$$

The proof length is bounded above by $S(\chi) + S(\psi) + 2 + w$, which is bounded above by our universal bound $S(\chi) + S(\psi) + 33 + 15w + 9w^2$.

- If $\phi = (\chi \lor \psi)$ and if $\phi$ is false in the model, then $\Sigma \vdash (\neg \chi)$ and $\Sigma \vdash (\neg \psi)$. We use the proof below to obtain sequents $\Sigma, \chi \vdash$ and $\Sigma, \psi \vdash$ similarly. But we need to start from $\chi \vdash \chi$ and $\psi \vdash \psi$ using the short proofs in the previous lemma

$$
\frac{\chi \vdash \chi}{\chi, (\neg \chi) \vdash (\neg \neg)} (\vdash \neg) \\
\frac{\chi \vdash (\neg (\neg \chi))}{\Sigma, \chi \vdash (\neg (\neg \chi))} \quad \text{repeated (linearly bounded) use of ($\bullet \vdash$)} \\
\frac{\Sigma \vdash (\neg (\neg \chi))}{\Sigma, \chi \vdash (\neg (\neg \chi))} (\vdash \neg) \\
\frac{\Sigma, \chi \vdash (\neg (\neg \chi))}{\Sigma, \chi \vdash (\neg (\neg \chi))} (\vdash \neg)
$$
We then do similarly for \( \psi \) and use both to complete the proof.

\[
\frac{\Sigma, \chi \vdash \Sigma, \psi \vdash (\lor \vdash)}{\Sigma, (\chi \lor \psi) \vdash (\lnot \vdash)}
\]

Using known results on arithmetic series, the proof length is bounded above by \( S(\chi) + S(\psi) + 30 + 22w + \frac{17}{2} w^2 \), which is bounded above by our universal bound \( S(\chi) + S(\phi) + 33 + 15w + 9w^2 \).

- If \( \phi = (\chi \land \psi) \) and if \( \phi \) is true in the model, then \( \Sigma \vdash \chi \) and \( \Sigma \vdash \psi \) as with a proof length bounded by our inductive hypothesis, the proof for \( \psi \) follows as;

\[
\frac{\Sigma \vdash \chi}{\Sigma \vdash \psi} (\lnot \vdash)
\]

The proof length is bounded above by \( S(\chi) + S(\psi) + w + 3 \), which is bounded above by our universal bound \( S(\chi) + S(\phi) + w + 33 + 15w + 9w^2 \).

- If \( \phi = (\chi \land \psi) \) and if \( \phi \) is false in the model, then either \( \Sigma \vdash (\lnot \chi) \) or \( \Sigma \vdash (\lnot \psi) \). Without loss of generality if \( \Sigma \vdash (\lnot \chi) \), then we use its bounded proof via the induction hypothesis then we use the bounded length proofs of \( \chi \vdash \chi \) from the previous lemma and proceed as follows,

\[
\frac{\chi \vdash \chi}{\chi, (\lnot \chi) \vdash (\lnot \vdash)} \quad \frac{\Sigma, \chi \vdash (\lnot(\lnot \chi)) \text{ repeated (linearly bounded) use of } (\lnot \vdash)}{\Sigma, (\chi \lor \psi) \vdash \psi} (\lnot \vdash)
\]

Using known results on arithmetic series, the proof length is bounded above by \( S(\chi) + S(\psi) + 3 + 14w + \frac{33}{2} w^2 \), which is bounded above by our universal bound \( S(\chi) + S(\psi) + 33 + 15w + 9w^2 \).

- If \( \phi = (\chi \rightarrow \psi) \) and \( \phi \) is true in the model, then either \( \Sigma \vdash \lnot \chi \) or \( \Sigma \vdash \psi \). If \( \Sigma \vdash \lnot \chi \) we use the short proof from the induction hypothesis and the short proof of \( \chi \vdash \chi \) from the previous lemma to proceed as follows;

\[
\frac{\chi \vdash \chi}{\chi, (\lnot \chi) \vdash (\lnot \vdash)} \quad \frac{\Sigma \vdash (\lnot \psi)}{\Sigma, \chi \vdash (\lnot(\lnot \chi)) \text{ repeated (linearly bounded) use of } (\lnot \vdash)} \quad \frac{\Sigma \vdash (\lnot \chi)}{\Sigma, (\chi \lor \psi) \vdash \psi} (\lnot \vdash)
\]

If instead \( \Sigma \vdash \psi \) we use the short proof of it from the inductive hypothesis and proceed as follows;

\[
\frac{\Sigma \vdash \psi}{\Sigma, \chi \vdash \psi} (\lnot \vdash)
\]

Using known results on arithmetic series, in either case it is bounded above by \( S(\chi) + 29 + 15w + \frac{17}{2} w^2 \), which is bounded above by our universal bound \( S(\chi) + S(\psi) + 33 + 15w + 9w^2 \).

- If \( \phi = (\chi \rightarrow \psi) \) and \( \phi \) is false in the model, then \( \Sigma \vdash \chi \) and \( \Sigma \vdash \lnot \psi \). We use the short proof of \( \phi \vdash \phi \) again in the same way;

\[
\frac{\Sigma \vdash \psi}{\Sigma, \chi \vdash \psi} (\lnot \vdash)
\]
And finish the proof as follows;

\[
\frac{\Sigma, \psi \vdash \Sigma' \vdash \chi \quad \Sigma' \vdash \neg \phi}{\Sigma \vdash \neg \phi}
\]

Using known results on arithmetic series, this is bounded above by \(S(\chi) + S(\psi) + 32 + 13w + \frac{17}{2}w^2\), which is bounded above by our universal bound \(S(\chi) + S(\psi) + 33 + 15w + 9w^2\).

We have shown the universal bound \(S(\chi) + S(\psi) + 33 + 15w + 9w^2\) holds in the inductive connective case.

We use the induction hypothesis to derive the fact \(S(\chi) + S(\psi) \leq 9(|\phi| - 1)^3 + 14\).

We now substitute this into the upper bound and then proceed to use the fact that \(\phi \geq 4\)

\[
S(\phi) \leq 9w^3 - 18w^2 + 42w + 38 \leq 9w^3 - 30w + 38 \leq 9w^3 - 82 \leq 9w^3 + 7
\]

Hence the inductive hypothesis is true for the inductive case. In order to complete the proof for \(\Sigma, \Gamma \vdash \Delta\) we can use the \((\vdash \bullet), (\bullet \vdash)\) for weakening. In the case that the left hand side is inconsistent because \(\Sigma \vdash \neg \phi\) for some \(\phi \in \Gamma\) it can be done quickly by using the \((\neg \vdash)\) rule to bring over \((\neg \psi)\) and change this into \(\phi\) using the same trick that we have done in many of the proofs above. The remainder follows from weakening.

Let \(z = |\Sigma, \Gamma \vdash \Delta|\). Adding these operations, using known results on arithmetic series, gives an upper bound of \(9z^3 + 9z^2 + 14z + 34\).

\(\square\)