Proof Complexity of Modal Resolution Systems

by

Sarah Elizabeth Sigley

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computing
September 2019

Declarations

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in this thesis have been published in the following articles:

- Sarah Sigley. *Resolution Calculi for Modal Logic and their Relative Proof Complexity.* In *Proceedings of the ESSLLI 2017 Student Session*, pages 60–72. 2017.

Some parts of the work presented in this thesis have been submitted for publication in the following articles:

- Sarah Sigley, Olaf Beyersdorff. *Proof Complexity of Modal Resolution.* Submitted to *Journal of Automated Reasoning*.

Abstract

In this thesis we initiate the study of the proof complexity of modal resolution systems. To our knowledge there is no previous work on the proof complexity of such systems. This is in sharp contrast to the situation for propositional logic where resolution is the most studied proof system, in part due to its close links with satisfiability solving.

We focus primarily on the proof complexity of two recently proposed modal resolution systems of Nalon, Hustadt and Dixon, one of which forms the basis of an existing modal theorem prover. We begin by showing that not only are these two proof systems equivalent in terms of their proof complexity, they are also equivalent to a number of natural refinements. We further compare the proof complexity of these systems with an older, more complicated modal resolution system of Enjalbert and Fariñas del Cerro, showing that this older system p-simulates the more streamlined calculi.

We then investigate lower bound techniques for modal resolution. Here we see that whilst some propositional lower bound techniques (i.e. feasible interpolation) can be lifted to the modal setting with only minor modifications, other propositional techniques (i.e. size-width) fail completely. We further develop a new lower bound technique for modal resolution using Prover-Delayer games. This technique can be used to establish "genuine" modal lower bounds (i.e lower bounds on the number of modal inferences) for the size of tree-like modal resolution proofs. We apply this technique to a new family of modal formulas, called the modal pigeonhole principle to demonstrate that these formulas require exponential size modal resolution proofs.

Finally we compare the proof complexity of tree-like modal resolution systems with that of modal Frege systems, using our modal pigeonhole principle to obtain a "genuinely" modal separation between them.

Acknowledgements

I would firstly like to thank my main supervisor Olaf Beyersdorff, who has provided helpful suggestions and guidance throughout the course of my studies. I am also grateful to Kristina Vuskovic for helping supervisor me in my final year, as well as all the members of staff in the computer science department who have assisted me at various points throughout my PhD.

I would also like to acknowledge the friends I have met during my time at Leeds, both in the computer science department and beyond. In particular I want to say thank you to Richard Whyman for reading numerous drafts of my thesis and providing invaluable encouragement and support whilst I was writing up.

Finally, I would like to thank my family for always assuring me that there is still a lot that I do not know.

Contents

List of Figures

Index

Chapter 1

Introduction

Proof complexity In proof complexity we analyse how efficiently theorems can be proved in a given proof system, where generally speaking, a proof system is a finite collection of inference rules and axioms. The main goal in proof complexity is to obtain lower bounds on the minimum proof size required to prove a given theorem in a specified proof system. In particular we generally wish to prove superpolynomial lower bounds on the minimum proof size required for a given family of formulas in a given proof system.

This is analogous to the field of computational complexity where the minimal running times of algorithms are analysed. Indeed one motivation for the study of proof complexity is its strong links to computational complexity. The systematic study of proof complexity was begun by Cook and Reckhow in [27] who formally defined a proof system to be a function which can efficiently verify whether or not a string of symbols is a proof of a given logical formula. Using this definition the class of languages which can be decided by a non-deterministic algorithm in polynomial-time (NP) can be naturally defined in terms of proof complexity. As a result the study of proof complexity can be regarded as a potential route to solving the famous P vs NP question [23].

Another major motivation for the study of proof complexity is its strong links with satisfiability (SAT) solving [69]. The SAT problem is the problem of determining whether or not a given propositional formula has a satisfying assignment. This problem is known to be NP-complete [25, 57], hence every decision problem in NP can be reduced to a SAT problem. As a result SAT solvers have important applications throughout numerous areas of computer science [60].

A SAT solver is essentially an implementation of some propositional proof system. As such whenever a solver is applied to some unsatisfiable formula the trace of its run corresponds to a proof of unsatisfiabilty in its underlying proof system. Hence a theoretical understanding of SAT solving can be gained through studying the proof complexity of such proof systems. Most modern

SAT solvers use conflict-driven clause learning (CDCL) as their underlying algorithm, it is well known that the trace of any such algorithm is essentially a resolution refutation [7, 74]. Hence through studying the proof complexity of propositional resolution we can gain a deeper theoretical understanding of state of the art SAT solvers.

Propositional resolution [20, 31, 80] is a very simple proof system, consisting of only a single rule. This rule works by eliminating contradictory literals (i.e. propositional variables and their negations) from a formula. Crucially, repeatedly applying this rule to an unsatisfiable formula will always result in the derivation of a contradiction.

Given its simplicity it is perhaps unsurprising that from a proof complexity perspective, propositional resolution is considered to be a rather weak proof system. As such many superpolynomial proof size lower bounds have been shown for propositional resolution (cf. [85]). This is in direct contrast to many stronger propositional proof systems, in particular Frege systems, for which the existence of superpolynomial lower bounds is a major open problem [21].

Proof complexity of modal logics Whilst the vast majority of work within proof complexity has been focused on propositional logic, more recently an increasing amount of work has been carried out on the proof complexity of more stronger logics, including many non-classical logics [15]. In particular there has been a large amount of work concerning the proof complexity of quantified boolean formulas (QBF) [10, 11], which has accompanied significant improvements in QBF solving [48, 58]. Whereas the non-classical logics whose proof complexity has recently begun to be studied include intuitionist logic [44, 46, 50], superintuitionistic logic [49], default logic [16] and modal logic [45, 46, 49]).

Modal logics have wide ranging applications throughout computer science. For example description logics, which are known to be syntactic variants of modal logics [81], are used in knowledge representation [30, 43]. Other areas where modal logics have been successfully applied include game theory [59], knowledge compilation [17, 35] and formal verification [24, 72].

As a result of these varied applications many propositional proof systems, including Frege [46, 49], sequent calculus [87], tableaux systems [38] and resolution systems [63, 64], have been extended to modal logics. Furthermore a number of these proof systems are the basis of modal theorem provers (e.g. for tableau [39, 40] and for resolution [47, 66]). Hence, as in the propositional case with SAT solvers, through understanding the proof complexity of these underlying proof systems we can gain a deeper theoretical understanding of their associated provers.

Another motivation for the study of the proof complexity of modal logics comes from computational complexity. Most commonly used modal logics are in PSPACE [55] (i.e. the class of languages which are decidable by an algorithm which uses polynomial memory/space). Hence studying the proof complexity of modal logics can be seen as an attempt to separate PSPACE from NP.

Finally, through the study of proof complexity of modal logics we can hopefully gain a deeper understanding of proof complexity in the wider sense. It is clear from the existing work on the

proof complexity of modal logics that the picture is very different than it is for propositional logic. For example there exist modal formulas which require exponentially sized modal Frege proofs [46], whereas no such formulas are known to exist for propositional logic.

In propositional proof complexity resolution is by far the most studied proof system. This is in direct contrast to the proof complexity analysis of modal proof systems which is, to our knowledge, limited to the study of modal Frege, extended Frege and substitution Frege systems [46, 49].

Modal resolution systems As we saw above, for propositional logic, resolution is a very simple proof system which consists of only a single inference rule. However constructing a resolution based proof system for even the weakest normal multimodal logic $\mathbf{K}_n$ is not straightforward. This is because two complementary literals can occur at different "modal contexts" (i.e. be nested within different numbers or types of modalities) within a single modal formula, and so fail to contradict one another. As a result many different resolution methods have been proposed for modal logics.

Generally speaking modal resolution methods have followed two different approaches. The first method involves translating modal formulas into some other language for which there are well-developed existing resolution systems. Most often this language is first order logic [32, 82], although translations into other languages such as propositional logic [51, 84] and QBF [73] have also been proposed.

The second approach is to devise a resolution system that works directly on the modal logic that is being considered, usually after it has been translated into some clausal form. We informally refer to such systems as direct resolution systems. Examples of direct resolution systems which do not require translation into any clausal form are [1] and [36]. Examples of direct clausal resolution systems include [3, 5, 34, 61–64].

As we are interested in the proof complexity of modal logics as opposed to that of first order logic, in this thesis we consider only direct modal resolution systems. In particular, our main focus throughout is on the proof systems of Nalon and Dixon, and Nalon, Hustadt and Dixon given in [63] and [64] respectively. These two proof systems are closely related to one another, operating on similar normal forms and having almost identical inference rules.

There are several motivations for choosing to focus primarily on the resolution systems of Nalon and Dixon, and Nalon, Hustadt and Dixon. Firstly these proof systems are the most recently proposed of the direct modal resolution systems. Secondly the clausal forms that each of these systems act on are both much simpler than those of any other direct clausal resolution system. Finally, whilst many direct resolution methods for modal logics have been proposed, very few of these proof systems have been implemented as automated theorem provers. The direct clausal resolution system of Nalon, Hustadt and Dixon is an exception to this trend as it has an associated prover [65, 66].

We also consider the proof complexity of the direct clausal resolution system $\mathbf{RK}_n$ of Enjalbert and Fariñas del Cerro given in [34]. This proof system was among the earliest proposed modal resolution systems and both its inference rules and clausal form are much more involved than either

of the two more recent resolution systems we consider.

1.1 Contributions

We will now highlight our key contributions and theorems.

Comparing the strength of modal resolution systems In proof complexity we can compare the strength of two proof systems using simulations. Informally we say that one proof system *simulates* another if we can transfer proofs from the latter system to the former without causing a super-polynomial blow up in proof size.

In Chapters 4 and 5 we compare the strength of various resolution systems for the multimodal logic $\mathbf{K}_n$. We begin by reviewing the two modal resolution systems of Nalon and Dixon [63], and Nalon, Hustadt and Dixon [64], which we refer to as $\mathbf{K}_n$-Res and $\mathbf{K}_{ml}$-Res respectively. These systems are both extensions of propositional resolution, with a number of additional rules which are used to resolve on modal pivots. Each of these proof systems are clausal and so can only be applied to a modal formula once it has been translated into an appropriate normal form.

The earlier of these two proof systems is $\mathbf{K}_n$-Res whose normal form is such that the "modal context" of each clause is encoded by the extension variables added in the translation. This allows clauses to have modal depth at most one and hence greatly limits the number of ways complementary literals can be resolved together. However, although the "modal context" of each clause is encoded in its extension variables it cannot be easily read off and so this does not prevent us from unnecessarily resolving together two clauses with different modal contexts.

The normal form for $\mathbf{K}_{ml}$-Res similarly works by encoding the "modal context" of clauses within extension variables, however each clause is also labelled by a natural number denoting its modal level "modal level" (i.e. its modal depth within the original formula). This allows the inference rules of $\mathbf{K}_{ml}$-Res to be defined so that clauses can only be resolved together if they are labelled by the same natural number, and hence occurred at the same modal depth within the original formula. Hence $\mathbf{K}_{ml}$-Res has a smaller search space than $\mathbf{K}_n$-Res.

We further define two new resolution systems for $\mathbf{K}_n$ called $\mathbf{K}_{mp}$-Res (Definition 4.3.5) and $\mathbf{K}_{mc}$-Res (Definition 4.4.7). Both of these systems are natural refinements of $\mathbf{K}_{ml}$-Res, differing only in their respective normal forms.

The normal form for $\mathbf{K}_{mp}$-Res prefixes each clause by a finite sequence of positive modal operators (i.e. box operators). As this sequence of modal operators contains no diamond operators it does not tell us the precise "modal context" of the clause, however the sequence of agents corresponding to these modalities does specify the clause's "modal position" within the original formula[1]. The inference rules of $\mathbf{K}_{mp}$-Res are then defined so that clauses with different "modal positions" cannot be resolved together. Similarly, the resolution system $\mathbf{K}_{mc}$-Res works on a normal form where each clause is annotated by its "modal context", which is expressed as a word over

[1] If we apply the resolution system to the monomodal logic $\mathbf{K}_1$ then this resolution system is identical to $\mathbf{K}_{ml}$-Res.

the agents and extension variables. The inference rules of $\mathbf{K}_{mc}$-Res rules are then defined so that clauses with different "modal contexts" cannot be resolved together. Both of these proof systems admit less "unnecessary" resolution inferences than $\mathbf{K}_{ml}$-Res, with $\mathbf{K}_{mc}$-Res admitting the least. However whilst $\mathbf{K}_{mc}$-Res has the smallest search space of all the **K**-Res systems (i.e. $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res, $\mathbf{K}_{mp}$-Res and $\mathbf{K}_{mc}$-Res), $\mathbf{K}_{mp}$-Res has the advantage of not requiring translation to a language with any annotations and so can be compared to other $\mathbf{K}_n$ proof systems more easily.

We show in Theorem 4.5.1 that each of the **K**-Res systems simulates every other **K**-Res system, and hence that they are all equivalent to one another in terms of their proof complexity. The proof of this amounts to showing that any inference that can be preformed in $\mathbf{K}_n$-Res, but not $\mathbf{K}_{mc}$-Res (i.e. any inference on two clauses with different "modal contexts") does not contribute to the proof and so can be removed. This result allows us to focus only on the most convenient **K**-Res system both when proving lower bounds for them, and when comparing the strength of these systems with other modal proof systems.

In Definition 4.6.2 we propose a further refinement of the **K**-Res systems. This refinement allows us to drop two of the five (seven in the case of $\mathbf{K}_n$-Res) inference rules from each of the proof systems by making a minor change to the each of the normal forms that these systems operate on. In Theorem 4.6.3 we show that this refinement results in an equivalent proof system. This refinement is of particular use when proving that certain propositional lower bound techniques can be lifted to the **K**-Res systems.

Finally we compare the efficiency of the **K**-Res systems with that of the resolution system $\mathbf{RK}_n$ of Enjalbert and Fariñas del Cerro [34]. In particular we show in Theorem 5.2.1 that the older clausal resolution system $\mathbf{RK}_n$ simulates each of the **K**-Res systems. Due to the equivalence of the **K**-Res systems we only have to prove that $\mathbf{RK}_n$ simulates some **K**-Res system. Hence we choose to prove that $\mathbf{RK}_n$ simulates $\mathbf{K}_{mp}$-Res, as this is the only **K**-Res systems whose normal form $\mathbf{RK}_n$ can be directly applied to.

Proof size lower bound techniques In proof complexity we aim to prove superpolynomial lower bounds on the size of proofs. As previously discussed, as well as offering insight into how proof systems work, proving such lower bounds can be seen as a route to separating complexity classes. Further, due to the correspondence between proof systems and automated theorem provers, proof size lower bounds can also be regarded as worst case running times for provers.

What is arguably even more important than proving superpolynomial lower bounds for the size of proofs is to devise general techniques from which such lower bounds can be obtained. Indeed many such techniques have been devised for propositional proof systems, in particular for propositional resolution [8, 85]. A natural question to ask when considering the proof complexity of modal proof systems is whether any of these propositional techniques can be lifted to the modal setting. As we shall see the answer to this question is dependant on the technique being considered.

In this thesis we consider three lower bound techniques for modal resolution. The first two techniques are extensions of successful lower bound techniques for propositional resolution. The

first of these techniques can be lifted to modal resolution, whereas the latter cannot be. We further develop a new lower bound technique which can only be applied to modal proof systems. We will now give details of each of these techniques.

1. The first lower bound technique that we consider is feasible interpolation [53, 75], which is a well established propositional lower bound technique. An interpolant of an implication formula $A \rightarrow B$ is a circuit (or formula) which is satisfied whenever A is satisfied and falsified whenever B is falsified. A proof system has feasible interpolation if given any true implication formula and any proof of this formula we can efficiently extract an interpolating circuit whose size is polynomial in the size of the proof. Hence if some formula has only large interpolating circuits then every proof of this formula must also be large. As a result we can use feasible interpolation to obtain proof size lower bounds from circuit lower bounds.

   Feasible interpolation can be used to show exponential lower bounds for both propositional resolution [53] and another stronger propositional proof system called cutting planes [75]. In Theorem 6.2.1 we show that, with some minor adjustments, feasible interpolation can be lifted fairly straightforwardly to the **K**-Res modal resolution systems.

2. The second propositional technique we consider is size-width [9], which is arguably the most successful lower bound technique for propositional resolution. This technique allows exponential proof size lower bounds for propositional resolution to be proved indirectly via linear lower bounds on another standard measurement of proof complexity, proof width. The width of a proof is the largest number of literals contained within any line of the proof.

   In this thesis we show that the size-width technique cannot be lifted to either the **K**-Res systems (Theorem 7.3.1) or $\mathbf{RK}_n$ (Theorem 7.4.1). The proof of both theorems essentially consists of showing that there exist families of formulas which require proofs with linear width, but also have proofs of polynomial size.

3. Finally we propose a modal game theoretic lower bound technique for *tree-like* **K**-Res modal resolution systems (that is, **K**-Res systems where inferred clauses cannot be reused). This technique is inspired by game theoretic lower bound techniques for propositional resolution [13, 14, 76] and QBF resolution [12].

   Our modal game is played by a Prover and a Delayer. The Delayer claims to know some model for an unsatisfiable modal formula and the Prover refutes this claim by repeatedly asking questions about the structure of this model until a contradiction is exposed. Delayer scores points every time Prover poses a question, however the amount of points he scores is proportionate to the amount of useful information Prover gains from the answer. As Delayer can never win the game, his goal is to score as many points as possible before it ends.

   We show in Theorem 8.3.1 that if Delayer can devise a strategy which ensures he always scores at least s points when playing on a given modal formula ϕ then every tree-like $\mathbf{K}_{mc}$-Res refutation of ϕ contains at least 2^s modal resolution steps (i.e. resolution inferences where a

modal pivot is resolved on). Hence proving a linear lower bound on s yields an exponential lower bound on the number of modal resolution inferences needed to refute ϕ (and so the size of such a refutation).

Our technique differs significantly from the game theoretic lower bound techniques for propositional and QBF resolution mentioned above. At each round of the non-modal games Prover asks for an assignment to a variable, whereas in our modal game Prover asks about the model's accessibility relation. As a result using our modal game we are able to prove "truly" modal lower bounds, i.e. lower bounds on the number of modal resolution steps required to refute a given family of formulas.

Exponential lower bounds for modal resolution We use our modal lower bound techniques to prove two new exponential proof size lower bounds. These lower bounds are for the full *dag-like* version of the **K**-Res modal resolution systems (where inferred clauses can be reused) and the tree-like variant of the **K**-Res modal resolution systems respectively. The first of these lower bounds is for an existing family of hard modal formulas, whereas the other is for a new family of modal formulas.

The existing family of formulas that we use to show an exponential lower bound for the **K**-Res systems are the modal clique-colour formulas of Hrubeš [46]. The propositional clique-colour formulas encode that no graph with a clique of size $k+1$ can also be k-colourable. Hrubeš' modal version of these formulas encodes the same statement and is obtained by augmenting the propositional version with some additional modal operators. However whilst the propositional clique-colour formulas can be used to obtain lower bounds for propositional resolution [53] and cutting planes [75], the modal clique-colour formulas give lower bounds for modal Frege and modal extended Frege systems [46, 49].

In Theorem 6.4.2 we use our modal feasible interpolation technique to show that the modal clique-colour formulas require exponential size **K**-Res proofs.

To our knowledge the modal clique-colour formulas are the only family of modal formulas in the literature that have successfully been used to obtain lower bounds for modal proof systems. However these formulas cannot be used to prove exponential lower bounds via our game theoretic technique (see Section 10.3). Further, whilst there exist many benchmark formulas on which the performance of modal theorem provers can be analysed (e.g. [6]) the hardness of these formulas is typically due to proofs being hard to find as opposed to large in size. As in proof complexity we are interested only in the minimal size proofs and not how hard they are to obtain these benchmark formulas are not generally suitable for proof complexity analysis.

Hence in order to obtain lower bounds using our game-theoretic technique in Definition 9.1.1 we define a new family of hard modal formulas. These formulas are essentially a modal encoding of the pigeonhole principle (i.e. if $m > n$ then given any assignment of m pigeons to n pigeonholes some pigeonhole contains at least two pigeons). The pigeonhole principle has a well known propositional encoding (see for example [22]) which has been shown to be hard for (tree-like and dag-like)

propositional resolution using a number of different propositional lower bound techniques (e.g. size-width [9] and game theoretic techniques [76]). Despite both encoding the same principle the propositional pigeonhole formulas and our modal pigeonhole formulas are not obviously equivalent. In particular in our modal pigeonhole formulas pigeons are encoded through the accessibility relation and pigeonholes are encoded as propositional variables, unlike in the propositional encoding where both pigeons and pigeonholes are represented by propositional variables.

We show in the proof of Theorem 9.2.1 that there exists a Delayer strategy which ensures Delayer scores at least $\log n! - 1$ points whenever our modal Prover Delayer game is played on our modal pigeonhole formulas. Consequently every tree-like **K**-Res refutation of our modal pigeonhole formulas must contain at least $n! - 1$ modal resolution steps. Notably as this lower bound ignores all propositional inferences (i.e. inferences where only propositional variables are resolved on), it is a truly modal lower bound as opposed to a lifted propositional one.

Modal proof systems beyond resolution There are a wealth of proof systems beyond resolution for both propositional logic and modal logics. The most studied of these systems in terms of its propositional proof complexity are Frege systems.

Propositional Frege systems are known to be strictly stronger systems than propositional resolution. By this we mean that propositional Frege not only simulates propositional resolution, but there also exists a *separation* between the two systems [22]. That is, there exist formulas which have polynomial sized Frege proofs yet do not have short resolution proofs.

Frege systems have been extended to many modal logics (including the modal logic $\mathbf{K}_n$), and indeed their proof complexity has also begun to be considered [46]. However, to our knowledge, there is no existing work comparing the strength of modal Frege systems with presumably weaker modal proof systems, such as the modal resolution systems considered throughout this thesis.

We show in Proposition 10.1.1 that modal Frege systems simulate the modal resolution system $\mathbf{K}_{mp}$-Res, from which it follows immediately that modal Frege systems simulate the family of **K**-Res systems and their tree-like variants (as any tree-like proof system is simulated by its dag-like variant). We further show that our modal pigeonhole principle has short modal Frege proofs (Theorem 10.2.2), hence separating $\mathbf{K}_n$-Frege from the family of tree-like **K**-Res systems. Whilst a separation between modal Frege systems and the tree-like **K**-Res resolution systems follows trivially from the separation between the analogous propositional systems, ours is the first modal separation between these systems. The proof of our modal separation is similar to the proof that the pigeonhole principle is easy for propositional Frege given by Buss in [22].

1.2 Organisation

The remainder of the thesis is organised as follows. In Chapters 2 and 3 we give necessary preliminaries on modal logic and proof complexity respectively. For a full introduction to modal logic see [19, 54], and for a full introduction to proof complexity see [26].

The main content of the thesis begins in Chapter 4 where we investigate the proof complexity of the family of **K**-Res resolution systems. We begin this chapter by reviewing the modal resolution systems $\mathbf{K}_n$-Res and $\mathbf{K}_{ml}$-Res of Nalon, Hustadt and Dixon [63, 64]. We then define a number of refinements of $\mathbf{K}_{ml}$-Res and compare the strength of these new systems with $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res and each other using simulations.

In Chapter 5 we review the modal resolution system $\mathbf{RK}_n$ of Enjalbert and Fariñas del Cerro [34]. We further compare the proof complexity of this system with that of the family of **K**-Res systems.

The next three chapters are concerned with lower bound proving techniques. In Chapter 6 we show that the propositional technique of feasible interpolation can be extended to the family of **K**-Res proof systems. We further show that this technique can be applied to Hrubeš' modal clique-colour formulas [46] to obtain a lower bound for the **K**-Res systems. In Chapter 7 we show that the size-width lower bound proving technique cannot be extended to either **K**-Res systems or $\mathbf{RK}_n$. In Chapter 8 we give a game theoretic lower bound proving technique for tree-like **K**-Res proof systems.

In Chapter 9 we introduce a new family of formulas called the modal pigeonhole formulas. We further prove that these formulas are hard for each of the tree-like **K**-Res proof systems using the game theoretic lower bound proving technique which we introduced in Chapter 8.

In Chapter 10 we compare the proof complexity of the family of **K**-Res systems with that of modal Frege systems. In particular in Section 10.1 we show that modal Frege systems simulate the family of **K**-Res systems and in Section 10.2 we show a separation between the $\mathbf{K}_n$-Frege and tree-like **K**-Res.

Finally we conclude with a discussion of our work in Chapter 11.

1.3 Publications containing work in this thesis

Some of the work contained in Chapters 4 and 5 appeared in the paper:

- Sarah Sigley. *Resolution Calculi for Modal Logic and their Relative Proof Complexity.* In *Proceedings of the ESSLLI 2017 Student Session*, pages 60–72. 2017.

for which I was the sole author.
Some of the work contained in Chapters 4, 8, 9 and 10 is included in the paper:

- Sarah Sigley, Olaf Beyersdorff. *Proof Complexity of Modal Resolution.* Submitted to *Journal of Automated Reasoning*.

I was the main author for this paper.

Chapter 2

Preliminaries I: Modal Logic

Modal logics are extensions of propositional logic, hence we begin this chapter by giving an introduction to propositional logic. We then give an overview of modal logics.

2.1 Propositional logic

Propositional logic is constructed from a set of propositional variables, $P = \{p_1, p_2, \dots\}$, a complete set of propositional connectives $\{\neg, \wedge, \vee\}$ and the constants 0 and 1.

The connective $\neg$ is a unary connective denoting negation. The formula $\neg p_1$ states that the negation of p_1 is true, which is equivalent to stating that p_1 is not true. The connectives $\vee$ and $\wedge$ are both binary connectives denoting disjunction and conjunction respectively. Hence the formula $p_1 \vee p_2$ is read "as p_1 is true or p_2 is true" and the formula $p_1 \wedge p_2$ is read as "p_1 is true and p_2 is true". The constants 1 and 0 correspond to true and false respectively.

Definition 2.1.1. The set of well formed formulas (denoted wff) over the set of propositional variables P and propositional connectives $\{\neg, \wedge, \vee\}$ is defined inductively as follows:

- the constants $0,\ 1 \in \mathit{wff}$,
- if $\phi = p$ for some $p \in P$ then $\phi \in \mathit{wff}$,
- if $\phi \in \mathit{wff}$ then $\neg\phi \in \mathit{wff}$,
- if $\phi_1,\ \phi_2 \in \mathit{wff}$ then $\phi_1 \vee \phi_2 \in \mathit{wff}$ and $\phi_1 \wedge \phi_2 \in \mathit{wff}$.

A *literal* is either a propositional variable, $p \in P$, or its negation, $\neg p$. We let $\mathcal{L}$ denote the set of all literals. A *clause* is a disjunction of literals. We let $\mathcal{CL}$ denote the set of all propositional clauses. We say a propositional formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses.

We adopt the convention of identifying the empty disjunction/clause with 0 and the empty conjunction/CNF with 1. As conjunctions and disjunctions are both commutative and idempotent we can treat clauses and CNFs as sets of literals and clauses respectively.

Definition 2.1.2. A *model* for a formula $\phi \in \mathit{wff}$ is an assignment $\alpha : var(\phi) \to \{0, 1\}$, where $var(\phi)$ denotes the set of all propositional variables in ϕ. We say that α *satisfies* ϕ if when every propositional variable p in ϕ is replaced by $\alpha(p)$ then ϕ evaluates to 1.

If there exists a model that satisfies a formula $\phi \in \mathit{wff}$ then we say that ϕ is *satisfiable*. If every model for a given formula ϕ satisfies said formula then we say that ϕ is a *tautology*. If no model satisfies ϕ then we say that it is *unsatisfiable*.

2.2 Modal logic

A *multimodal logic* over some finite set of agents $\mathcal{A} = \{a_1, \dots, a_n\}$ is an extension of propositional logic constructed from a set of propositional variables, $P = \{p_1, p_2, \dots\}$, a complete set of propositional connectives $\{\neg, \wedge, \vee\}$, the constants 0 and 1, and a set of unary modal operators $\{\Box_{a_i} \mid a_i \in \mathcal{A}\}$. The formula $\Box_{a_i}\phi$ is read as "agent a_i considers ϕ to be necessary".

We further define the binary connective $\to$ so that $\phi \to \psi \equiv \neg\phi \vee \psi$, and for each $i \in [n]$ (where $[n]$ denotes the set $\{1, \dots, n\}$) we define the modal operator $\Diamond_{a_i} \equiv \neg\Box_{a_i}\neg$. The formulas $\phi_1 \to \phi_2$ and $\Diamond_{a_i}\phi_1$ are read as "if ϕ_1 is true then ϕ_2 is true" and "agent a_i considers ϕ_1 to be possible", respectively.

Notation 2.2.1. Throughout this thesis we take $\circ_a$ to be either $\Box_a$ or $\Diamond_a$. For any set Σ we define Σ^* to be the set of all finite words over Σ and ε to be the empty word. Further we define $\Box_\mu = \Box_{a_1} \dots \Box_{a_m}$ for $\mu = a_1 \dots a_m \in \mathcal{A}^*$.

Definition 2.2.1. Let $\mathcal{A}$ be a finite set of agents. The set of *well formed multimodal formulas* (denoted wfmf) over the set of propositional variables P, the set of modal connectives $\{\Box_a \mid a \in \mathcal{A}\}$ and the set of propositional connectives $\{\neg, \wedge, \vee\}$ is defined inductively as follows:

- the constants 0, $1 \in \mathit{wfmf}$,
- if $\phi = p$ such that $p \in P$ then $\phi \in \mathit{wfmf}$,
- if $\phi \in \mathit{wfmf}$ then $\neg\phi \in \mathit{wfmf}$,
- if $\phi_1, \phi_2 \in \mathit{wfmf}$ then $\phi_1 \wedge \phi_2 \in \mathit{wfmf}$ and $\phi_1 \vee \phi_2 \in \mathit{wfmf}$,
- if $\phi \in \mathit{wfmf}$ then $\Box_a\phi \in \mathit{wfmf}$.

A *positive modal literal* is a formula of the form $\Box_a l$, where $a \in \mathcal{A}$ and $l \in \mathcal{L}$. Similarly, a *negative modal literal* is a formula of the form $\Diamond_a l$. A *modal literal* is either a positive or negative modal literal.

Definition 2.2.2. Let $\phi \in \mathit{wfmf}$. We define a *subformula* of ϕ inductively as follows:

- ϕ is a subformula of itself,
- If $\neg\phi_1$ is a subformula of ϕ then so is ϕ_1,
- If $\phi_1 \wedge \phi_2$ is a subformula of ϕ then so are ϕ_1 and ϕ_2,
- If $\phi_1 \vee \phi_2$ is a subformula of ϕ then so are ϕ_1 and ϕ_2,
- If $\Box_a\phi_1$ is a subformula of ϕ then so is ϕ_1.

Definition 2.2.3. Let $\phi \in \mathit{wfmf}$. We define the *modal depth* of ϕ inductively as follows:

- If $\phi \in \mathit{wff}$ then the modal depth of ϕ is 0.
- If $\phi = \neg\phi_1$ for some $\phi_1 \in \mathit{wfmf}$ with modal depth m then ϕ has modal depth m.
- If $\phi = \phi_1 \wedge \phi_2$ for some $\phi_1, \phi_2 \in \mathit{wfmf}$ with modal depth m_1 and m_2 respectively, then ϕ has modal depth $\max(m_1, m_2)$.
- If $\phi = \phi_1 \vee \phi_2$ for some $\phi_1, \phi_2 \in \mathit{wfmf}$ with modal depth m_1 and m_2 respectively, then ϕ has modal depth $\max(m_1, m_2)$.
- If $\phi = \Box_a\phi_1$ for some ϕ_1 with modal depth m then ϕ has modal depth $m + 1$.

We further define the *modal depth* of some subformula ψ of ϕ as follows:

- If $\phi = \psi$ then the modal depth of ψ in ϕ is 0.
- If the modal depth of $\neg\psi$ in ϕ is m then the modal depth of ψ in ϕ is m.
- If the modal depth of $\psi \wedge \phi_1$ in ϕ is m, where $\phi_1 \in \mathit{wfmf}$ then the modal depth of ψ in ϕ is m.
- If the modal depth of $\psi \vee \phi_1$ in ϕ is m, where $\phi_1 \in \mathit{wfmf}$ then the modal depth of ψ in ϕ is m.
- If the modal depth of $\Box_a\psi$ in ϕ is m, where $\phi_1 \in \mathit{wfmf}$ then the modal depth of ψ in ϕ is $m + 1$.

Definition 2.2.4. Let $\mathcal{A}$ be some set of agents of size n. The *normal multimodal logic* multimodal logic $\mathbf{K}_n$ is the smallest set that contains all propositional tautologies, all formulas of the form:

$$\mathrm{K}_{a_i} : \Box_{a_i}(\phi \to \psi) \to (\Box_{a_i}\phi \to \Box_{a_i}\psi),$$

and is closed under the inference rules:

$$\text{modus ponens (MP): } \frac{\phi \to \psi \qquad \phi}{\psi} \quad \text{and} \quad a_i\text{-necessitation: } \frac{\phi}{\Box_{a_i}\phi}$$

for all formulas ϕ, $\psi \in \mathit{wfmf}$ and all agents $a_i \in \mathcal{A}$.

The semantics of multimodal logics are given using Kripke models.

Definition 2.2.5. A *Kripke model* (henceforth a model) over a set of propositional variables P and a set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$ is a tuple:

$$M = (W, R_{a_1}, \ldots, R_{a_n}, V),$$

where W is a non-empty set of "worlds", each R_{a_i} is a binary relation over W, which we call the a_i-accessibility relation, and V is a set of valuation functions $\{V(w) \mid w \in W\}$ such that $V(w) : P \to \{0, 1\}$.

Definition 2.2.6. We say that a model:

$$M' = (W', R'_{a_1}, \ldots, R'_{a_n}, V'),$$

extends a model:

$$M = (W, R_{a_1}, \ldots, R_{a_n}, V),$$

if $W' \supseteq W$, $V' \supseteq V$ and $R'_{a_i} \supseteq R_{a_i}$ for all $i \in [n]$.

Definition 2.2.7. Let ϕ, ψ be formulas and $p \in P$. Given a model $M = (W, R_1, \ldots R_n, V)$ and a world $w \in W$ the *satisfiability* of a formula at w in M is defined inductively as follows:

- $(M, w) \models p \iff w \in V(p)$,
- $(M, w) \models \neg\phi \iff (M, w) \models \phi$ does not hold (written as $(M, w) \not\models \phi$)
- $(M, w) \models \phi \wedge \psi \iff (M, w) \models \phi$ and $(M, w) \models \psi$,
- $(M, w) \models \phi \vee \psi \iff (M, w) \models \phi$ or $(M, w) \models \psi$,
- $(M, w) \models \Box_a \phi \iff (M, w') \models \phi$ for all w' such that $(w, w') \in R_a$.

We say ϕ is *locally satisfiable* (or just satisfiable) if there exists some world $w_0 \in W$ such that $(M, w_0) \models \phi$. We say that ϕ is *globally satisfiable* if $(M, w) \models \phi$ for all $w \in W$. This is denoted $M \models \phi$. We say that ϕ is *valid*, denoted $\models \phi$, if for every model M we have $M \models \phi$.

Definition 2.2.8. We define a *pointed model* to be a pair $\langle M, w \rangle$ consisting of a model $M = (W, R_{a_1}, \ldots, R_{a_n}, V)$ together with some distinguished world $w \in W$.

We further say a formula $\phi \in \mathit{wfmf}$ is satisfied by a pointed model $\langle M, w \rangle$ if $(M, w) \models \phi$. Hence a modal formula is locally satisfiable if and only if there exists some pointed model which satisfies it.

In this thesis we are concerned only with proof systems which determine whether or not a given formula $\phi \in \mathit{wfmf}$ is locally satisfiable in $\mathbf{K}_n$, as opposed to globally satisfiable. This is in some sense the easier of the two problems as the local satisfiability problem for $\mathbf{K}_n$ is PSPACE-complete [42, 55] whereas the global satisfiabilty problem for $\mathbf{K}_n$ is EXPTIME-complete [86].

Chapter 3

Preliminaries II: Proof systems and proof complexity

In this chapter we give the formal definition of a proof system and give some examples of proof systems for propositional logics. We then give an introduction to proof complexity.

3.1 Proof systems

Definition 3.1.1 ([27])**.** A *proof system* for some language $L \subseteq \Sigma^*$ is a polynomial time partial function $P : \Sigma^* \to L$ where Σ^* denotes the set of all finite words over Σ, and a *P-proof* of some $\tau \in L$ is a finite word $\pi \in \Sigma^*$ such that $P(\pi) = \tau$.

Intuitively, given a proof π of a formula $\tau \in L$, an L proof system efficiently verifies that π is a correct proof of τ.

Definition 3.1.2. Let P be a proof system. We say that ψ is *P provable* from ϕ if there exists a P-proof of ψ from ϕ. We denote this by $\phi \vdash_P \psi$.

The above definition of a proof system is rather general. In this document we will only consider *line based* proof systems.

Definition 3.1.3. A *line based proof system* is a proof system defined by some finite set of inference rules and axioms. A *proof* in a line based proof system is a sequence of proof lines, say $\lambda_1, \ldots, \lambda_n$ such that each λ_i is either an axiom of P or can be inferred by applying some rule of P to some subset of $\{\lambda_1, \ldots, \lambda_{i-1}\}$.

A proof in a line based system P must be either *tree-like* or *dag-like*. We say a P-proof is *tree-like* if every line is used as a premise of exactly one inference of P, that is if the proof has a tree structure. Otherwise, if lines can be reused, the structure of the proof is a directed acyclic graph (dag), and so we say the proof is *dag-like*.

Throughout this thesis if we do not specify whether we are considering the tree-like or dag-like version of a proof system then we assume that the full dag-like version is being referred to.

Definition 3.1.4. Let P be a line-based proof system and let π be a P-proof. Further let R be some inference rule of P and let λ_1 and λ_2 be lines of π. Then λ_2 is a *child* (an *R child*) of λ_1 if it is inferred by applying an inference rule (R) to a set of lines containing λ_1.
We say that λ_2 is a *descendant* (an *R descendant*) of λ_1 if it is either:

(i) a child (an R child) of λ_1 or,

(ii) a child (an R child) of a descendant (an R descendant) of λ_1.

If λ_2 is a descendant (an R descendant) of λ_1 then λ_1 is an *ancestor* (an *R ancestor*) of λ_2.

Definition 3.1.5. We say a proof system P is *strongly complete* if for every $\phi, \psi \in \mathit{wfmf}$ such that $\phi \models \psi$ we have $\phi \vdash_P \psi$. Further we say a proof system is *complete* if for every ϕ such that $\phi \models 0$ we have $\phi \vdash_P 0$.

We say P is *strongly sound* if for every ϕ and ψ such that $\phi \vdash_P \psi$ we have $\phi \models \psi$. We say P is *sound* if for every formula ϕ such that $\vdash \phi$ we have $\models \phi$.

3.1.1 Examples of proof systems

Propositional resolution Resolution [20, 31, 80] is a simple proof system for propositional logic. It acts on formulas in CNF and consists of the single rule:

$$\text{RES:} \quad \frac{C_1 \vee l \quad C_2 \vee \neg l}{C_1 \vee C_2}$$

where C_1, C_2 are clauses and l is a literal. The intuition behind this rule is straightforward. No propositional model can simultaneously satisfy a literal and its negation, hence if we take the disjunct of any two clauses containing complementary literals we may "cut away" (resolve on) said complementary literals. Throughout we will refer to the variable resolved on as a *pivot* variable.

Resolution is a *refutational* proof system. This means that to prove that a formula is valid using resolution we prove that its negation is unsatisfiable. So to prove that some formula ϕ is valid we would first convert its negation into CNF and then repeatedly apply the resolution rule until we derive the empty clause which is logically equivalent to 0.

Frege A *Frege system* for propositional logic is a line based proof system P consisting of a finite set of inference rules and axioms of the form $\phi_1, \ldots, \phi_k \vdash_P \phi$ and $\vdash_P \phi$ respectively, where

$\phi_1, \ldots, \phi_k, \phi$ are propositional formulas. Further P must be sound and strongly complete. An example of a propositional Frege system is given in Figure 3.1.

One way to prove that a propositional formula ϕ is a tautology using a Frege system is to refute its negation. That is, to derive a formula of the form $\neg\phi \to 0$. Alternatively, we can prove that a propositional formula is tautological by deriving it using the axioms and rules of a Frege system.

Modus ponens: $\dfrac{\neg\phi_1 \qquad \phi_1 \vee \phi_2}{\phi_2}$

$A1$	$\phi_1 \to (\phi_2 \to \phi_1)$	$A6$	$(\phi_1 \wedge \phi_2) \to \phi_2$
$A2$	$(\neg\phi_1 \to \neg\phi_2) \to (\phi_2 \to \phi_1)$	$A7$	$\phi_1 \to (\phi_2 \to (\phi_1 \wedge \phi_2))$
$A3$	$\phi_2 \to (\phi_1 \vee \phi_2)$	$A8$	$(\phi_1 \to \phi_2) \to ((\phi_2 \to \phi_3) \to (\phi_1 \to \phi_3))$
$A4$	$\phi_1 \to (\phi_1 \vee \phi_2)$	$A9$	$(\phi_1 \to (\phi_2 \to \phi_3)) \to ((\phi_1 \to \phi_2) \to (\phi_1 \to \phi_3))$
$A5$	$(\phi_1 \wedge \phi_2) \to \phi_1$	$A10$	$(\phi_1 \to \phi_3) \to (\phi_2 \to \phi_3) \to (\phi_1 \vee \phi_2 \to \phi_3)$

Figure 3.1: A propositional Frege system

3.2 Proof complexity

Broadly speaking there are two main goals in proof complexity. The first is two measure the minimum complexity of proofs required to prove some tautology in some proof system. The second is to compare the efficiency of proof systems. The most common measurement of proof complexity is proof size (Definition 3.2.1), however there also exist a number of other measurements such as proof length (Definition 3.2.2) and proof width (Definition 7.1.1).

Definition 3.2.1. The *size* of a proof π is the number of symbols it contains, denoted $|\pi|$.

Definition 3.2.2. Let P be a line based proof system for some language L. The *length* of a P-proof π is the number lines it contains.

Note that no propositional clause in a resolution refutation can contain more than $2n$ literals, where n is the number of variables in the formula being refuted. Hence the size of such a refutation can be super-polynomial in n if and only if its length is also super-polynomial in n. Thus from a proof complexity perspective size and length are interchangeable for resolution.

Generally in proof complexity we are not interested in the size of proofs required for individual tautologies but rather how proofs of families of tautologies behave asymptotically.

Definition 3.2.3. We say that an infinite family of formulas $\Phi \subseteq L$ is a *super-polynomial lower bound* for an L-proof system P if there exists no constant k such that every $\phi_n \in \Phi$ has a P-proof π_n where $|\pi_n| \leq k|\phi_n|^k$.

Similarly, we say that Φ is an *exponential lower bound* for P if there exists some $k > 0$ such that $|\pi_n| > 2^{kn}$.

Definition 3.2.4. An L-proof system is *polynomially bounded* if there exists a constant k such that for every $\tau \in L$ there exists a P proof π such that $P(\pi) = \tau$ and $|\pi| \leq k|\tau|^k$.

The original motivation for the study of proof complexity was the following seminal result of Cook and Reckhow [27].

Theorem 3.2.1 ([27])**.** There exists a polynomially bounded propositional proof system if and only if NP = coNP, where coNP denotes the class of decision problems whose complements are in NP.

As NP $\neq$ coNP only if P $\neq$ NP (where P denotes the class of polynomial-time decision problems), it follows immediately from the above theorem that if there does not exist a polynomially bounded propositional proof system then P $\neq$ NP.

We can compare the strength of two proof systems for a given language L using polynomial simulations.

Definition 3.2.5 ([27])**.** Let P and Q be L-proof systems. We say that P *polynomially simulates* (p-simulates) Q if there exists a polynomial time computable function f such that for any Q proof π such that $Q(\pi) = \tau$ where $\tau \in L$ we have $P(f(\pi)) = \tau$. We denote that P p-simulates Q by $Q \leq_p P$.

We say that P and Q are *polynomially equivalent* (p-equivalent) if $P \leq_p Q$ and $Q \leq_p P$, denoted $P \equiv_p Q$.

We say there exists a *separation* between two proof systems if they are not p-equivalent. Typically this is proved by showing that there exists a formula which is a *super-polynomial lower bound* for one proof system but has polynomial-sized proofs for the other.

Chapter 4

Resolution with modal positions and modal contexts

Constructing a resolution-based proof system for even the basic multimodal logic $\mathbf{K}_n$ is not as straightforward as it is for propositional logic. This is because whether or not we can only resolve complementary literals with one another now depends on the "modal context" in which they occur. To see this consider the formulas:

$$\phi = \Box_{a_1}(l_1 \vee l_2 \vee l_3),\ \psi = \neg l_1 \vee l_2,\ \theta = \Box_{a_1} \neg l_2 \text{ and } \zeta = \Diamond_{a_1} \neg l_3.$$

In any sound and complete $\mathbf{K}_n$ resolution system the following three statements should be true:

1. The instance of l_1 in ϕ cannot be resolved with the instance of $\neg l_1$ in ψ.
2. The instance of l_2 in ϕ can be resolved with the instance of $\neg l_2$ in θ to obtain a resolvent of the form $\Box_{a_1}(l_1 \vee l_3)$.
3. The instance of l_3 in ϕ can be resolved with the instance of $\neg l_3$ in ζ to obtain a resolvent of the form $\Diamond_{a_1}(\neg l_3 \wedge (l_1 \vee l_2))$.

Statement 1 is true as the instance of l_1 in ϕ is nested within the scope of a $\Box_{a_1}$ operator whereas the instance of $\neg l_1$ in ψ is not within the scope of any modal operator.

Statement 2 holds as the instance of $\neg l_2$ in ϕ and the instance of $\neg l_2$ in θ are both nested within a single $\Box_{a_1}$. Hence it follows that if ϕ and θ are both satisfied at some world w in some model $M = (W, R_1, \ldots, R_n, V)$ then $l_1 \vee l_2 \vee l_3$ and $\neg l_2$ must both be satisfied at every world w_1 such that $(w, w_1) \in R_1$ and so $l_1 \vee l_3$ must also be satisfied at every w_1.

Finally, the instance of l_3 in ϕ appears within the scope of a $\Box_{a_1}$ operator and the instance of $\neg l_3$ in ζ appears within the scope of a $\Diamond_{a_1}$ operator. Hence if ϕ and ζ are both satisfied at some world w in some model $M = (W, R_1, \ldots, R_n, V)$ then $l_1 \vee l_2 \vee l_3$ must be satisfied at every world $w_1 \in W$ such that $(w, w_1) \in R_1$ and $\neg l_3$ must be satisfied at some world $w_2 \in W$ such that $(w, w_2) \in R_1$. And so it follows by classical resolution that $l_1 \vee l_2$ must also be satisfied at w_2, hence statement 3 holds.

As a result of this added complexity several different $\mathbf{K}_n$ resolution systems have been proposed. In this chapter we shall revisit two such clausal resolution systems. These systems, which we shall refer to as $\mathbf{K}_n$-Res and $\mathbf{K}_{ml}$-Res, are closely related to each other and were proposed by Nalon and Dixon [63], and Nalon, Hustadt and Dixon [64], respectively.

4.1 The proof system $\mathbf{K}_n$-Res

The resolution system $\mathbf{K}_n$-Res [63] determines whether a formula ϕ is satisfiable at some distinguished "start" world, $s_0 \in W$. However as the choice of s_0 is arbitrary determining the satisfiability of ϕ at s_0 is essentially equivalent to determining the satisfiability of ϕ.

Let $M = (W, R_1, \ldots, R_n, V)$ be a model and $w_1, w_2 \in W$. We say w_2 is *reachable* from w_1 if (w_1, w_2) is in the reflexive and transitive closure of $\bigcup_{i=1}^{n} R_i$. Note that every world is reachable from itself. We define the *master modality*, denoted $\Box^*$, such that $(M, w) \models \Box^*\phi$ if and only if $(M, w') \models \phi$ for all w' reachable from w.

The proof system $\mathbf{K}_n$-Res operates on formulas that have been translated into the following normal form.

Definition 4.1.1 ([63])**.** Let $l, l', l_j \in \mathcal{L}$ and let $\mathbf{S}$ be a nullary connective defined such that $(M, w) \models \mathbf{S}$ if and only if $w = s_0$. We refer to $\mathbf{S}$ as the *start connective*. A formula ϕ is in *Separated Normal Form* (SNF) if:

$$\phi = \bigwedge_{i=1}^{r} \Box^* C_i,$$

where each C_i is of one of the following types of clauses:

- Start clause: $\mathbf{S} \to \bigvee_{j=1}^{t} l_j$,
- Literal clause: $\bigvee_{j=1}^{t} l_j$,
- Positive modal clause: $l' \to \Box_a l$,
- Negative modal clause: $l' \to \Diamond_a l$.

Definition 4.1.2. A modal formula over the set of operators $\{\Box_a, \Diamond_a, \neg, \wedge, \vee\}$ is in *negation normal form* (NNF) if only propositional variables are allowed to be within the scope of $\neg$.

Definition 4.1.3 ([63])**.** Any $\phi \in$ *wfmf* in NNF can be translated into a set of SNF clauses by applying the function:

$$T(\phi) = \Box^*(\mathbf{S} \to x) \wedge \rho(\Box^*(x \to \phi)),$$

where x is a new variable and the function ρ is defined inductively as follows:

$$\rho(\Box^*(x \to \theta \wedge \psi)) = \rho(\Box^*(x \to \theta)) \wedge \rho(\Box^*(x \to \psi)),$$

$$\rho(\Box^*(x \to \circ_a\theta)) = \begin{cases} \Box^*(x \to \circ_a\theta), & \text{if } \theta \in \mathcal{L}, \\ \Box^*(x \to \circ_a x_1) \wedge \rho(\Box^*(x_1 \to \theta)), & \text{otherwise.} \end{cases}$$

$$\rho(\Box^*(x \to \theta \vee \psi)) = \begin{cases} \Box^*(\neg x \vee \theta \vee \psi), & \text{if } \theta, \psi \in \mathcal{CL}, \\ \rho(\Box^*(x \to \theta \vee x_1)) \wedge \rho(\Box^*(x_1 \to \psi)) & \text{otherwise,} \end{cases}$$

where θ and ψ are formulas and x_1 is a new propositional variable.

Note that $\rho(\Box^*(x \to \theta \vee x_1)) = \rho(\Box^*(x \to (x_1 \vee x_2))) \wedge \rho(\Box^*(x_2 \to \theta))$ and so the translation always terminates.

We refer to the variables introduced when translating a formula $\phi \in \mathit{wfmf}$ into a set of SNF clauses $\mathcal{C}$ as *extension variables* and define $\mathcal{X}_\mathcal{C}$ to be the set of all such variables. Further we define:

$$\mathcal{X}_{\mathcal{C}+} = \{x' \in \mathcal{X} \mid \Box^*(x \to \Box_a x') \in \mathcal{C}\},\ \mathcal{X}_{\mathcal{C}-} = \{x' \in \mathcal{X} \mid \Box^*(x \to \Diamond_a x') \in \mathcal{C}\}$$
$$\text{and } \mathcal{X}_{\mathcal{C}\pm} = \mathcal{X}_{\mathcal{C}+} \cup \mathcal{X}_{\mathcal{C}-}.$$

Note that $\mathcal{X}_\mathcal{C} \subseteq \mathcal{L}$.

Let $\mathcal{C}$ be a set of SNF clauses and let $C \in \mathcal{C}$. We say $x \in \mathcal{X}_\mathcal{C}$ *appears positively* in C if either C is a literal clause of the form $\Box^*(x \vee D)$ where $D \in \mathcal{CL}$ or C is a modal clause of the form $(x' \to \circ_a x)$ where $x' \in \mathcal{X}_\mathcal{C}$. We say x *appears negatively* in C if either C is a literal clause of the form $\Box^*(\neg x \vee D)$ or C is a modal clause of the form $(x' \to \circ_a \neg x)$ or $(x \to \circ_a y)$, where $y \in \mathcal{L}$.

Example 4.1.1. Consider the modal formula $\phi = (x \vee \Diamond_a \neg y) \wedge \Box_a y \wedge \neg x$. Then:

$$\begin{aligned} T(\phi) &= \Box^*(\mathbf{S} \to x_0) \wedge \rho(x_0 \to \phi) \\ &= \Box^*(\mathbf{S} \to x_0) \wedge \rho(\Box^*(x_0 \to (x \vee \Diamond_a \neg y))) \wedge \rho(\Box^*(x_0 \to \Box_a y)) \wedge \rho(\Box^*(x_0 \to \neg x)) \\ &= \Box^*(\mathbf{S} \to x_0) \wedge \rho(\Box^*(x_0 \to x_1 \vee x_2)) \wedge \rho(\Box^*(x_1 \to x)) \wedge \\ &\qquad \rho(\Box^*(x_2 \to \Diamond_a \neg y)) \wedge \Box^*(x_0 \to \Box_a y) \wedge \Box^*(\neg x_0 \vee \neg x) \\ &= \Box^*(\mathbf{S} \to x_0) \wedge \Box^*(\neg x_0 \vee x_1 \vee x_2) \wedge \\ &\qquad \Box^*(\neg x_1 \vee x) \wedge \Box^*(x_2 \to \Diamond_a \neg y) \wedge \Box^*(x_0 \to \Box_a y) \wedge \Box^*(\neg x_0 \vee \neg x). \end{aligned}$$

Further $\mathcal{X}_{T(\phi)} = \{x_0, x_1, x_2\}$ and $\mathcal{X}_{T(\phi)+} = \mathcal{X}_{T(\phi)-} = \emptyset$.

Not only is the function T able to translate every modal formula into SNF it is also satisfiability preserving.

Theorem 4.1.1 ([63])**.** A formula ϕ is satisfiable if and only if the formula $T(\phi)$ is satisfiable.

As every SNF clause is prefixed by $\Box^*$ it follows that every SNF clause occurs within the same modal context. Hence the inference rules of $\mathbf{K}_n$-Res are relatively straightforward.

Definition 4.1.4 ([63])**.** The inference rules of $\mathbf{K}_n$-Res are given in Figure 4.1.

$$\text{IRES1: } \frac{\Box^*(\mathbf{S} \to D \vee l) \quad \Box^*(E \vee \neg l)}{\Box^*(\mathbf{S} \to D \vee E)} \qquad \text{IRES2: } \frac{\Box^*(\mathbf{S} \to D \vee l) \quad \Box^*(\mathbf{S} \to E \vee \neg l)}{\Box^*(\mathbf{S} \to D \vee E)} \qquad \text{LRES: } \frac{\Box^*(D \vee l) \quad \Box^*(E \vee \neg l)}{\Box^*(D \vee E)}$$

$$\text{MRES: } \frac{\Box^*(l_1 \to \Box_a l) \quad \Box^*(l_2 \to \Diamond_a \neg l)}{\Box^*(\neg l_1 \vee \neg l_2)} \qquad \text{GEN2: } \frac{\Box^*(l_1 \to \Box_a l) \quad \Box^*(l_2 \to \Box_a \neg l) \quad \Box^*(l_3 \to \Diamond_a l')}{\Box^*(\neg l_1 \vee \neg l_2 \vee \neg l_3)}$$

$$\text{GEN1: } \frac{\begin{array}{c}\Box^*(l'_1 \to \Box_a l_1) \\ \vdots \\ \Box^*(l'_z \to \Box_a l_z) \\ \Box^*(l' \to \Diamond_a l)\end{array}}{\Box^*(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l) \quad \Box^*(\neg l'_1 \vee \ldots \neg l'_z \vee \neg l')} \qquad \text{GEN3: } \frac{\begin{array}{c}\Box^*(l'_1 \to \Box_a l_1) \\ \vdots \\ \Box^*(l'_z \to \Box_a l_z) \\ \Box^*(l' \to \Diamond_a l)\end{array}}{\Box^*(\neg l_1 \vee \cdots \vee \neg l_z) \quad \Box^*(\neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l')}$$

where $l, l', l_j \in \mathcal{L}$ and $D, E \in \mathcal{CL}$.

Figure 4.1: Rules for $\mathbf{K}_n$-Res

The rules of $\mathbf{K}_n$-Res can be split into two categories, modal rules (MRES, GEN1, GEN2 and GEN3) and propositional rules (LRES, IRES1 and IRES2). MRES is the only modal rule where the resolution takes place outside of the modal operator, the rest of the modal rules resolve on literals inside some modal operator.

The rules IRES1, IRES2 and LRES are essentially propositional resolution. The rule MRES is the modal analogue of propositional resolution. The rule GEN2 says that if we have some negative modal clause, say $\Box^*(l'_3 \to \Diamond_a l_2)$, then we can resolve two positive modal literals of the form $\Box_a l_1$ and $\Box_a \neg l_1$ with one another. The negative modal clause is required for soundness as $\Box^*(l'_1 \to \Box_a l_1)$ and $\Box^*(l'_2 \to \Box_a \neg l_1)$ can both be satisfied by a model M at a world $w \in W$ such that $(w, w') \notin R_a$ for all $w' \in W$.

The rules GEN1 and GEN3 resolve literals with modal literals. More specifically, GEN1 says that given some clause $\Box^*(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$ we can simultaneously resolve the $z + 1$ literals $\neg l_1, \ldots, \neg l_z$ and $\neg l$ with the modal literals $\Box_a l_1, \ldots, \Box_a l_z$ and $\Diamond_a l$. When resolving literals with modal literals in this way we are taking advantage of the fact that, by the definition of $\Box^*$, any world in any model which satisfies $\Box^*(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$ must also satisfy $\Box_a(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$. Every literal in $\Box^*(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$ must be resolved on simultaneously as otherwise the resolvent obtained may not be in SNF. For example the resolvent obtained by resolving $\neg l_1$ in $\Box_a(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$ with $\Box_a l_1$ in $\Box^*(l'_1 \to \Box_a l_1)$ would be $\Box^*(\neg l'_1 \vee \Box_a(\neg l_2 \vee \ldots \neg l_z \vee \neg l))$. The rule GEN3 is similar to GEN1, however the negative modal literal, $\Diamond_a \neg l$, is not resolved on. Instead, as in the case for GEN2, it is necessary only for soundness.

Like propositional resolution, the proof system $\mathbf{K}_n$-Res is a refutational system. However in this proof system a refutation ends when the clause $\Box^*(\mathbf{S} \to 0)$ is derived.

Definition 4.1.5. Let π be a $\mathbf{K}_n$-Res refutation of some set of SNF clauses $\mathcal{C}$ and let C be some clause in π. If $C \in \mathcal{C}$ then we say that C is an *initial clause*. If $C \notin \mathcal{C}$ then we say C is a *non-initial* clause.

Note that none of the rules of $\mathbf{K}_n$-Res can be used to infer any modal clause. Hence every such clause in an $\mathbf{K}_n$-Res refutation must be initial.

Example 4.1.2. Let ϕ be defined as in Example 4.1.1. Further let $\mathcal{C}$ be the set of SNF clauses obtained by applying T to ϕ. We can refute $\mathcal{C}$ using $\mathbf{K}_n$-Res as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Box^*(x_0 \to \Box_a y)}{\Box^*(x_2 \to \Diamond_a \neg y)}}{\Box^*(\neg x_0 \vee \neg x_2)}\text{MRES}}{\Box^*(\neg x_0 \vee x_1 \vee x_2)}\text{LRES}}{\Box^*(\neg x_0 \vee x_1)}\text{LRES}}{\Box^*(\neg x_1 \vee x)}}{\Box^*(\neg x_0 \vee x)}\text{LRES}}{\Box^*(\neg x_0 \vee \neg x)}}{\Box^*(\neg x_0)}\text{IRES1}}{\Box^*(\mathbf{S} \to x_0)}}{\Box^*(\mathbf{S} \to 0)}$$

4.2 The proof system $\mathbf{K}_{ml}$-Res

In [64] Nalon, Hustadt and Dixon introduced a layered resolution system for $\mathbf{K}_n$ which we shall call $\mathbf{K}_{ml}$-Res. This resolution system is similar to $\mathbf{K}_n$-Res, however it operates on a normal form where each clause is labelled by its *modal level* (Definition 4.2.2). Informally the modal level of a clause is the number of modal operators it was nested within in the original formula.

Definition 4.2.1 ([64])**.** A formula ϕ is in *separated normal form with modal levels* (SNF$_{ml}$) if:

$$\phi = \bigwedge_{i=1}^{r} C_i,$$

where each clause C_i is either a:

- Positive modal clause: $(m : l' \to \Box_a l)$,
- Negative modal clause: $(m : l' \to \Diamond_a l)$,
- Literal clause: $(m : \bigvee_{j=1}^{s} l_j)$,

where $l, l', l_j \in \mathcal{L}$ and $m \in \mathbb{N}$ representing the modal level of the clause.

Let $M = (W, R_1, \ldots, R_n, V)$ be a model and $w, w' \in W$. We say w' is of *distance* m from w if there exists a path of length m from w to w' through the union of all accessibility relations in M.

The satisfiability of some $\phi \in \mathit{wfmf}$ labelled by its modal level, $m \in \mathbb{N}$, is given as follows:

$$(M, w_0) \models (m : \phi) \iff (M, w) \models \phi \text{ for all } w \in W \text{ such that } w \text{ is of distance } m \text{ from } w_0.$$

We can formally define the modal level of a subformula ψ within some formula ϕ as follows.

Definition 4.2.2 ([64]). Let $\Sigma = \{0, 1, 2, 3\}$. Further let $\phi, \psi \in \mathit{wfmf}$, let $p \in P$, let $\lambda \in \Sigma^*$ and let $m \in \mathbb{N}$. We define the function $\tau : \mathit{wfmf} \times \Sigma^* \times \mathbb{N} \to \mathcal{P}(\mathit{wfmf} \times \Sigma^* \times \mathbb{N})$ inductively as follows:

- $\tau(p, \lambda, m) = \{(p, \lambda, m)\}$,
- $\tau(\neg\phi, \lambda, m) = \{(\neg\phi, \lambda, m)\} \cup \tau(\phi, \lambda 0, m)$,
- $\tau(\phi \wedge \psi, \lambda, m) = \{(\phi \wedge \psi, \lambda, m)\} \cup \tau(\phi, \lambda 1, m) \cup \tau(\psi, \lambda 2, m)$,
- $\tau(\Box_a\phi, \lambda, m) = \{(\Box_a\phi, \lambda, m)\} \cup \tau(\phi, \lambda 3, m + 1)$.

Applying τ to $(\phi, \varepsilon, 0)$, where ε denotes the empty word, gives an *annotated syntactic tree* for ϕ. Each vertex in the tree corresponds to a subformula of ϕ, its unique position in the tree and its modal level in ϕ.

If $(\psi, \lambda, m) \in \tau(\phi, \varepsilon, 0)$ then we say that the modal level of ψ at position λ in ϕ is $\mathrm{ml}(\psi, \lambda) = m$.

The following procedure for efficiently translating any NNF formula into SNF_{ml}, whilst preserving satisfiability, is given in [64].

Definition 4.2.3. To convert an NNF formula ϕ into SNF_{ml} we apply the translation function:

$$T_{ml}(\phi) = x \wedge \rho_{ml}(0 : x \to \phi),$$

where x is a new propositional variable and ρ_{ml} is defined as follows:

$$\rho_{ml}(m : x \to \theta \wedge \psi) = \rho_{ml}(m : x \to \theta) \wedge \rho_{ml}(m : x \to \psi),$$

$$\rho_{ml}(m : x \to \circ_a\theta) = \begin{cases} (m : x \to \circ_a\theta), & \text{if } \theta \in \mathcal{L}, \\ (m : x \to \circ_a x_1) \wedge \rho_{ml}(m + 1 : x_1 \to \theta), & \text{otherwise.} \end{cases}$$

$$\rho_{ml}(m : x \to \theta \vee \psi)) = \begin{cases} (m : \neg x \vee \theta \vee \psi), & \text{if } \theta, \psi \in \mathcal{CL}, \\ \rho_{ml}(m : x \to \theta \vee x_1) \wedge \rho_{ml}(m : x_1 \to \psi), & \text{otherwise,} \end{cases}$$

where θ, ψ are formulas, x_1 is a new propositional variable and $m \in \mathbb{N}$.

The termination of this function follows as in Definition 4.1.3.

Example 4.2.1. Let $\phi = (x \vee \Diamond_a(\neg y \wedge x)) \wedge \Box_a y \wedge \neg x$. Then:

$$T_{mp}(\phi) = (0 : x_0) \wedge (0 : \neg x_0 \vee x_1 \vee x_2) \wedge (0 : \neg x_1 \vee x) \wedge (0 : x_2 \to \Diamond_a x_3) \wedge$$
$$(1 : \neg x_3 \vee \neg y) \wedge (1 : \neg x_3 \vee x) \wedge (0 : x_0 \to \Box_a y) \wedge (0 : \neg x_0 \vee \neg x).$$

Theorem 4.2.1 ([64])**.** An NNF formula ϕ is satisfiable if and only if $T_{ml}(\phi)$ is satisfiable.

Definition 4.2.4 ([64])**.** The inference rules of $\mathbf{K}_{ml}$-Res are given in Figure 4.2.

$$\text{LRES: } \frac{(m : D \vee l) \quad (m : E \vee \neg l)}{(m : D \vee E)} \qquad \text{MRES: } \frac{(m : l_1 \to \Box_a l) \quad (m : l_2 \to \Diamond_a \neg l)}{(m : \neg l_1 \vee \neg l_2)}$$

$$\text{GEN1: } \frac{\begin{array}{c}(m : l'_1 \to \Box_a l_1) \\ \vdots \\ (m : l'_z \to \Box_a l_z) \\ (m : l' \to \Diamond_a l)\end{array}}{(m : \neg l'_1 \vee \ldots \neg l'_z \vee \neg l')} \qquad \text{GEN2: } \frac{\begin{array}{c}(m : l_1 \to \Box_a l) \\ (m : l_2 \to \Box_a \neg l) \\ (m : l_3 \to \Diamond_a l')\end{array}}{(m : \neg l_1 \vee \neg l_2 \vee \neg l_3)}$$

$$\text{GEN3: } \frac{\begin{array}{c}(m : l'_1 \to \Box_a l_1) \\ \vdots \\ (m : l'_z \to \Box_a l_z) \\ (m : l' \to \Diamond_a l)\end{array}}{(m : \neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l')}$$

where $l, l', l_j \in \mathcal{L}$, $m \in \mathbb{N}$ and $D, E \in \mathcal{CL}$.

Figure 4.2: Rules for $\mathbf{K}_{ml}$-Res

The rules of $\mathbf{K}_{ml}$-Res are almost identical to those of $\mathbf{K}_n$-Res, however now LRES, MRES and GEN2 may only be applied to clauses that are at the same modal level. Further GEN1 and GEN3 may only be applied to sets of clauses where each modal clause is the same modal level and the literal clause is at the modal level above. The rules IRES1 and IRES2 are no longer necessary as we are no longer determining satisfiability at a fixed start world.

4.3 Resolution with modal positions

We shall now present a new resolution system for $\mathbf{K}_n$ called $\mathbf{K}_{mp}$-Res. This proof system is a refinement of $\mathbf{K}_{ml}$-Res where complementary literals can be resolved together if and only if they have the same *modal position*. The modal position of a clause tells us not only how many modal

operators it was nested within in the original formula, but also which agents these modal operators correspond to.

We formally define the modal position of a subformula ψ of a formula ϕ by extending Definition 4.2.2 as follows.

Definition 4.3.1. Let $\Sigma = \{0, 1, 2, 3\}$. Further let $\phi, \psi \in \mathit{wfmf}$, let p be a propositional variable, let $\lambda \in \Sigma^*$, let $m \in \mathbb{N}$ and let $\mu \in \mathcal{A}^*$. We define $\tau : \mathit{wfmf} \times \Sigma^* \times \mathbb{N} \times \mathcal{A}^* \to \mathcal{P}(\mathit{wfmf} \times \Sigma^* \times \mathbb{N} \times \mathcal{A}^*)$ inductively as follows:

- $\tau(p, \lambda, m, \mu) = (p, \lambda, m, \mu)$,
- $\tau(\neg\phi, \lambda, m, \mu) = (\neg\phi, \lambda, m, \mu) \cup \tau(\phi, \lambda 0, m, \mu)$,
- $\tau(\phi \vee \psi, \lambda, m, \mu) = (\phi \vee \psi, \lambda, m, \mu) \cup \tau(\phi, \lambda 1, m, \mu) \cup \tau(\psi, \lambda 2, m, \mu)$,
- $\tau(\Box_a\phi, \lambda, m, \mu) = (\Box_a\phi, \lambda, m, \mu) \cup \tau(\phi, \lambda 3, m + 1, \mu a)$.

Applying τ to $(\phi, \varepsilon, 0, \varepsilon)$ gives an annotated syntactic tree for ϕ. Each vertex corresponds to a subformula of ϕ, its unique position in the tree, its modal level in ϕ and its modal position in ϕ.

Suppose ϕ and ψ are such that $(\psi, \lambda, m, \mu) \in \tau(\phi, \varepsilon, 0, \varepsilon)$. Then the *modal position* of ψ at position λ in ϕ is defined to be $\mathrm{mp}(\psi, \lambda, m) = \mu$.

To refute a formula using $\mathbf{K}_{mp}$-Res we must first translate it into a clausal form where each clauses modal position with respect to the original formula is explicitly given.

Definition 4.3.2. Let $l, l', l_j \in \mathcal{L}$. A formula ϕ is in *separated normal form with modal positions* (SNF$_{mp}$) if:

$$\phi = \bigwedge_{i=1}^{r} C_i,$$

where each C_i is either a:

- Positive modal clause: $\Box_\mu(l' \to \Box_a l)$,
- Negative modal clause: $\Box_\mu(l' \to \Diamond_a l)$,
- Literal clause: $\Box_\mu(\bigvee_{j=1}^{t} l_j)$.

Note that if $\mu = \varepsilon$ then $\Box_\mu A \equiv A$. Hence for example $\Box_\varepsilon(l \to \Diamond_a l') \equiv l \to \Diamond_a l'$ is a negative modal clause and we use the two forms interchangeably.

Definition 4.3.3. Let $M = (w, R_{a_1}, \ldots, R_{a_n}, V)$ be a Kripke model. We say a world $w_m \in W$ is $\mu = a_1, a_2, \ldots, a_m$-accessible from $w_0 \in W$ if there exists a path $(w_0, w_1), \ldots, (w_{m-1}, w_m)$ from w_0 to w_m such that $(w_{i-1}, w_i) \in R_{a_i}$ for all $i \in \{0, \ldots, m\}$.
Further w_m is ε-accessible from w_0 if and only if $w_m = w_0$.

Note that an SNF$_{mp}$ clause $\Box_\mu C$ is satisfied at some world w_0 in some model $M = (W, R_1, \ldots, R_n, V)$ only if C is satisfied at every world $w \in W$ which is μ-reachable from w_0.

Definition 4.3.4. To convert an NNF formula ϕ into SNF$_{mp}$ we apply the translation function:

$$T_{mp}(\phi) = x \wedge \rho_{mp}(x \to \phi),$$

where x is a new propositional variable and ρ_{mp} is defined as follows:

$$\rho_{mp}(\Box_\mu(x \to \theta \wedge \psi)) = \rho_{mp}(\Box_\mu(x \to \theta)) \wedge \rho_{mp}(\Box_\mu(x \to \psi)),$$

$$\rho_{mp}(\Box_\mu(x \to \circ_a\theta)) = \begin{cases} \Box_\mu(x \to \circ_a\theta), & \text{if } \theta \in \mathcal{L}, \\ \Box_\mu(x \to \circ_a x_1) \wedge \rho_{mp}(\Box_{\mu a}(x_1 \to \theta)), & \text{otherwise.} \end{cases}$$

$$\rho_{mp}(\Box_\mu(x \to \theta \vee \psi)) = \begin{cases} \Box_\mu(\neg x \vee \theta \vee \psi), & \text{if } \theta, \psi \in \mathcal{CL}, \\ \rho_{mp}(\Box_\mu(x \to \theta \vee x_1)) \wedge \rho_{mp}(\Box_\mu(x_1 \to \psi)), & \text{otherwise,} \end{cases}$$

where θ, ψ are formulas, x_1 is a new propositional variable and $\mu \in \mathcal{A}^*$.

The termination of this translation function follows as for the analogous functions given in Definitions 4.1.3 and 4.2.3.

Example 4.3.1. Let $\phi = (x \vee \Diamond_a(\neg y \wedge x)) \wedge \Box_a y \wedge \neg x$. Then:

$$T_{mp}(\phi) = \Box_\varepsilon(x_0) \wedge \Box_\varepsilon(\neg x_0 \vee x_1 \vee x_2) \wedge \Box_\varepsilon(\neg x_1 \vee x) \wedge \Box_\varepsilon(x_2 \to \Diamond_a x_3) \wedge \Box_a(\neg x_3 \vee \neg y) \wedge \Box_a(\neg x_3 \vee x) \wedge \Box_\varepsilon(x_0 \to \Box_a y) \wedge \Box_\varepsilon(\neg x_0 \vee \neg x).$$

Theorem 4.3.1. An NNF formula ϕ is satisfiable if and only if $T_{mp}(\phi) = x \wedge \rho_{mp}(x \to \phi)$ is satisfiable.

Proof. ($\Rightarrow$): Let $M = (W, R_1, \ldots, R_n, V)$ be a model and $w_0 \in W$ such that $(M, w_0) \models \phi$. Further let $M_1 = (W, R_1, \ldots, R_n, V_1)$ where:

$$V_1(w)(p) = V(w)(p) \text{ for all } w \in W \text{ and all variables } p \text{ in the domain of } V,$$

$$\text{and} \qquad V_1(w)(x) = \begin{cases} 1 & \text{if } w = w_0, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(M_1, w_0) \models x$ and $(M_1, w_0) \models \phi$, and so $(M_1, w_0) \models x \to \phi$.

We will prove by induction on the structure of ϕ that, given the existence of M_1, there exists a model $M_3 = (W, R_1, \ldots, R_n, V_3)$ such that $(M_3, w_0) \models \rho_{mp}(x \to \phi)$ and $V_3(w)(p) = V_1(w)(p)$ for all $w \in W$ and all propositional variables p in the domain of $V_1(w)$. It then follows that $(M_3, w_0) \models x$ and so $(M_3, w_0) \models x \wedge \rho_{mp}(x \to \phi)$.

There are two base cases, the first is when ϕ is a modal literal and the second is when ϕ is a propositional clause. In both cases it follows immediately from the definition of ρ_{mp} that $(M_1, w_0) \models \rho_{mp}(\Box_\mu(x \to \phi))$ as either $\rho_{mp}(\Box_\mu(x \to \phi)) = \Box_\mu(x \to \phi)$ if ϕ is a modal literal or

$\rho_{mp}(\Box_\mu(x \to \phi)) = \Box_\mu(\neg x \vee \phi)$ if ϕ is a propositional clause[1]. Hence we take $M_3 = M_1$.

The inductive cases are when $\phi = \psi \vee \theta$, $\phi = \circ_a \psi$ and $\phi = \psi \wedge \theta$.

Suppose $\phi = \psi \vee \theta$ where at least one of ψ and θ is not a propositional clause. Then:

$$\rho_{mp}(\Box_\mu(x \to \phi)) = \rho_{mp}(\Box_\mu(x \to \psi \vee x_1)) \wedge \rho_{mp}(\Box_\mu(x_1 \to \theta)).$$

Let $M_2 = (W, R_1, \ldots, R_n, V_2)$ where:

$$V_2(w)(p) = V_1(w)(p) \text{ for all variables } p \text{ in the domain of } V_1,$$

$$\text{and} \quad V_2(w)(x_1) = \begin{cases} 1 & \text{if } w \in W \text{ such that } (M_1, w) \not\models \psi \text{ and } w \text{ is } \mu\text{-accessible from } w_0, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(M_2, w_0) \models \Box_\mu(x_1 \to \theta)$ and $(M_2, w_0) \models \Box_\mu(x \to \psi \vee x_1)$. By the inductive hypothesis there exists a model $M_1' = (W, R_1, \ldots, R_n, V_1')$ such that $(M_1', w_0) \models \rho_{mp}(\Box_\mu(x \to \psi \vee x_1))$[2] and a model $M_2' = (W, R_1, \ldots, R_n, V_2')$ such that $(M_2', w_0) \models \rho(\Box_\mu(x_1 \to \theta))$. Further $V_1'(w)(p) = V_2'(w)(p) = V_1(w)(p)$ for all $w \in W$ and all p in the domain of V_1. For $i \in \{1, 2\}$ let X_i be the domain of $V_i'(w_0)$, then for all $w \in W$ let:

$$V_3(w)(p) = \begin{cases} V_1'(w)(p) & \text{if } p \in X_1 \\ V_2'(w)(p) & \text{if } p \in X_2 \setminus X_1. \end{cases}$$

Then $(M_3, w_0) \models \rho_{mp}(\Box_\mu(x \to \psi \vee x_1)) \wedge \rho_{mp}(\Box_\mu(x_1 \to \theta))$.

Suppose $\phi = \circ_a \psi$ where $\psi \notin \mathcal{L}$. Then:

$$\rho_{mp}(\Box_\mu(x \to \circ_a \psi)) = \Box_\mu(x \to \circ_a x_1) \wedge \rho_{mp}(\Box_{\mu a}(x_1 \to \psi)).$$

Let $M_2 = (W, R_1, \ldots, R_n, V_2)$ where:

$$V_2(w)(p) = V_1(w)(p) \text{ for all variables } p \text{ in the domain of } V_1,$$

$$\text{and} \quad V_2(w)(x_1) = \begin{cases} 1 & \text{if } w \in W \text{ such that } (M_1, w) \models \psi \text{ and } w \text{ is } \mu a\text{-accessible from } w_0, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(M_2, w_0) \models \Box_\mu(x \to \circ_a x_1)$ and $(M_2, w_0) \models \Box_{\mu a}(x_1 \to \psi)$. Hence by the inductive hypothesis there exists a model $M_3 = (W, R_1, \ldots, R_n, V_3)$ such that $(M_3, w_0) \models \rho_{mp}(\Box_{\mu a}(x_1 \to \psi))$ and $V_3(w)(p) = V_2(w)(p)$ for every $w \in W$ and every propositional variable p in the domain of V_2. Thus $(M_3, w_0) \models \Box_\mu(x \to \circ_a x_1) \wedge \rho_{mp}(\Box_{\mu a}(x_1 \to \psi))$.

[1] Note that here we have replaced $\Box_\varepsilon$ with $\Box_\mu$ where μ is some arbitrary finite word in $\mathcal{A}^*$ (possibly ε). This is necessary to apply the inductive hypothesis as for instance, if $\phi = \Box_a D$ where D is a propositional clause then $\rho_{mp}(\Box_\varepsilon(x \to \phi)) = \Box_\varepsilon(x \to \Box_a x_1) \wedge \rho_{mp}(\Box_a(x_1 \to D))$.

[2] Since $\rho_{mp}(\Box_\mu(x \to \psi \vee x_1)) = \rho_{mp}(\Box_\mu(x \to (x_1 \vee x_2))) \wedge \rho_{mp}(\Box_\mu(x_2 \to \psi))$.

Finally suppose $\phi = \psi \wedge \theta$. Then:

$$\rho_{mp}(\Box_\mu(x \to \psi \wedge \theta)) = \rho_{mp}(\Box_\mu(x \to \psi)) \wedge \rho_{mp}(\Box_\mu(x \to \theta)).$$

Let $M_2 = (W, R_1, \ldots, R_n, V_2)$ where:

$$V_2(w)(p) = V_1(w)(p) \text{ for all variables } p \text{ in the domain of } V_1,$$

$$\text{and} \quad V_2(w)(x_1) = \begin{cases} 1 & \text{if } w \in W \text{ such that } (M_1, w) \models \psi \wedge \theta \text{ and } w \text{ is } \mu\text{-accessible from } w_0, \\ 0 & \text{otherwise.} \end{cases}$$

Then $(M_2, w_0) \models \Box_\mu(x \to \psi)$ and $(M_2, w_0) \models \Box_\mu(x \to \theta)$. Hence by the inductive hypothesis there exist models $M_1' = (W, R_1, \ldots, R_n, V_1')$ and $M_2' = (W, R_1, \ldots, R_n, V_2')$ such that $(M_2', w_0) \models \rho_{mp}(\Box_\mu(x \to \psi))$ and $(M_2', w_0) \models \rho_{mp}(\Box_\mu(x \to \theta))$. Further $V_1'(w)(p) = V_2'(w)(p) = V_2(w)(p)$ for all $w \in W$ and all propositional variables p in the domain of V_2. For each $i \in \{1, 2\}$ let X_i denote the domain of V_i'. Further let $M_3 = (W, R_1, \ldots, R_n, V_3)$, where

$$V_3(w)(p) = \begin{cases} V_1'(w)(p) & \text{if } p \in X_1 \\ V_2'(w)(p) & \text{if } p \in X_2 \setminus X_1. \end{cases}$$

Then $(M_3, w_0) \models \rho_{mp}(\Box_\mu(x \to \psi)) \wedge \rho_{mp}(\Box_\mu(x \to \theta))$.

($\Leftarrow$): Clearly, if $(M, w_0) \models x \wedge \rho(x \to \phi)$ then $(M, w_0) \models \rho(x \to \phi)$ and $(M, w_0) \models x$. Hence to prove that $(M, w_0) \models \phi$ it suffices to prove that if $(M, w_0) \models \rho(x \to \phi)$ then $(M, w_0) \models x \to \phi$. We do this by induction on the structure of ϕ.

There are two base cases, the first is when ϕ is a modal literal and the second is when ϕ is a propositional clause. In both cases $(M, w_0) \models \Box_\mu(x \to \phi)$ by the definition of ρ_{mp}.

Suppose $\phi = \psi \wedge \theta$. Then $\rho_{mp}(\Box_\mu(x \to \phi)) = \rho_{mp}(\Box_\mu(x \to \psi)) \wedge \rho_{mp}(\Box_\mu(x \to \theta))$ and so $(M, w_0) \models \rho_{mp}(\Box_\mu(x \to \psi)) \wedge \rho_{mp}(\Box_\mu(x \to \theta))$. As ψ and θ are both subformulas of ϕ it follows by the inductive hypothesis that $(M, w_0) \models \Box_\mu(x \to \psi)$ and $(M, w_0) \models \Box_\mu(x \to \theta)$, hence $(M, w_0) \models \Box_\mu(x \to \psi \wedge \theta)$.

Suppose $\phi = \circ_a \psi$. Then $\rho_{mp}(\Box_\mu(x \to \circ_a \psi)) = \Box_\mu(x \to \circ_a x_1) \wedge \rho_{mp}(\Box_{\mu a}(x_1 \to \psi))$ and so $(M, w_0) \models \Box_\mu(x \to \circ_a x_1)$ and $(M, w_0) \models \rho_{mp}(\Box_{\mu a}(x_1 \to \psi))$. It follows by the inductive hypothesis that $(M, w_0) \models \Box_{\mu a}(x_1 \to \psi)$ and so $(M, w_0) \models \Box_\mu(x \to \circ_a \psi)$.

Finally suppose $\phi = \psi \vee \theta$. Then $\rho_{mp}(\Box_\mu(x \to \psi)) = \rho_{mp}(\Box_\mu x \to \psi \vee x_1) \wedge \rho_{mp}(\Box_\mu(x_1 \to \theta))$ and so $(M, w_0) \models \rho_{mp}(\Box_\mu x \to \psi \vee x_1)$ and $(M, w_0) \models \rho_{mp}(\Box_\mu(x_1 \to \theta))$. It follows by the inductive hypothesis that $(M, w_0) \models \Box_\mu(x \to \psi \vee x_1)$ and $(M, w_0) \models \Box_\mu(x_1 \to \theta)$ and so $(M, w_0) \models \Box_\mu(x \to \psi \vee \theta)$. □

Definition 4.3.5. The inference rules of $\mathbf{K}_{mp}$-Res are given in Figure 4.3.

Note that the inference rules for $\mathbf{K}_{mp}$-Res are the same as those for $\mathbf{K}_{ml}$-Res but are applied to a set of clauses in SNF$_{mp}$ as opposed to a set of clauses in SNF$_{ml}$.

$$\text{LRES: } \frac{\Box_\mu(D \vee l) \quad \Box_\mu(E \vee \neg l)}{\Box_\mu(D \vee E)} \qquad \text{MRES: } \frac{\Box_\mu(l_1 \to \Box_a l) \quad \Box_\mu(l_2 \to \Diamond_a \neg l)}{\Box_\mu(\neg l_1 \vee \neg l_2)}$$

$$\text{GEN1: } \frac{\begin{array}{c}\Box_\mu(l'_1 \to \Box_a l_1) \\ \vdots \\ \Box_\mu(l'_z \to \Box_a l_z) \\ \Box_\mu(l' \to \Diamond_a l)\end{array}}{\Box_\mu a(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l) \over \Box_\mu(\neg l'_1 \vee \ldots \neg l'_z \vee \neg l')} \qquad \text{GEN2: } \frac{\Box_\mu(l_1 \to \Box_a l) \quad \Box_\mu(l_2 \to \Box_a \neg l) \quad \Box_\mu(l_3 \to \Diamond_a l')}{\Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)}$$

$$\text{GEN3: } \frac{\begin{array}{c}\Box_\mu(l'_1 \to \Box_a l_1) \\ \vdots \\ \Box_\mu(l'_z \to \Box_a l_z) \\ \Box_\mu(l' \to \Diamond_a l)\end{array}}{\Box_\mu a(\neg l_1 \vee \cdots \vee \neg l_z) \over \Box_\mu(\neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l')}$$

where $l, l', l_j \in \mathcal{L}$, $\mu \in \mathcal{A}^*$ and $D, E \in \mathcal{CL}$.

Figure 4.3: Rules for $\mathbf{K}_{mp}$-Res

Theorem 4.3.2. The proof system $\mathbf{K}_{mp}$-Res is strongly sound.

Proof. We prove the theorem by showing that each of the rules of $\mathbf{K}_{mp}$-Res are sound.

To see that LRES is sound we let $C_1 = \Box_\mu(l \vee D_1)$ and $C_2 = \Box_\mu(\neg l \vee D_2)$. We further suppose we have some model $M = (W, R_1, \ldots, R_n, V)$ and some world $w \in W$ such that $(M, w) \models C_1$ and $(M, w) \models C_2$. Then as M satisfies C_1 at w, for every world w' which is μ-accesesible from w we have either $(M, w') \models l$ or $(M, w') \models D_1$. In the latter case it follows that $(M, w') \models D_1 \vee D_2$, whereas in the former case we have that $(M, w') \not\models \neg l$ and so as C_2 is also satisfied at w in M it follows that $(M, w') \models D_2$ and so $(M, w') \models D_1 \vee D_2$. Hence in both cases $(M, w) \models \Box_\mu(C_1 \vee C_2)$ and so LRES is sound.

To see that MRES is sound we let $C_1 = \Box_\mu(l_1 \to \Box_a l)$ and $C_2 = \Box_\mu(l_2 \to \neg\Box_a l)$. We further let M be a model and w be a world in M such that $(M, w) \models C_1$ and $(M, w) \models C_2$. It follows that for every world w' that is μ-accessible from w either $(M, w') \not\models l_1$ or $(M, w') \models \Box_a l$. In the former case $(M, w') \models \neg l_1 \vee \neg l_2$. In the latter case either there exists no world which is a-accessible from w' and so $(M, w') \not\models \neg\Box_a l$ or at every such world w'' we have $(M, w'') \models l$ and so once again $(M, w') \not\models \neg\Box_a l$. Hence as C_2 is satisfied at w in M it follows that $(M, w') \models \neg l_2$ and so $(M, w') \models \neg l_1 \vee \neg l_2$. Hence in every case $(M, w) \models \Box_\mu(\neg l_1 \vee \neg l_2)$ and so MRES is sound.

To see that GEN1 is sound we let $C_i = \Box_\mu(l'_i \to \Box_a l_i)$ for each $i \in [z]$, $C_{z+1} = \Box_\mu(l' \to \Diamond_a l)$ and $C_{z+2} = \Box_{\mu a}(\neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)$. We further let M be a model and w be a world in M such

that $(M, w) \models C_j$ for every $j \in [z+2]$. As every C_i is satisfied at w in M it follows that for each i and each world w' in M which is μ-accessible from w either $(M, w') \not\models l'_i$ or $(M, w') \models \Box_a l_i$. If $(M, w') \not\models l'_i$ for some i then $(M, w') \models \neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l'$. Otherwise if $(M, w') \models \Box_a l_i$ for every i then we note that as $(M, w) \models C_{z+1}$ either $(M, w') \not\models l'$ or $(M, w') \models \Diamond_a l$. In the former case $(M, w') \models \neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l'$. In the latter case there must exist some world w'' which is a-accessible from w'. Further $(M, w'') \models l$ and $(M, w'') \models l_i$ for all i. However this would mean that $(M, w'') \not\models \neg l_1 \vee \cdots \vee \neg l_z \vee \neg l$ contradicting our original assumption that $(M, w) \models C_{z+2}$. Hence in every case $(M, w) \models \Box_\mu(\neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l')$ and so GEN1 is sound.

To see that GEN2 is sound we let $C_1 = \Box_\mu(l_1 \rightarrow \Box_a l)$, $C_2 = \Box_\mu(l_2 \rightarrow \Box_a \neg l)$ and $C_3 = \Box_\mu(l_3 \rightarrow \Diamond_a l')$. We further let M be a model and w be a world in M such that $(M, w) \models C_j$ for every $j \in [3]$. As C_1 is satisfied at w in M it follows that for each world w' in M which is μ-accessible from w either $(M, w') \not\models l_1$ or $(M, w') \models \Box_a l$. In the former case $(M, w') \models \neg l_1 \vee \neg l_2 \vee \neg l_3$. In the latter case we note that as C_2 is also satisfied at w in M either $(M, w') \not\models l_2$ and so $(M, w') \models \neg l_1 \vee \neg l_2 \vee \neg l_3$, or $(M, w') \models \Box_a \neg l$. Note that the latter is only possible if M contains no worlds that are a-accessible from w', however as $(M, w) \models C_3$ this can only be the case if $(M, w') \models \neg l_3$, in which case we once again have that $(M, w') \models \neg l_1 \vee \neg l_2 \vee \neg l_3$. Hence in every case $(M, w) \models \Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)$ and so GEN2 is sound.

Finally to see that GEN3 is sound we let $C_i = \Box_\mu(l'_i \rightarrow \Box_a l_i)$ for each $i \in [z]$, $C_{z+1} = \Box_\mu(l' \rightarrow \Diamond_a l)$ and $C_{z+2} = \Box_{\mu a}(\neg l_1 \vee \cdots \vee \neg l_z)$. We further let M be a model and w be a world in M such that $(M, w) \models C_j$ for every $j \in [z+2]$. As every C_i is satisfied at w in M it follows that for each i and each world w' in M which is μ-accessible from w either $(M, w') \not\models l'_i$ or $(M, w') \models \Box_a l_i$. If $(M, w') \not\models l'_i$ for some i then $(M, w') \models \neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l'$. Otherwise if $(M, w') \models \Box_a l_i$ for every i then we note that as $(M, w) \models C_{z+1}$ either $(M, w') \not\models l'$ or $(M, w') \models \Diamond_a l$. In the former case $(M, w') \models \neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l'$. In the latter case there must exist some world w'' which is a-accessible from w'. Further $(M, w'') \models l$ and $(M, w'') \models l_i$ for all i. However this would mean that $(M, w'') \not\models \neg l_1 \vee \cdots \vee \neg l_z$ contradicting our original assumption that $(M, w) \models C_{z+2}$. Hence in every case $(M, w) \models \Box_\mu(\neg l'_1 \vee \cdots \vee \neg l'_z \vee \neg l')$ and so GEN3 is sound. □

To prove that the proof system $\mathbf{K}_{mp}$-Res is complete we follow a similar method to that of the proof of $\mathbf{K}_{ml}$-Res's completeness in [64]. We show that given a set of SNF$_{mp}$ clauses we can construct a unique graph whose vertices and edges correspond to worlds and accessibility relations respectively. We then show that this graph is empty if and only if $\mathcal{C}$ is unsatisfiable and that if $\mathcal{C}$ is unsatisfiable then the construction of the graph corresponds to some $\mathbf{K}_{mp}$-Res refutation of $\mathcal{C}$.

Let $\mathcal{C}$ be a set of SNF$_{mp}$ clauses and let $\mathcal{MP}$ be the set of modal positions of these clauses (i.e. $\mathcal{MP} = \{\mu \in \mathcal{A}^* \mid \Box_\mu C \in \mathcal{C}\}$). We construct a *behaviour graph*:

$$\mathcal{G} = \left\langle \bigcup_{\mu \in \mathcal{MP}} \mathcal{N}_\mu, \bigcup_{a \in \mathcal{A}} \mathcal{E}_a \right\rangle,$$

for $\mathcal{C}$ as follows. Let $\mathcal{N}$ be the set of all maximal consistent sets of the literals in $\mathcal{C}$. Further for each

$\mu \in \mathcal{MP}$ let $\mathcal{N}_\mu = \{(\mu, \eta) \mid \eta \in \mathcal{N}\}$ and for each $a \in \mathcal{A}$ let $\mathcal{E}_a = \bigcup_{\mu a \in \mathcal{MP}} \{(\eta_\mu, \eta'_{\mu a})\}$.

Let ϕ be a disjunction of literals. We say that ϕ is satisfied by vertex $\eta_\mu = (\mu, \eta)$, denoted $\eta_\mu \models \phi$, if and only if:

- ϕ is a literal and $\phi \in \eta$,
- $\phi = \psi \vee \theta$ and $\eta_\mu \models \psi$ or $\eta_\mu \models \theta$,
- $\phi = \psi \to \theta$ where either ψ is a literal such that $\neg\psi \in \eta$, or $\eta_\mu \models \theta$,
- $\phi = \Diamond_a x$ and there exists some $\eta'_{\mu a}$ such that $(\eta_\mu, \eta'_{\mu a}) \in \mathcal{E}_a$ and $\eta'_{\mu a} \models x$,
- $\phi = \Box_a x$ and $\eta'_{\mu a} \models x$ for every $\eta'_{\mu a}$ such that $(\eta_\mu, \eta'_{\mu a}) \in \mathcal{E}_a$.

From $\mathcal{G}$ we can construct a unique *reduced behaviour graph*, $\mathcal{G}'$, for $\mathcal{C}$ by deleting every vertex that fails to satisfy some clause in $\mathcal{C}$. First, for each literal clause $\Box_\mu D \in \mathcal{C}$ delete every $\eta_\mu \in \mathcal{N}_\mu$ such that $\eta_\mu \not\models D$. Then for each positive modal clause $\Box_\mu(l' \to \Box_a l) \in \mathcal{C}$ and each vertex $\eta_\mu \in \mathcal{N}_\mu$, if there exists an edge $(\eta_\mu, \eta'_{\mu a}) \in \mathcal{E}_a$, where $\eta'_{\mu a} \in \mathcal{N}_{\mu a}$, $\eta_\mu \models l'$ and $\eta'_{\mu a} \not\models l$ then we delete $(\eta_\mu, \eta'_{\mu a})$ from $\mathcal{E}_a$. Finally for each negative modal clause $\Box_\mu(l' \to \Diamond_a l)$ and each vertex $\eta_\mu \in \mathcal{N}_\mu$ if there does not exist an edge $(\eta_\mu, \eta'_{\mu a}) \in \mathcal{E}_a$, where $\eta'_{\mu a} \in \mathcal{N}_{\mu a}$, $\eta_\mu \models l'$ and $\eta'_{\mu a} \models l$ then we delete η_μ from $\mathcal{N}_\mu$.

To show that the reduced behaviour graph of a satisfiable set of clauses is non-empty we require the following well known property of the modal logic $\mathbf{K}_n$ (see for example [19]).

Theorem 4.3.3 (The finite tree model property)**.** Any satisfiable modal formula has a finite tree-like model.

Lemma 4.3.1. Let $\mathcal{C}$ be a set of SNF$_{mp}$ clauses and let:

$$\mathcal{G} = \left\langle \bigcup_{\mu \in \mathcal{MP}} \mathcal{N}_\mu, \bigcup_{a \in \mathcal{A}} \mathcal{E}_a \right\rangle,$$

be its reduced behaviour graph. Then $\mathcal{C}$ is satisfiable if and only if $\mathcal{N}_\varepsilon$ is non-empty.

Proof. ($\Rightarrow$): Suppose $\mathcal{C}$ is a satisfiable. Let $M = (W, R_{a_1}, \ldots, R_{a_n}, V)$ be a finite tree-like model and let $w_\varepsilon \in W$ such that $(M, w_\varepsilon) \models \mathcal{C}$. That such a model exists follows from the finite tree model property for $\mathbf{K}_n$. Further let:

$$N(M) = \{\eta_w \mid w \in W\} \text{ and } E(M)_a = \{(\eta_w, \eta_{w'}) \mid (w, w') \in R_a\} \text{ for each } a \in \mathcal{A},$$

where $\eta_w = (\mu, \{x \mid V(w)(x) = 1\} \cup \{\neg x \mid V(w)(x) = 0\})$ and $\mu \in \mathcal{A}^*$ such that w is μ-accessible from w_ε. In particular $\eta_{w_\varepsilon} \in \mathcal{N}_\varepsilon$. Note that as M is a tree-like model μ is unique, however if there exist two distinct worlds, w and w' that are both μ accessible from w_ε and have

exactly the same valuation functions then $\eta_w = \eta_{w'}$. If we further let:

$$G(M) = \left\langle N(M), \bigcup_{a \in \mathcal{A}} E(M)_a \right\rangle,$$

then $G(M)$ is a subgraph of the non-reduced behaviour graph for $\mathcal{C}$. Note that $G(M)$ is non-empty, in particular $\eta_{w_\varepsilon} \in N(M)$.

We will now show that $G(M)$ is also a subgraph of the reduced behaviour graph, $\mathcal{G}$ for $\mathcal{C}$. Consider some literal clause $\Box_\mu D \in \mathcal{C}$ with modal position μ. As $(M, w_\varepsilon) \models \Box_\mu D$ it follows that $(M, w) \models D$ for every $w \in W$ that is μ-accessible from w_ε and so $\eta_w \models D$. Suppose $C = \Box_\mu(l \to \Box_a l') \in \mathcal{C}$ (respectively $C = \Box_\mu(l \to \Diamond_a l') \in \mathcal{C}$). As $(M, w_\varepsilon) \models \mathcal{C}$ it follows that for all $w \in W$ that are μ-accessible from w_ε either $V(w)(l) = 0$ or, $V(w)(l) = 1$ and $V(w')(l') = 1$ for every (respectively some) $w' \in W$ such that $(w, w') \in R_a$. In the former case it follows that $\eta_w \models \neg l$ and so $\eta_w \models l \to \Box_a l'$ (respectively $\eta_w \models l \to \Diamond_a l'$). In the latter case note that $G(M)$ contains an a-edge from η_w to $\eta_{w'}$ if and only if w' is a-accessible from some $w'' \in W$ such that $\eta_{w'} = \eta_w$. Hence as $V(w'')(l) = V(w)(l) = 1$ for every such w'' it follows that $V(w')(l') = 0$ and so $\eta_{w'} \models l'$ and $\eta_w \models l \to \Box_a l'$ (respectively $\eta_w \models l \to \Diamond_a l'$).

As in the construction of $\mathcal{G}$ vertices and edges are only deleted if they fail to satisfy some clause it follows that $G(M)$ is a subgraph of $\mathcal{G}$ and so $\mathcal{N}_\varepsilon \supseteq \{\eta_{w_\varepsilon}\}$ is non-empty.

($\Leftarrow$): Suppose $\mathcal{N}_\varepsilon$ is non-empty. We construct a model $M = (W, R_1, \ldots, R_n, V)$ that satisfies $\mathcal{C}$ at some $w_\varepsilon \in W$ as follows. Let $f : \mathcal{N}_\mu \to \mathbb{N}$ be an injective function. Then for each $\mu \in \mathcal{MP}$ and each $\eta_\mu \in \mathcal{N}_\mu$ let $w_{\langle \mu, f(\eta)\rangle}$ be a world named by $\langle \mu, f(\eta)\rangle$. Further let $W_\mu = \bigcup_{\eta \in \mathcal{N}_\mu} w_{\langle \mu, f(\eta)\rangle}$ and let $W = \bigcup_{\mu \in \mathcal{MP}} W_\mu$. Let w_ε be an arbitrarily chosen world from the set W_ε. Then for each $a_j \in \mathcal{A}$ define the relation R_j so that $(w_{\langle \mu, f(\eta)\rangle}, w_{\langle \mu', f(\eta')\rangle}) \in R_j$ if and only if $(\eta_\mu, \eta'_{\mu'}) \in \mathcal{E}_{a_j}$. Finally we define V such that $V(w_{\langle \mu, f(\eta)\rangle})(p) = 1$ if and only if $\eta \models p$. It follows that $(M, w_\varepsilon) \models C$. $\Box$

Theorem 4.3.4. The proof system $\mathbf{K}_{mp}$-Res is complete.

Proof. Let $\mathcal{C}$ be an unsatisfiable set of SNF$_{mp}$ clauses and let:

$$\mathcal{G} = \left\langle \bigcup_{\mu \in \mathcal{MP}} \mathcal{N}_\mu, \bigcup_{a \in \mathcal{A}} E_a \right\rangle,$$

be its reduced behaviour graph. We will show that there exists a refutation of $\mathcal{C}$ corresponding to the deletion procedure used to construct $\mathcal{G}$ from the corresponding non-reduced behaviour graph $\mathcal{G}'$.

The first step used in the construction of $\mathcal{G}$ from $\mathcal{G}'$ is to delete all vertices in $\mathcal{N}_\mu$ that fail to satisfy some literal clause in $\mathcal{C}$ with modal position μ for each $\mu \in \mathcal{A}^*$. If $\mathcal{N}_\varepsilon$ becomes empty at this stage then the subset of $\mathcal{C}$ consisting of all literal clauses at modal position ε must be unsatisfiable. Hence by the completeness of propositional resolution there exists an LRES refutation of $\mathcal{C}$.

Suppose $\mathcal{N}_\varepsilon$ is not yet empty and there exist two clauses $\Box_\mu(l' \to \Box_a l) \in \mathcal{C}$ and $\Box_\mu(l'' \to \Diamond_a \neg l) \in \mathcal{C}$. Then by the construction of the graph any vertex $\eta \in \mathcal{N}_\mu$ that satisfies both l' and

$\Diamond_a \neg l$ is deleted, as is every vertex that satisfies both l'' and $\Box_a l$. This is equivalent to deleting every vertex that satisfies both l' and l''. Applying MRES to $\Box_\mu(l' \rightarrow \Box_a l)$ and $\Box_\mu(l'' \rightarrow \Diamond_a \neg l)$ we obtain $\Box_\mu(\neg l' \vee \neg l'')$, effectively mimicking the deletion of such vertices.

If $\mathcal{N}_\varepsilon$ is still non-empty then we proceed by considering the vertices that fail to satisfy some negative modal clause in $\mathcal{C}$. Let $C_a^{\eta_\mu} \subset \mathcal{C}$ be the set of all positive modal clauses, which contain a modal literal of the form $\Box_a l$ and are satisfied by η_μ and let $X_a^{\eta_\mu} = \{l_2 \mid \Box_\mu(l_1 \rightarrow \Box_a l_2) \in C_a^{\eta_\mu}\}$. Further let $<_1$ be a total ordering of $\mathcal{MP}$ where $\mu_1 <_1 \mu_2$ only if $|\mu_1| \leq |\mu_2|$. For each $\mu \in \mathcal{MP}$ in descending order we proceed as follows. Let $\mathcal{D}_{\mu a}$ be the set of all literal clauses with modal position μa that are either in $\mathcal{C}$ or whose derivation corresponds to some previous deletion. Suppose there exists a clause $\Box_\mu(l \rightarrow \Diamond_a l') \in \mathcal{C}$ and a vertex $\eta_\mu \in \mathcal{N}_\mu$ such that $\eta_\mu \models l$, but there does not exist a vertex which is a a-accessible from η_μ and satisfies l'. As l' alone cannot be contradictory and we have already dealt with the case when $X_a^{\eta_\mu} \cup \{l'\}$ is contradictory above, there are four cases:

1. The set $X_a^{\eta_\mu}$ is contradictory. Then there must exist two positive modal clauses of the form $\Box_\mu(l_1 \rightarrow \Box_a l_2)$ and $\Box_\mu(l_3 \rightarrow \Box_a \neg l_2)$, where $\eta_\mu \models l_1$ and $\eta_\mu \models l_3$. By applying GEN2 to these two clauses and $\Box_\mu(l \rightarrow \Diamond_a l')$ we obtain the clause $\Box_\mu(\neg l \vee \neg l_1 \vee \neg l_3)$, which corresponds to deleting η_μ.

2. Suppose $\mathcal{D}_{\mu a} \cup \{l'\}$ is contradictory. By the consequence completeness of propositional resolution[3] [56] there exists an LRES derivation of $\Box_{\mu a} \neg l'$ from $\mathcal{D}_{\mu a}$. By applying GEN1 to this formula and $\Box_\mu(l \rightarrow \Diamond_a l')$ we obtain $\Box_\mu \neg l$, which corresponds to deleting η_μ.

3. Suppose $X_a^{\eta_\mu} \cup \mathcal{D}_{\mu a}$ is contradictory. Then there exists some literal clause such that $\mathcal{D}_{\mu a} \models \Box_{\mu a}(\neg l_1 \vee \cdots \vee \neg l_n)$ where each $l_i \in \mathcal{X}_a^{\eta_\mu}$. Hence by the consequence completeness of propositional resolution there must exist an LRES derivation of some literal clause $C = \Box_{\mu a}(\neg l'_1 \vee \cdots \vee \neg l'_{n'})$, where each $l'_j \in \{l_1, \ldots, l_n\}$. Applying GEN3 to the clause C, the set of modal clauses $\{\Box_\mu(l''_i \rightarrow \Box_a l'_i) \in \mathcal{D}_{\mu a} \mid l'_i$ appears in $C\}$ and $\Box_\mu(l \rightarrow \Diamond_a l')$ corresponds to deleting η_μ.

4. Suppose $X_a^{\eta_\mu} \cup \mathcal{D}_{\mu a} \cup \{l'\}$ is contradictory. Then as above it follows from the consequence completeness of propositional resolution that there exists an LRES derivation from $\mathcal{D}_{\mu a}$ of some literal clause $C = \Box_\mu(\neg l_1 \vee \cdots \vee \neg l_n)$, where each $l_i \in \{l'\} \cup X_a^{\eta_\mu}$. Applying GEN1 to this clause, the set of modal clauses $\{\Box_\mu(l'_i \rightarrow \Box_a l_i) \in \mathcal{D}_{\mu a} \mid l_i$ appears in $C\}$ and $\Box_\mu(l \rightarrow \Diamond_a l')$ corresponds to deleting η_μ.

The above cases cover all possible deletions carried out when constructing a reduced behaviour graph for $\mathcal{C}$. As $\mathcal{C}$ is unsatisfiable it follows by Lemma 4.3.1 that every vertex in $\mathcal{N}_\varepsilon$ must have been deleted at some stage. As the deletion of each vertex $\eta_\varepsilon \in \mathcal{N}_\varepsilon$ corresponds to the derivation of a clause which it fails to satisfy it follows that the set $\mathcal{D}_\varepsilon$ is contradictory. Hence by the completeness of propositional resolution this set of clauses can be refuted using LRES. □

[3]That is, given any set of propositional clauses T, if T semantically implies some clause D then $T \vdash_{\mathbf{K}_{mc}-Res} C$, where C is a subclause of D.

4.4 Resolution with modal contexts

In this section we will define another new resolution system for $\mathbf{K}_n$, called $\mathbf{K}_{mc}$-Res which is a refinement $\mathbf{K}_{mp}$-Res. The rules of $\mathbf{K}_{mc}$-Res are essentially identical to those of $\mathbf{K}_{ml}$-Res and $\mathbf{K}_{mp}$-Res, however it acts on a normal form where each clause is labelled by its *modal context* as opposed to its modal level or position.

Informally, if we give each $\Diamond$ operator in some modal formula ϕ a unique label then the modal context of a subformula ψ of ϕ is the sequence of modal operators that it is nested within in ϕ. So for example if $\phi_1 = \Box_a \Diamond_a^1 x \wedge \Box_a \Diamond_a^2 y$ then x has modal context $\Box_a \Diamond_a^1$ and y has modal context $\Box_a \Diamond_a^2$. Whereas if $\phi_2 = \Diamond_a^1 (x \wedge y)$ then both x and y have modal context $\Diamond_a^1$. Intuitively two subformulas of ϕ have the same modal context if and only if in any model of ϕ these subformulas must be evaluated at exactly the same world or worlds. There exist models that satisfy ϕ_1 but do not contain any world w such that $V(w)(x) = 1$ and $V(w)(y) = 1$, however every model that satisfies ϕ_2 contains a world where $V(w)(x) = V(w)(y) = 1$. Hence in our new calculus $\mathbf{K}_{mc}$-Res we label each clause by its modal context to avoid unnecessary inferences.

Note that other formalisms in which the modal contexts of formulas are explicitly given have been previously defined such as Ohlbach's world paths [70, 71] and Schmidt's path logic [83].

To refute a formula using $\mathbf{K}_{mc}$-Res we must first translate it into a clausal form, where each clauses modal context with respect to the original formula is explicitly given. As the translation used introduces a new extension variable for every subformula of the form $\Diamond_a \phi$ where $\phi \notin \mathcal{L}$ we do not need to label the $\Diamond_a$ operators. The modal context of a clause can instead be specified by a finite word over the set of agents $\mathcal{A}$ and the set of pairs of the form (a, x) where a is an agent and x is an extension variable.

Definition 4.4.1. Let $l, l', l_j \in \mathcal{L}$. A formula ϕ is in *separated normal form with modal contexts* (SNF$_{mc}$) if:

$$\phi = \bigwedge_{i=1}^{r} C_i,$$

where each C_i is either a:

- Positive modal clause: $(e : l' \rightarrow \Box_a l)$,
- Literal clause: $(e : \bigvee_{j=1}^{t} l_j)$.
- Negative modal clause: $(e : l' \rightarrow \Diamond_a l)$,

Here e is a finite word over $\mathcal{E}_\mathcal{C}$ (Definition 4.4.3) denoting the modal context of the clause.

Definition 4.4.2. To convert an NNF formula ϕ into SNF$_{mc}$ we apply the translation:

$$T_{mc}(\phi) = x_\varepsilon \wedge \rho_{mc}(\varepsilon : x_\varepsilon \rightarrow \phi),$$

where x_ε is a new propositional variable and ρ_{mc} is defined as follows:

$$\rho_{mc}(e : x \to \theta \wedge \psi) = \rho_{mc}(e : x \to \theta) \wedge \rho_{mc}(e : x \to \psi),$$

$$\rho_{mc}(e : x \to \Box_a\theta) = \begin{cases} (e : x \to \Box_a\theta), & \text{if } \theta \in \mathcal{L}, \\ (e : x \to \Box_a x_1) \wedge \rho_{mc}(ea : x_1 \to \theta), & \text{otherwise.} \end{cases}$$

$$\rho_{mc}(e : x \to \Diamond_a\theta) = \begin{cases} (e : x \to \Diamond_a\theta), & \text{if } \theta \in \mathcal{L}, \\ (e : x \to \Diamond_a x_1) \wedge \rho_{mc}(e(a, x_1) : x_1 \to \theta), & \text{otherwise.} \end{cases}$$

$$\rho_{mc}(e : x \to \theta \vee \psi)) = \begin{cases} (e : \neg x \vee \theta \vee \psi), & \text{if } \theta, \psi \in \mathcal{CL}, \\ \rho_{mc}(e : x \to \theta \vee x_1) \wedge \rho_{mc}(e : x_1 \to \psi), & \text{otherwise,} \end{cases}$$

where θ, ψ are formulas, x_1 is a new propositional symbol and $e \in (\mathcal{A} \cup (\mathcal{A} \times \mathcal{X}_{\mathcal{C}-}))^*$.

The termination of this ρ_{mc} follows as for the analogous functions given in Definitions 4.1.3, 4.2.3 and 4.3.4.

Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses inferred by applying ρ_{mc} to some formula $\phi \in \mathit{wfmf}$. As in Section 4.1 we refer to the variables added during the translation as extension variables and define the sets $\mathcal{X}_{\mathcal{C}}$, $\mathcal{X}_{\mathcal{C}-}$, $\mathcal{X}_{\mathcal{C}+}$ and $\mathcal{X}_{\mathcal{C}\pm}$ in the obvious way.

Example 4.4.1. Let $\phi = (x \vee \Diamond_a(\neg y \wedge x)) \wedge \Box_a y \wedge \neg x$. Then:

$$T_{mc}(\phi) = (\varepsilon : x_0) \wedge (\varepsilon : \neg x_0 \vee x_1 \vee x_2) \wedge (\varepsilon : \neg x_1 \vee x) \wedge (\varepsilon : x_2 \to \Diamond_a x_3) \wedge$$
$$((a, x_3) : \neg x_3 \vee \neg y) \wedge ((a, x_3) : \neg x_3 \vee x) \wedge (\varepsilon : x_0 \to \Box_a y) \wedge (\varepsilon : \neg x_0 \vee \neg x).$$

Definition 4.4.3. For any set of SNF$_{mc}$ clauses $\mathcal{C}$ we define the set of *context markers* to be:

$$\mathcal{E}_{\mathcal{C}} = \mathcal{A} \cup (\mathcal{A} \times \mathcal{X}_{\mathcal{C}-}).$$

The set of all finite words over $\mathcal{E}_{\mathcal{C}}$ (denoted $\mathcal{E}_{\mathcal{C}}^*$) then consists of all modal contexts for $\mathcal{C}$.

Intuitively each label $(a, x) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-}$ refers to the unique $\Diamond_a$ operator such that $(e : x' \to \Diamond_a x) \in \mathcal{C}$. That this $\Diamond_a$ is unique follows from the definition of the translation T_{mc} as each extension variable in $\mathcal{X}_{\mathcal{C}-}$ appears exactly once as a modal literal. Each label $a \in \mathcal{A}$ refers to a $\Box_a$ operator.

Definition 4.4.4. Let $M = (W, R_{a_1}, \ldots, R_{a_n}, V)$ be a Kripke model, let $\phi \in \mathit{wfmf}$ and let $\mathcal{C} = T_{mc}(\phi)$. We say $w \in W$ is *ε-reachable* from $w_\varepsilon \in W$ if $w = w_\varepsilon$.
We say w is *ea-reachable* from w_ε if $(w', w) \in R_a$ for some $w' \in W$ such that w' is e-reachable from w. We say w is $e(a, x)$-reachable from w_ε where $x \in \mathcal{X}_{\mathcal{C}-}$ and $a \in \mathcal{A}$ if $(w', w) \in R_a$ for some $w' \in W$ such that w' is e-reachable from w and $V(w)(x) = 1$.

We define the satisfiability of a clause with modal context $e \in \mathcal{E}_{\mathcal{C}}^*$ as follows:

$$(M, w_\varepsilon) \models (e : C) \iff (M, w) \models C \text{ for all } w \in W \text{ such that } w \text{ is } e\text{-reachable from } w_\varepsilon.$$

Definition 4.4.5. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses and $x' \in \mathcal{X}_{\mathcal{C}}$. We say that x' is *propositionally reachable* from $x \in \mathcal{X}_{\mathcal{C}\pm}$ if there exists some subset of $\mathcal{C}$:

$$\mathcal{C}_{(x,x')} = \{(e : x_0 \to \circ_a x_1), (e : D_1 \vee \neg x_1 \vee x_2), \ldots, (e : D_{n-1} \vee \neg x_{n-1} \vee x_n)\}$$

where $x_1 = x$, $x_n = x'$, $x_i \in \mathcal{X}_{\mathcal{C}}$ for each $i \in \{0, \ldots, n-1\}$ and each $D_i \in \mathcal{CL}$. We say that such a set $\mathcal{C}_{(x,x')}$ *witnesses* that x' is propositionally reachable from x.

It follows immediately from the above definition and the definition of T_{mc} that every variable $x' \in \mathcal{X}_{\mathcal{C}}$ is propositionally reachable from some unique $x \in \mathcal{X}_{\mathcal{C}\pm}$. Further the set $\mathcal{C}_{(x,x')}$ witnessing this is unique.

Theorem 4.4.1. An NNF formula $\phi \in \mathit{wfmf}$ is satisfiable if and only if $T_{mc}(\phi)$ is satisfiable.

Proof. By Theorem 4.3.1 an NNF formula ϕ is satisfiable if and only if the set of SNF$_{mp}$ clauses $T_{mp}(\phi)$ is satisfiable. Hence we prove the theorem by showing that $T_{mp}(\phi)$ is satisfiable if and only if $T_{mc}(\phi)$ is satisfiable.

It follows immediately from the definitions of T_{mp} and T_{mc} that there is a one-to-one correspondence between the set of SNF$_{mp}$ clauses $\mathcal{C}_{mp} = T_{mp}(\phi)$ and the set of SNF$_{mc}$ clauses $\mathcal{C}_{mc} = T_{mc}(\phi)$. That is, $\Box_\mu C \in \mathcal{C}_{mp}$ if and only if $(e : C) \in \mathcal{C}_{mc}$ for some $e \in \mathcal{E}^*_{\mathcal{C}}$ such that $|e| = |\mu|$. In particular μ is the finite word over $\mathcal{A}$ obtained by replacing each pair $(a, x) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-}$ in e with a.

($\Rightarrow$): Suppose $\mathcal{C}_{mp}$ is satisfiable. Then there exists some model $M = (W, R_1, \ldots, R_n, V)$ and some $w_\varepsilon \in W$ such that $(M, w_\varepsilon) \models \Box_\mu C$ for every $\Box_\mu C \in \mathcal{C}_{mp}$. If $(M, w_\varepsilon) \models \Box_\mu C$, where $\mu = a_1 \ldots a_z$, then $(M, w_\varepsilon) \models (e : C)$ for all $e \in \mathcal{E}^*_{\mathcal{C}}$ such that $e = c_1 \ldots c_z$ where for each $i \in [z]$ either $c_i = a_i$ or $c_i = (a_i, x)$ for some $x \in \mathcal{X}_{\mathcal{C}-}$. Hence $(M, w_\varepsilon) \models \mathcal{C}_{mc}$.

($\Leftarrow$): Now suppose the set $\mathcal{C}_{mc}$ is satisfiable. Let $M = (W, R_1, \ldots, R_n, V)$ be a model such that $(M, w_\varepsilon) \models \mathcal{C}_{mc}$ for some $w_\varepsilon \in W$. Suppose $(e : C) \in \mathcal{C}_{mc}$ where $e \in \mathcal{E}^*_{\mathcal{C}}$ and let μ be the corresponding modal position. We will show that $(M, w_\varepsilon) \models \Box_\mu C$ via induction on $|e|$.

Suppose $|e| = 0$, then $e = \varepsilon$. By definition w_ε is the only world in W that is ε-reachable from itself. By assumption $(M, w_\varepsilon) \models (\varepsilon : C)$ hence it follows that $(M, w_\varepsilon) \models C$.

Now suppose $|e| > 0$. By definition $(M, w) \models C$ for every $w \in W$ which is e-reachable from w_ε. Hence all that remains is to show that $(M, w) \models C$ for every world that is μ-reachable from w_ε, but not e-reachable. By the definition of T_{mc} the clause C must contain exactly one negative extension literal $\neg x$. Further x must be propositionally reachable from some unique $x' \in \mathcal{X}_{\mathcal{C}\pm}$. We prove by induction on the size of the set $\mathcal{C}_{(x',x)}$ witnessing this that we can assume without loss of generality that $V(w)(x) = 0$ for all $w \in W$ such that w is not e-reachable from w_ε. From this it follows trivially that $(M, w) \models C$ at every $w \in W$ which is μ-accessible from w_ε.

If $|\mathcal{C}_{(x',x)}| = 1$ then $x \in \mathcal{X}_{\mathcal{C}\pm}$. Suppose $x \in \mathcal{X}_{\mathcal{C}-}$ (respectively $x \in \mathcal{X}_{\mathcal{C}+}$). It follows from the definition of T_{mc} that $e = e'(x, a)$ (respectively $e = e'a$) and $(e' : x' \to \Diamond_a x) \in \mathcal{C}_{mc}$ (respectively $(e' : x' \to \Box_a x) \in \mathcal{C}_{mc}$). Further by the definition of T_{mc} this is the only clause containing x

positively and every clause containing x negatively has modal context e. Hence we can assume without loss of generality that $V(w)(x) = 0$ for all $w \in W$ such that w is not e-reachable from w_ε. As C contains $\neg x$ it is satisfied at every such w.

Now suppose $|\mathcal{C}_{(x',x)}| > 1$. Then by definition $\mathcal{C}_{(x',x)}$ must contain some clause $(e : C') = (e : \neg x_1 \vee D \vee x)$ where $x_1 \in \mathcal{X}_\mathcal{C}$. Further $\mathcal{C}_{(x',x)} \setminus \{(e : C')\}$ witnesses that x_1 is propositionally reachable from x'. Hence by the inductive hypothesis we can assume without loss of generality that $V(w)(x_1) = 0$ at all worlds w that are not e-reachable from w_ε. Hence we can further assume without loss of generality that $V(w)(x) = 0$ at every such world as doing so will not change the truth valuation of C', which is the only clause in $\mathcal{C}_{mc}$ containing the positive literal x. □

In our new calculus we allow inferences to be made from sets of clauses with different modal contexts under certain conditions. To see why this is necessary consider the formula:

$$\phi = \Box_a(x \wedge y) \wedge \Diamond_a(\neg x \wedge z),$$

and the corresponding set of SNF$_{mc}$ clauses:

$$\mathcal{C} = \{(\varepsilon : x_\varepsilon),\ (\varepsilon : x_\varepsilon \to \Box_a x_1),\ (a : \neg x_1 \vee x),\ (a : \neg x_1 \vee y),\\ (\varepsilon : x_\varepsilon \to \Diamond_a x_2),\ ((a, x_2) : \neg x_2 \vee \neg x),\ ((a, x_2) : \neg x_2 \vee z)\}.$$

Clearly ϕ is unsatisfiable, however we cannot refute $\mathcal{C}$ using similar rules to those of $\mathbf{K}_n$-Res and $\mathbf{K}_{ml}$-Res if we do not allow inferences on clauses with different modal contexts. Hence we have the following definition.

Definition 4.4.6. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses. We define the *unification function* $\sigma : \mathcal{E}^*_\mathcal{C} \times \cdots \times \mathcal{E}^*_\mathcal{C} \to \mathcal{E}^*_\mathcal{C}$ so that $\sigma(\varepsilon, \ldots, \varepsilon) = \varepsilon$, further for every $c_1, \ldots, c_n \in \mathcal{E}_\mathcal{C}$:

$$\sigma(c_1, \ldots, c_n) = \begin{cases} c_j & \text{if some } c_j = (a, x) \in \mathcal{A} \times \mathcal{X}_\mathcal{C} \text{ and } c_k \in \{a, (a, x)\} \text{ for all } k \neq j, \\ a & \text{if } c_1 = \cdots = c_n = a \in \mathcal{A}, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

and for every $e_1, \ldots e_n \in \mathcal{E}^*_\mathcal{C}$:

$$\sigma(e_1, \ldots, e_n) = \begin{cases} \sigma(c_{1,1}, \ldots, c_{1,n}) \ldots \sigma(c_{m,1}, \ldots, c_{m,n}) & \text{if } |e_1| = \cdots = |e_n| = m > 0, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where $c_{i,j}$ denotes the ith letter in the word e_j. Note that if $\sigma(c_{i,1}, \ldots, c_{i,n})$ is undefined for any $i \in [m]$ then so is $\sigma(e_1, \ldots, e_n)$.

We say that the modal contexts $e_1, \ldots, e_n \in \mathcal{E}^*_\mathcal{C}$ are *unifiable* if $\sigma(e_1, \ldots, e_n)$ is defined. Otherwise we say that $e_1, \ldots, e_n$ are *non-unifiable*.

Example 4.4.2. Consider the modal contexts $e_1 = a_1a_2$, $e_2 = (a_1, x_1)a_2$ and $e_3 = (a_1, x_2)(a_2, x_3)$.

The modal context e_1 is unifiable with e_2 as:

$$\sigma(e_1, e_2) = \sigma(a_1, (a_1, x_1))\sigma(a_2, a_2) = (a_1, x_1)a_2.$$

Similarly, e_1 is unifiable with e_3 as:

$$\sigma(e_1, e_3) = \sigma(a_1, (a_1, x_2))\sigma(a_2, (a_2, x_3)) = (a_1, x_2)(a_2, x_3),$$

and so is defined. However the modal contexts e_2 and e_3 are not unifiable as $\sigma((a_1, x_1), (a_1, x_2))$ is undefined.

Intuitively allowing resolution on sets of clauses with unifiable modal contexts can be thought of as allowing $\Box_a$ to be resolved with $\Diamond_a$ to infer $\Diamond_a$, which is of course sound.

Definition 4.4.7. The inference rules of $\mathbf{K}_{mc}$-Res are given in Figure 4.4.

$$\text{LRES: } \frac{(e_1 : D \vee l) \quad (e_2 : E \vee \neg l)}{(\sigma(e_1, e_2) : D \vee E)} \qquad \text{MRES: } \frac{(e_1 : l_1 \to \Box_a l) \quad (e_2 : l_2 \to \Diamond_a \neg l)}{(\sigma(e_1, e_2) : \neg l_1 \vee \neg l_2)}$$

$$\text{GEN2: } \frac{(e_1 : l_1 \to \Box_a l) \quad (e_2 : l_2 \to \Box_a \neg l) \quad (e_3 : l_3 \to \Diamond_a l')}{(\sigma(e_1, e_2, e_3) : \neg l_1 \vee \neg l_2)}$$

$$\text{GEN1: } \frac{\begin{array}{c}(e_1 : l_1' \to \Box_a l_1) \\ \vdots \\ (e_z : l_z' \to \Box_a l_z) \\ (e_{z+1} : l' \to \Diamond_a l)\end{array} \quad (e_{z+2} y : \neg l_1 \vee \cdots \vee \neg l_z \vee \neg l)}{(\sigma(e_1, \ldots, e_{z+2}) : \neg l_1' \vee \ldots \neg l_z' \vee \neg l')} \qquad \text{GEN3: } \frac{\begin{array}{c}(e_1 : l_1' \to \Box_a l_1) \\ \vdots \\ (e_z : l_z' \to \Box_a l_z) \\ (e_{z+1} : l' \to \Diamond_a l)\end{array} \quad (e_{z+2} y : \neg l_1 \vee \cdots \vee \neg l_z)}{(\sigma(e_1, \ldots, e_{z+2}) : \neg l_1' \vee \cdots \vee \neg l_z' \vee \neg l')}$$

where $l, l', l_j \in \mathcal{L}$, $e_j \in \mathcal{E}_\mathcal{C}^*$, $y \in \mathcal{E}_\mathcal{C}$ and $D, E \in \mathcal{CL}$. We may only use the above rules to infer a clause $(\sigma(e_1, \ldots, e_n) : C)$ if $\sigma(e_1, \ldots, e_n)$ is defined.

Figure 4.4: Rules for $\mathbf{K}_{mc}$-Res

Remark 4.4.1. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses and let C be some clause which is $\mathbf{K}_{mc}$-Res derivable from $\mathcal{C}$. If C has modal context $e \in \mathcal{E}_\mathcal{C}^*$ then it must be inferred by applying some rule of $\mathbf{K}_{mc}$-Res to a set of clauses whose modal contexts are either unifiable with e or unifiable with ec for some $c \in \mathcal{E}_\mathcal{C}$. It follows by induction that C is $\mathbf{K}_{mc}$-Res derivable from the subset of $\mathcal{C}$ consisting of every clause whose modal context is of the form $e_1 e_2$, where $e_1, e_2 \in \mathcal{E}_\mathcal{C}^*$ and e_1 is unifiable with e.

Theorem 4.4.2. $\mathbf{K}_{mc}$-Res is complete and strongly sound.

Proof. Any proof system that p-simulates a complete proof system is complete and any proof system that is p-simulated by a strongly sound proof system is strongly sound. By Theorems 4.3.2 and 4.3.4 the proof system $\mathbf{K}_{mp}$-Res is strongly sound and complete, hence the theorem follows immediately from the fact that $\mathbf{K}_{mc}$-Res both p-simulates and is p-simulated by $\mathbf{K}_{mp}$-Res (Theorem 4.5.1). □

Note that we could have proven this theorem directly by following a very similar method to the one used in the proofs of Theorems 4.3.2 and 4.3.4.

4.5 Comparing $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res, $\mathbf{K}_{mp}$-Res and $\mathbf{K}_{mc}$-Res

In this section we prove that the modal resolution systems $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res, $\mathbf{K}_{mp}$-Res and $\mathbf{K}_{mc}$-Res all p- equivalent. As we will see, that $\mathbf{K}_{mc}$-Res $\leq_p$ $\mathbf{K}_{mp}$-Res $\leq_p$ $\mathbf{K}_{ml}$-Res $\leq_p$ $\mathbf{K}_n$-Res follows fairly straightforwardly from the respective definitions of the proof systems. Whereas our proof that $\mathbf{K}_n$-Res $\leq_p$ $\mathbf{K}_{ml}$-Res $\leq_p$ $\mathbf{K}_{mp}$-Res $\leq_p$ $\mathbf{K}_{mc}$-Res is more involved and essentially consists of showing that given any unsatisfiable set of SNF clauses, $\mathcal{C}$ and any $\mathbf{K}_n$-Res refutation π of $\mathcal{C}$, the following statement is true:

"The sequence of clauses obtained from π by deleting every clause inferred from a set of clauses whose modal contexts would prevent the analogous rule of $\mathbf{K}_{mc}$-Res from being applied to the analogous set of SNF$_{mc}$ clauses, along with every descendant of such a clause, is also a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$."

So for example if π contains a clause C which is inferred by applying LRES to two literal clauses with modal contexts e_1 and e_2 respectively and $\sigma(e_1, e_2)$ is undefined then π' would not contain C or any descendant of C.

4.5.1 Modal contexts for clauses in SNF

To prove that $\mathbf{K}_{mc}$-Res $\leq_p$ $\mathbf{K}_{mp}$-Res $\leq_p$ $\mathbf{K}_{ml}$-Res $\leq_p$ $\mathbf{K}_n$-Res we must be able to "read off" the modal context of a given clause in SNF. In Section 4.4 we saw that the extension variables introduced when translating a modal formula into SNF$_{mc}$ encode the modal context of each clause. This is also true of the extension variables introduced when translating a modal formula into SNF. Hence in this subsection we give a series of definitions which enable us determine the modal context of an SNF clause simply by looking at the extension variables it contains.

In Section 4.4 we defined what it meant for an extension variable $x' \in \mathcal{X}_\mathcal{C}$ to be propositionally reachable from some $x \in \mathcal{X}_{\mathcal{C}\pm}$ for a given set of SNF$_{mc}$ clauses $\mathcal{C}$. Similarly, for any set of SNF clauses $\mathcal{C}$ we say that $x' \in \mathcal{X}_\mathcal{C}$ is propositionally reachable from some $x \in \mathcal{X}_{\mathcal{C}\pm}$ if there exists some subset of $\mathcal{C}$ of the form:

$$\mathcal{C}_{(x,x')} = \{\Box^*(x_0 \rightarrow \circ_a x_1), \Box^*(D_1 \vee \neg x_1 \vee x_2), \ldots, \Box^*(D_n \vee \neg x_{n-1} \vee x_n)\},$$

where $x_1 = x$ and $x_n = x'$.

We further define the set $E_{\mathcal{C}}$ so that for every $x_1, x_2 \in \mathcal{X}$ we have $(x_1, x_2) \in E_{\mathcal{C}}$ if and only if x_2 is propositionally reachable from x_1.

Definition 4.5.1. Let $y \in \mathcal{X}_{\mathcal{C}}$. We say that y is *a-positively modally reachable* (respectively *a-negatively modally reachable*) from x if $\mathcal{C}$ contains a clause of the form $\Box^*(x' \rightarrow \Box_a y')$ (respectively $\Box^*(x' \rightarrow \Diamond_a y')$) where $x', y' \in \mathcal{X}_{\mathcal{C}}$, $(x, x') \in E_{\mathcal{C}}$ and $(y', y) \in E_{\mathcal{C}}$.

We define $E^{a+}_{\mathcal{C}}$ (respectively $E^{a-}_{\mathcal{C}}$) so that $(x, y) \in E^{a+}_{\mathcal{C}}$ (respectively $(x, y) \in E^{a-}_{\mathcal{C}}$) if and only if y is a-positively modally reachable (respectively a-negatively modally reachable) from x.

Let $\mathcal{C}$ be a set of SNF clauses. We can specify the modal context of a given extension variable in $\mathcal{X}_{\mathcal{C}}$ or clause in $\mathcal{C}$ using finite words over the set of clausal modal context markers for $\mathcal{C}$, $\mathcal{E}_{\mathcal{C}}$ (Definition 4.4.3).

Definition 4.5.2. Let $\mathcal{C}$ be a set of SNF clauses and let $x_\varepsilon \in \mathcal{X}_{\mathcal{C}}$ be such that $\Box^*(\mathbf{S} \rightarrow x_\varepsilon) \in \mathcal{C}$. Note that for every $\mathcal{C}$ the variable x_ε is uniquely defined. We define:

$$\mathcal{X}_\varepsilon = \{x \in \mathcal{X}_{\mathcal{C}} \mid (x_\varepsilon, x) \in E_{\mathcal{C}}\}.$$

For every $e \in \mathcal{E}^*_{\mathcal{C}}$ and every $c \in \mathcal{E}_{\mathcal{C}}$ we define:

$$\mathcal{X}_{ec} = \begin{cases} \{x \in \mathcal{X}_{\mathcal{C}} \mid (z, x) \in E^{a+}_{\mathcal{C}} \text{ for some } z \in \mathcal{X}_e\} & \text{if } c \in \mathcal{A}, \\ \{x \in \mathcal{X}_{\mathcal{C}} \mid (z, x) \in E_{\mathcal{C}}\} & \text{if } c = (a, z) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-}, \\ \emptyset & \text{otherwise.} \end{cases}$$

We say x has *modal context* e if $x \in \mathcal{X}_e$.

Definition 4.5.3. Let π be a $\mathbf{K}_n$-Res refutation of some set of SNF clauses $\mathcal{C}$ and let C be some clause in π. If C contains the start connective $\mathbf{S}$ then we say that C has *modal context* ε. Further, if some $x \in \mathcal{X}_e$ appears as a negative literal (i.e. either as $\neg x$ in a literal clause or as x on the left hand side of a $\rightarrow$ operator in a modal clause) in C the we say that C has *modal context* e.

Let $\mathcal{C}$ be a set of SNF clauses. It follows from the definition of the translation function T that each $C \in \mathcal{C}$ contains only one negative extension literal, and so every such C has only one modal context. However, using the rules of $\mathbf{K}_n$-Res it is possible to derive SNF clauses that contain two or more negative extension literals with distinct modal contexts, and so have multiple modal contexts.

4.5.2 The polynomial simulations

This subsection contains a proof that $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res, $\mathbf{K}_{mp}$-Res and $\mathbf{K}_{mc}$-Res are all p-equivalent. Proving that $\mathbf{K}_{mc}$-Res $\leq_p$ $\mathbf{K}_{mp}$-Res $\leq_p$ $\mathbf{K}_{ml}$-Res $\leq_p$ $\mathbf{K}_n$-Res is straightforward. Hence the majority of the subsection is made up of a series of lemmas that are used to prove that $\mathbf{K}_n$-Res $\leq_p$ $\mathbf{K}_{ml}$-Res $\leq_p$ $\mathbf{K}_{mp}$-Res $\leq_p$ $\mathbf{K}_{mc}$-Res. We begin by giving some definitions and results concerning the structure of $\mathbf{K}_n$-Res proofs.

Definition 4.5.4. Let π be a $\mathbf{K}_n$-Res refutation of some set of SNF clauses $\mathcal{C}$, let $\mathcal{C}_\pi$ denote the set of all clauses in π and let $C_1, C_n \in \mathcal{C}_\pi$. We say that there is a *path* from C_1 to C_n if there exists a word $C_1 C_2 \dots C_{n-1} C_n \in {\mathcal{C}_\pi}^*$ such that for each $i \in [n-1]$ the clause C_{i+1} is a child of C_i.

Lemma 4.5.1. Let $\mathcal{C}$ be a set of clauses in SNF and π be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$. If $C_2 = \Box^*(x \vee D_2)$ is a descendant of $C_1 = \Box^*(x \vee D_1) \in \mathcal{C}$, where $x \in \mathcal{X}_\mathcal{C}$ then π contains a path P from C_1 to C_2 such that every clause in P contains x.

Proof. As C_2 is a descendant of C_1 the refutation π contains a path of clauses $P_1 = A_1 \dots A_n$ where $C_1 = A_1$, $C_2 = A_n$. Let S be the longest suffix of P_1 such that every $A_j \in S$ contains x. We proceed by induction on the size of S. If $|S| = |\pi|$ then as $|S| \leq |P_1| \leq |\pi|$ it follows that $S = P_1$.

Suppose $|S| < |\pi|$ then either $S = P_1$ or $S = A_j, \dots, A_n$ where $j > 1$. In the latter case $x \notin A_{j-1}$ and so A_j must also be a child of some clause $C' \neq A_{j-1}$. Further, it follows from the definition of the translation function T that C is the only initial clause which contains a positive instance of x and so either $C' = C_1$ or C' is a descendant of C_1 containing x. Hence there exists a path of clauses P_2 from C_1 to A_j. Concatenating P_2 with $A_{j+1}, \dots, A_n$ gives a path from C_1 to C_2 with a suffix of length $\geq |S| + 1$. Hence by the inductive hypothesis there exists a path P from C_1 to C_2 such that every clause in P contains x. □

Definition 4.5.5. We say a $\mathbf{K}_n$-Res refutation:

$$\pi = C_1, \dots, C_{n-1}, C_n,$$

is in *1-start form* if it contains precisely two start clauses, namely $C_n = \Box^*(\mathbf{S} \to 0)$ and some $C_j = \Box^*(\mathbf{S} \to x_\varepsilon) \in \mathcal{C}$ where $j \in [n-1]$. Equivalently, we say π is in 1-start form if it does not contain any clauses inferred using IRES2 and C_n is the only clause in π inferred using IRES1.

Proposition 4.5.1. Let $\mathcal{C}$ be an unsatisfiable set of SNF clauses and $\pi = C_1, \dots, C_n$ be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$. From π we can efficiently construct a 1-start refutation of $\mathcal{C}$ with size less than or equal to $|\pi|$.

Proof. If π is in 1-start form then the proposition holds trivially. Hence we suppose π is not in 1-start form and proceed to construct a new refutation as follows. First we delete from π every clause that is inferred by applying IRES1 to $\Box^*(\mathbf{S} \to x_\varepsilon)$ and some literal clause $\neq \Box^*(\neg x_\varepsilon)$. Let $\mathcal{S} = \{S_1, \dots, S_m\}$, where each S_i is of the form $\Box^*(\mathbf{S} \to D_i)$, be the set of all remaining non-initial start clauses in π. Replacing each S_i in π with $\Box^*(\neg x_\varepsilon \vee D_i)$ yields a derivation of $\Box^*(\neg x_\varepsilon)$, hence by adding the clauses $\Box^*(\mathbf{S} \to x_\varepsilon)$ and $\Box^*(\mathbf{S} \to 0)$ to the end of π we obtain a valid refutation of $\mathcal{C}$ in 1-start form. □

Hence from this point onwards we will consider only $\mathbf{K}_n$-Res refutations in 1-start form.

We will now prove three technical lemmas. Let π be a $\mathbf{K}_n$-Res refutation of some unsatisfiable set of SNF clauses $\mathcal{C}$. The first of the lemmas simply states that every literal clause in π contains at least one extension variable. The second lemma says that if a clause C in π contains a negative

extension literal with modal context e then any clause inferred from C using LRES must also contain a negative extension literal with modal context e. The third lemma says that if a clause C in π contains a negative extension literal with modal context e and is propositionally reachable from some $x \in \mathcal{X}_{\mathcal{C}\pm}$, then π must also contain a clause that is an LRES descedant of C and contains the literal $\neg x$.

Lemma 4.5.2. Let $\phi \in \mathit{wfmf}$, let $\mathcal{C} = T(\phi)$ and let π be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$. Every literal clause C in π contains at least one negative extension literal.

Proof. We prove the lemma by induction on the length of the derivation of C from $\mathcal{C}$. If $C \in \mathcal{C}$ then the lemma follows from the definition of the translation function T. Suppose C is inferred using some modal inference rule. By the definition of T each modal clause used to infer C contains a negative extension literal. The clause C must further contain each of these literals by the definition of the modal rules of $\mathbf{K}_n$-Res. Suppose C is inferred using LRES. Let C_1 and C_2 be the clauses used to infer C. By the inductive hypothesis C_1 contains some negative extension literal $\neg x_1$ and C_2 contains some negative extension literal $\neg x_2$. As $\neg x_1$ and $\neg x_2$ are both negative literals they cannot be resolved with each other and so C must contain at least one of $\neg x_1$ and $\neg x_2$. □

Lemma 4.5.3. Let $\phi \in \mathit{wfmf}$, let $\mathcal{C} = T(\phi)$ and let π be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$ in 1-start form. Further let C be some literal clause in π that is inferred by applying LRES to two literal clauses $C_1 = \Box^*(\neg y_1 \vee D_1)$ and C_2, where $y_1 \in \mathcal{X}_{\mathcal{C}}$. If $x \in \mathcal{X}_{\mathcal{C}\pm}$ is such that $(x, y_1) \in E_{\mathcal{C}}$ then C is of the form $\Box^*(\neg y \vee D)$, where $D \in \mathcal{CL}$ and $y \in \mathcal{X}_{\mathcal{C}}$ is such that $(x, y) \in E_{\mathcal{C}}$ and $\mathcal{C}_{(x,y)} \subseteq \mathcal{C}_{(x,y_1)}$.

Proof. If y_1 is not the pivot variable then $C = \Box^*(\neg y_1 \vee D_2)$ and so the lemma holds trivially. Hence we suppose that y_1 is the pivot variable (and hence that C_2 contains the literal y_1) and proceed by induction on $|\mathcal{C}_{(x,y_1)}|$.

Suppose $|\mathcal{C}_{(x,y_1)}| = 1$. Then $\mathcal{C}_{(x,y_1)} = \{C' = \Box^*(x_0 \to \circ_a x_1)\}$, where $x_1 = y_1 = x$. Recall that every variable in $\mathcal{X}_{\mathcal{C}}$ appears positively in exactly one clause of $\mathcal{C}$. Further no clause containing the literal y_1 can be inferred from C' using the rules of $\mathbf{K}_n$-Res. Hence π cannot contain a literal clause containing y_1, contradicting our assumption that C_2 is such a clause and so y_1 cannot be the pivot.

Suppose $|\mathcal{C}_{(x,y_1)}| \geq 1$. The set $\mathcal{C}$ contains exactly one clause, say $C' = \Box^*(\neg y_2 \vee D \vee y_1)$ in which y_1 appears positively. Hence there exists a set:

$$\mathcal{C}_{(x,y_1)} = \{\Box^*(x_0 \to \circ_a x_1), \Box^*(\neg x_1 \vee D'_1 \vee x_2), \ldots, \\ \Box^*(\neg x_{n-2} \vee D'_{n-2} \vee x_{n-1}) \vee \Box^*(\neg x_{n-1} \vee D'_{n-1} \vee x_n)\},$$

where $x_1 = x$, $x_{n-1} = y_2$, $x_n = y_1$ and $D'_{n-1} = D$. As C_2 contains y_1 and C' is the only clause in $\mathcal{C}$ containing y_1 it follows that C_2 is a descendant of C'. Hence by Lemma 4.5.1 the refutation π must contain some path P from C' to C_2 such that every clause in P contains y_1. Thus no clause in P is inferred by resolving on y_1. As P contains no start clauses and no clauses inferred by resolving

on y_1 each clause in P must be inferred using LRES[4]. The set of clauses $\mathcal{C}_{(x,y_2)} = \mathcal{C}_{(x,y_1)} \setminus \{C'\}$ witnesses that y_2 is propositionally reachable from x.

We proceed to show by induction on (i) $|\mathcal{C}_{(x,y_2)}|$ and (ii) $|P|$ that C_2 contains some negative extension literal $\neg y$ such that $(x, y) \in E_\mathcal{C}$ and $\mathcal{C}_{(x,y)} \subseteq \mathcal{C}_{(x,y_2)}$. If $|C_{(x,y_2)}| = 1$ then $y_2 = x = x_1$. As in the case when $x_1 = y_1$ it follows that every LRES descendant of C' contains $\neg y_2$ and so we take $y = y_2$. Suppose $|C_{(x,y_2)}| > 1$. If $|P| = 0$ then $C_2 = C'$ and so $\neg y_2 \in C_2$. Suppose $|P| > 1$ and let P_1 be the path from C' to C_3 such that C_2 is a child of C_3. By inductive hypothesis (ii) C_3 contains some negative extension literal $\neg y_3$ such that $(x, y_3) \in E_\mathcal{C}$ and $\mathcal{C}_{(x,y_3)} \subseteq \mathcal{C}_{(x,y_2)}$. Thus by inductive hypothesis (i) every LRES child of C_3 must contain some y_4 such that $(x, y_4) \in E_\mathcal{C}$ and $\mathcal{C}_{(x,y_4)} \subseteq \mathcal{C}_{(x,y_3)}$. In particular C_2 must contain some such literal.

As $C_{(x,y_2)} \subset C_{(x,y_1)}$ it follows that $y \neq y_1$ and so C must contain $\neg y$. □

Lemma 4.5.4. Let $\mathcal{C}$ be a set of SNF clauses and let π be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$ in 1-start form. Suppose π contains a literal clause $C = \Box^*(\neg x \vee D_1)$, where $x \in \mathcal{X}_\mathcal{C}$ and let $x_1 \in \mathcal{X}_{\mathcal{C}\pm}$ such that $(x_1, x) \in E_\mathcal{C}$. If $\Box^*(\mathbf{S} \to 0)$ is a descendant of C then π contains a literal clause $C' = \Box^*(\neg x_1 \vee E)$ such that C' is an LRES descendant of C and $\Box^*(\mathbf{S} \to 0)$ is a descendant of C'.

Proof. We will prove the lemma by induction on $|\mathcal{C}_{(x_1,x)}|$. If $|\mathcal{C}_{(x_1,x)}| = 1$ then $x_1 = x$ and so the lemma holds trivially.

Suppose $|\mathcal{C}_{(x_1,x)}| > 1$. As $\Box^*(\mathbf{S} \to 0)$ is a descendant of C there exists some descendant of C that is inferred by resolving on x (and possibly some other variables). Let C' be the first such descendant. One of the clauses that C' is inferred from must be a descendant of C containing $\neg x$. Let C_1 denote this clause. Note that every descendant of C is non-initial and so C_1 is a literal clause. As C' is inferred by resolving on x it must also be inferred from some clause $C_2 \neq C_1$ containing the literal x. As $|\mathcal{C}_{(x_1,x)}| > 1$ no modal clause in $\mathcal{C}$ contains the literal x and so C_2 is a literal clause. Hence C' must be inferred by applying LRES to C_1 and C_2. Furthermore, as C_1 contains $\neg x$ and C' is the first descendant of C inferred by resolving on x it follows that C_1 is an LRES descendant of C. Hence by Lemma 4.5.3, C' is of the form $\Box^*(\neg x_2 \vee D)$ where $x_2 \in \mathcal{X}_\mathcal{C}$ such that $(x_1, x_2) \in E_\mathcal{C}$ and $\mathcal{C}_{(x_1,x_2)} \subseteq \mathcal{C}_{(x_1,x)}$. Further as $x \neq x_2$ we have $\mathcal{C}_{(x_1,x_2)} \neq \mathcal{C}_{(x_1,x)}$. Hence by the inductive hypothesis there exists an LRES descendant of C', and so C, of the form $\Box^*(\neg x_1 \vee E)$ that is also an ancestor of $\Box^*(\mathbf{S} \to 0)$. □

Definition 4.5.6. We say an SNF clause C has *unifiable modal contexts* if it has modal contexts $e_1, \ldots, e_n$ such that $\sigma(e_1, \ldots, e_n)$ is defined. Similarly we say C has *non-unifiable modal contexts* if $\sigma(e_1, \ldots, e_n)$ is undefined.

The following lemma is the main result of this subsection.

Lemma 4.5.5. Let $\mathcal{C}$ be an unsatisfiable set of SNF clauses and let π be a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$ in 1-start form. Let π' be the sequence of clauses obtained by deleting some clauses $C_1, \ldots, C_m$,

[4]If IRES1 or IRES2 was used to infer some $A_j \in P$ then A_j would be a start clause. MRES and GEN2 can only be applied to modal clauses. GEN1 and GEN3 both require every variable in a literal clause to be resolved on simultaneously and so if either was used to infer some A_j then $y_1 \notin A_j$.

along with every descendant of each C_i from π. If each C_i has non-unifiable modal contexts then π' is a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$.

Proof. Clearly any sequence of clauses π' that is obtained from π by removing clauses that are not ancestors of $\Box^*(\mathbf{S} \to 0)$, as well as all of their descendants, is a refutation of $\mathcal{C}$. So to prove the lemma we show that $\Box^*(\mathbf{S} \to 0)$ cannot be a descendant of any clause C in π that has non-unifiable modal contexts. As all initial clauses have unifiable modal contexts any such C is a literal clause of the form $\Box^*(\neg x_1 \vee \neg x_2 \vee D)$ where $D \in \mathcal{CL}$ and $x_1, x_2 \in \mathcal{X}_\mathcal{C}$ which has non-unifiable modal contexts. Let e_1 and e_2 be the modal contexts of x_1 and x_2 respectively. We assume without loss of generality that $|e_1| \leq |e_2|$.

Suppose C is an ancestor of $\Box^*(\mathbf{S} \to 0)$. By Lemma 4.5.4 the refutation π contains some clause $C' = \Box^*(\neg y_1 \vee D_1)$ where $y_1 \in \mathcal{X}_{e_1} \cap (\{x_\varepsilon\} \cup \mathcal{X}_{\mathcal{C}_\pm})$ and $D_1 \in \mathcal{CL}$. Further C' is both an ancestor of $\Box^*(\mathbf{S} \to 0)$ and an LRES descendant of C. As C' is an LRES descendant of C, by Lemma 4.5.3 the disjunction D_1 contains some negative extension literal $\neg x_2'$ such that $x_2' \in \mathcal{X}_{e_2}$. Thus, by Lemma 4.5.4 π also contains a clause $C'' = \Box^*(\neg y_2 \vee D_2)$ where $y_2 \in \mathcal{X}_{e_2} \cap (\{x_\varepsilon\} \cup \mathcal{X}_{\mathcal{C}_\pm})$ and $D_2 \in \mathcal{CL}$, further C'' is both an ancestor of $\Box^*(\mathbf{S} \to 0)$ and an LRES descendant of C'. As $y_1 \in \mathcal{X}_{\mathcal{C}\pm} \cup \{x_\varepsilon\}$ it cannot appear positively in any literal clause. Hence as C'' is an LRES descendant of C' the disjunction D_2 must contain $\neg y_1$. As $\Box^*(\mathbf{S} \to 0)$ is a descendant of C'' both $\neg y_1$ and $\neg y_2$ must be resolved on at some stage in π. We proceed to show by induction on $|e_1|$ that this leads to a contradiction.

Suppose $|e_1| = 0$, then $e_1 = \varepsilon$ and $|e_2| > 0$. The only initial clause containing a positive instance of $y_1 = x_\varepsilon$ (respectively y_2) is $C_1' = \Box^*(\mathbf{S} \to x_\varepsilon)$ (respectively $C_2' = \Box^*(y_2' \to \circ_a y_2)$). Further no descendant of C_1' (respectively C_2') contains the positive literal x_ε (respectively y_2). Hence $\neg x_\varepsilon$ must be resolved on using IRES1 and $\neg y_2$ must be resolved on using either GEN1 or GEN3. As π is in 1-start form $\neg y_2$ must be resolved on first. Thus either GEN1 or GEN3 must be applied to some set of clauses $\mathcal{C}' \supseteq \{C_2', C'''\}$ where C''' is a literal clause and is either C'' or some descendant of C'' containing $\neg x_\varepsilon$ and $\neg y_2$. However the inference rules GEN1 and GEN3 both require every literal in C''' to be resolved on simultaneously and so $\neg x_\varepsilon$ must also be resolved on at this step in the refutation which is impossible.

Now suppose $|e_1| > 0$. For each $i \in [2]$ the only clause in $\mathcal{C}$ in which y_i appears positively is $C_i' = \Box^*(y_i' \to \circ_{a_i} y_i)$. Further no descendant of C_i' may contain a positive instance of y_i. As both y_1 and y_2 only appear positively in modal clauses they must be resolved on simultaneously by applying either GEN1 or GEN3 to some set $\mathcal{C}' \supseteq \{C_1', C_2', C'''\}$, where C''' is a literal clause and is either C'' or some descendant of C'' containing both $\neg y_1$ and $\neg y_2$. It follows that $a_1 = a_2$ and at most one of C_1' and C_2' is a negative modal clause as otherwise neither GEN1 nor GEN3 can be applied to $\mathcal{C}'$. In particular we can assume without loss of generality that $C_1' = \Box^*(y_1' \to \Box_{a_1} y_1)$. Let e_1' and e_2' be the modal contexts of y_1' and y_2' respectively. As C_1' and C_2' are both initial clauses it follows from the definition of the translation function T that $e_1 = e_1' a_1$ and e_2 is either equal to $e_2' a_2$ or $e_1'(a_2, y_2)$. Hence as $a_1 = a_2$ we have $\sigma(a_1, a_2) = \sigma(a_1, (y_2, a_2)) = a_1$ and so as $\sigma(e_1, e_2)$ is undefined e_1' and e_2' must be non-unifiable. Any clause inferred by applying either GEN1 or

GEN3 to $\mathcal{C}'$ is a literal clause of the form $\Box^*(\neg y'_1 \vee \neg y'_2 \vee D')$, where $D' \in \mathcal{CL}$. As $|e'_1| < |e_1|$ and $\sigma(e'_1, e'_2)$ is undefined it follows by induction that $\Box^*(\mathbf{S} \to 0)$ is not a descendant of C'' and therefore cannot be a descendant of C. □

Theorem 4.5.1. $\mathbf{K}_n$-Res $\equiv_p$ $\mathbf{K}_{ml}$-Res $\equiv_p$ $\mathbf{K}_{mp}$-Res $\equiv_p$ $\mathbf{K}_{mc}$-Res.

Proof. Let ϕ be a $\mathbf{K}_n$ formula in NNF. Translating ϕ into SNF, SNF$_{ml}$, SNF$_{mp}$ and SNF$_{mc}$ we obtain four sets of clauses, denoted by $\mathcal{C}$, $\mathcal{C}_{ml}$, $\mathcal{C}_{mp}$ and $\mathcal{C}_{mc}$ respectively. There is a one to one correspondence between the clauses in each set. That is, for any for any $e \in \mathcal{E}^*_{\mathcal{C}}$ such that $|e| = m$ and C has modal position $\mu \in \mathcal{A}^*$:

$$\begin{array}{ccccc}
\Box^*(D) \in \mathcal{C} & \iff & (m : D) \in \mathcal{C}_{ml} & \iff & \Box_\mu(D) \in \mathcal{C}_{mp} \\
\Box^*(x \to \circ_a l) \in \mathcal{C} & \iff & (m : x \to \circ_a l) \in \mathcal{C}_{ml} & \iff & \Box_\mu(x \to \circ_a l) \in \mathcal{C}_{mp} \\
\Box^*(\mathbf{S} \to x_\varepsilon) \in \mathcal{C} & \iff & (0 : x_\varepsilon) \in \mathcal{C}_{ml} & \iff & x_\varepsilon \in \mathcal{C}_{mp} \\
 & & \iff & (e : D) \in \mathcal{C}_{mc}, & \\
 & & \iff & (e : x \to \circ_a l) \in \mathcal{C}_{mc}, & \\
 & & \iff & (x_\varepsilon : x_\varepsilon) \in \mathcal{C}_{mc}, &
\end{array}$$

where $x_\varepsilon, x \in \mathcal{X}_{\mathcal{C}}$, $D \in \mathcal{CL}$ and $l \in \mathcal{L}$.

($\geq_p$): Let π_{mc} be a $\mathbf{K}_{mc}$-Res refutation of $\mathcal{C}_{mc}$. If we take π_{mp}, π_{ml} and π to be the corresponding sequences of SNF$_{mp}$, SNF$_{ml}$ and SNF clauses respectively then we obtain a $\mathbf{K}_{mp}$-Res refutation of $\mathcal{C}_{mp}$, a $\mathbf{K}_{ml}$-Res refutation of $\mathcal{C}_{ml}$ and a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$ respectively.

($\leq_p$): Now suppose π is a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$ in 1-start form. Let $\pi' = C_1, \ldots, C_m$ be the sequence of clauses obtained by deleting every clause with non-unifiable modal contexts from π. By Lemma 4.5.5 π' is a $\mathbf{K}_n$-Res refutation of $\mathcal{C}$. To prove that the analogous sequence of SNF$_{mc}$ clauses[5] is a $\mathbf{K}_{mc}$-Res refutation of $\mathcal{C}_{mc}$ we must verify that each clause in π' is inferred from a set of clauses whose modal contexts agree with those required to apply the corresponding inference rule of $\mathbf{K}_{mc}$-Res.

Note that as π' is in 1-start form it cannot contain any clauses inferred using IRES2. Suppose some C in π' is inferred from two clauses C_1 and C_2 using IRES1. Then as π' is in 1-start form we can assume without loss of generality that $C_1 = \Box^*(\neg x_\varepsilon)$ and $C_2 = \Box^*(\mathbf{S} \to x_\varepsilon)$. The clauses C_1 and C_2 both have modal context ε and so LRES can be applied to the analogous SNF$_{mc}$ clauses to infer $(\varepsilon : 0)$.

Suppose $C = \Box^* C'$ is inferred by applying LRES to some C_1 and C_2 in π'. Let $\{e_1, \ldots, e_{n_1}\}$ and $\{e'_1, \ldots, e'_{n_2}\}$ be the sets of all modal contexts of C_1 and all modal contexts of C_2 respectively. Then by Lemma 4.5.3 the clause C must contain negative extension variables with modal contexts $e_1, \ldots, e_{n_1}, e'_1, \ldots, e'_{n_2}$. As C is unifiable there exists some $e \in \mathcal{E}^*_{\mathcal{C}}$ such that $e = \sigma(e_1, \ldots, e_{n_1}, e'_1, \ldots, e'_{n_2})$. Hence we can apply LRES to the analogous SNF$_{mc}$ clauses to infer $(e : C)$.

[5] That is, the sequence of clauses where each clause is labelled by the unified modal context of the corresponding SNF clause.

Suppose $C = \Box^* C'$ is inferred by applying MRES (respectively GEN2) to some C_1 and C_2 (respectively C_1, C_2 and C_3). As C_1 and C_2 (respectively C_1, C_2 and C_3) are modal clauses they must each have a single modal context. Let e_1 and e_2 (respectively e_1, e_2 and e_3) be the modal contexts of C_1 and C_2 (respectively C_1, C_2 and C_3) respectively. It follows from the definition of MRES (respectively GEN2) that C has modal contexts e_1 and e_2 (respectively e_1, e_2 and e_3). Further as C has unifiable contexts $\sigma(e_1, e_2)$ (respectively $\sigma(e_1, e_2, e_3)$) is defined. Hence we can apply MRES (respectively GEN2) to C_1 and C_2 (respectively C_1, C_2 and C_3) to infer $(\sigma(e_1, e_2) : C)$ (respectively $(\sigma(e_1, e_2, e_3) : C)$).

Finally suppose $C = \Box^*(\neg l'_1 \vee \cdots \vee \neg l'_{z+1})$ is inferred using GEN1 (respectively GEN3). Then C is inferred from z positive modal clauses $C_1 = \Box^*(l'_1 \rightarrow \Box_a l_1), \ldots, C_z = \Box^*(l'_z \rightarrow \Box_a l_z)$, one negative modal clause $C_{z+1} = \Box^*(l'_{z+1} \rightarrow \Diamond_a l_{z+1})$ and one literal clause $C_{z+2} = \Box^*(\neg l_1 \vee \cdots \vee \neg l_{z+1})$ (respectively $C_{z+2} = \Box^*(\neg l_1 \vee \cdots \vee \neg l_z)$). Each of the modal clauses must have a single modal context hence we let $e_1, \ldots, e_{z+1}$ be the modal contexts of $C_1, \ldots, C_{z+1}$ respectively. By the definition of GEN1 (respectively GEN3) C has modal contexts $e_1, \ldots, e_{z+1}$ and so as C has unifiable modal contexts there exists some $e \in \mathcal{E}^*_{\mathcal{C}}$ such that $\sigma(e_1, \ldots, e_{z+1}) = e$. Further, it follows from the definition of the translation function T that the set of modal contexts of C_{z+2} is a subset of $\{e_1 a, \ldots, e_z a, e_{z+1}(l_{z+1}, a)\}$ (respectively $\{e_1 a, \ldots, e_z a\}$). Hence we can apply GEN1 (respectively GEN3) to the set of SNF$_{mc}$ clauses corresponding to $\{C_1, \ldots, C_{z+2}\}$ to infer $(e : C)$. □

Remark 4.5.1. Due to the p-equivalence of the proof systems $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res, $\mathbf{K}_{mp}$-Res and $\mathbf{K}_{mc}$-Res we henceforth refer to them collectively as the family of **K**-Res systems.

Corollary 4.5.1. The tree-like versions of the **K**-Res are all p-equivalent to one another.

Proof. This follows from the proofs of Lemma 4.5.5 and Theorem 4.5.1. □

4.6 A further variation of the K-Res systems

In this section we show that given any of the **K**-Res system, if we restrict the definition of its associated normal form so that only positive literals can appear within the scope of a modal operator then not only can we still efficiently transform any modal formula in NNF into this restricted version of SNF$_{mp}$, but we can also define a modal resolution system which operates on this normal form and whose rules are a proper subset of the rules of $\mathbf{K}_{mp}$-Res. Furthermore this proof system is p-equivalent to $\mathbf{K}_{mp}$-Res. We only give full details of this variation for the proof system $\mathbf{K}_{mp}$-Res. However due to the close relationship between the each of the **K**-Res systems, analogous variants of the other systems can be obtained in exactly the same way.

We begin by formally defining our restricted normal form.

Definition 4.6.1. Let l, $l_j \in \mathcal{L}$ and let p be a propositional variable. A formula ϕ is in *positive separated normal form with modal positions* (SNF$^+_{mp}$) if $\phi = \bigwedge_{i=1}^{r} C_i$ where each C_i is either a:

- Positive modal clause: $\Box_\mu(l \to \Box_a p)$,
- Negative modal clause: $\Box_\mu(l \to \Diamond_a p)$,
- Literal clause: $\Box_\mu(\bigvee_{j=1}^{t} l_j)$.

To convert an NNF formula ϕ into SNF^+_{mp} we apply the translation function:

$$T^+_{mp}(\phi) = x \wedge \rho^+_{mp}(x \to \phi),$$

where x is a new propositional variable and ρ^+_{mp} is defined as follows:

$$\rho^+_{mp}(\Box_\mu(x \to \theta \wedge \psi)) = \rho^+_{mp}(\Box_\mu(x \to \theta)) \wedge \rho^+_{mp}(\Box_\mu(x \to \psi)),$$

$$\rho^+_{mp}(\Box_\mu(x \to \circ_a \theta)) = \begin{cases} \Box_\mu(x \to \circ_a \theta), & \text{if } \theta \in P, \\ \Box_\mu(x \to \circ_a x_1) \wedge \rho^+_{mp}(\Box_{\mu a}(x_1 \to \theta)), & \text{otherwise.} \end{cases}$$

$$\rho^+_{mp}(\Box_\mu(x \to \theta \vee \psi)) = \begin{cases} \Box_\mu(\neg x \vee \theta \vee \psi), & \text{if } \theta, \psi \in \mathcal{CL}, \\ \rho^+_{mp}(\Box_\mu(x \to \theta \vee x_1)) \wedge \rho^+_{mp}(\Box_\mu(x_1 \to \psi)), & \text{otherwise,} \end{cases}$$

where θ, ψ are modal formulas, x_1 is a new propositional symbol and $\mu \in \mathcal{A}^*$.

We can similarly define *positive separated normal form* (SNF^+), *positive separated normal form with modal levels* (SNF^+_{ml}) and *positive separated normal form with modal contexts* (SNF^+_{mc}).

Theorem 4.6.1. An NNF formula ϕ is satisfiable if and only if the corresponding set of SNF^+_{mp} clauses $T^+_{mp}(\phi)$ is satisfiable.

Proof. By Theorem 4.3.1 ϕ is satisfiable if and only if the set of SNF_{mp} clauses $T_{mp}(\phi)$ is. Hence to prove our theorem we show that $T_{mp}(\phi)$ is satisfiable if and only if $T^+_{mp}(\phi)$ is satisfiable.

We begin by noting that every literal clause C is in $T_{mp}(\phi)$ if and only if it is also in $T^+_{mp}(\phi)$. Similarly a modal clause of the form $\Box_\mu(l \to \circ_a p)$, where $l \in \mathcal{L}$ and $p \in P$, is in the set $T_{mp}(\phi)$ if and only if it is also in the set $T^+_{mp}(\phi)$. Finally any modal clause of the form $\Box_\mu(l \to \circ_a \neg p)$ is in $T_{mp}(\phi)$ if and only if $T^+_{mp}(\phi)$ contains the clauses $C^+_1 = \Box_\mu(l \to \circ_a x)$ and $C^+_2 = \Box_{\mu a}(\neg x \vee \neg p)$, where x is an extension variable which does not appear in any clause in $T_{mp}(\phi)$ and only appears in the clauses C_1 and C_2 in $T^+_{mp}(\phi)$.

($\Rightarrow$): Suppose there exist some model $M = (W, R_1, \ldots, R_n, V)$ and some world $w \in W$ such that $(M, w) \models T_{mp}(\phi)$. Then $(M, w) \models T^+_{mp}(\phi) \cap T_{mp}(\phi)$. Let $C^+ \in T^+_{mp}(\phi) \setminus T_{mp}(\phi)$. Then C^+ is either a modal clause of the form $\Box_\mu(l \to \circ_a x)$ or a literal clause of the form $\Box_{\mu a}(\neg x \vee \neg p)$, where $l \in \mathcal{L}$, $p \in P$ and x is an extension variable. Further l and p must be such that the SNF_{mp} clause $C = \Box_\mu(l \to \circ_a \neg p)$ is contained within $T_{mp}(\phi) \setminus T^+_{mp}(\phi)$.

Suppose $C = \Box_\mu(l \to \Diamond_a \neg p)$. As $(M, w) \models C$ it follows that at every world $w_\mu \in W$ such that w_μ is μ-accessible from w either $V(w_\mu)(l) = 0$, or $V(w_\mu)(l) = 1$ and there exists some world $w_{\mu a} \in W$ which is a-accessible from w_μ and for which $V(w_{\mu a})(p) = 0$. Hence we let M' be the model obtained by extending M so that if $V(w_\mu)(l) = 0$ then at every world $w'_{\mu a} \in W$ which is a-accessible from w_μ we have $V(w'_{\mu a})(x) = 0$, and if $V(w_\mu)(l) = 1$ then we have $V(w_{\mu a})(x) = 1$

and $V(w'_{\mu a})(x) = 0$ for every world $w'_{\mu a} \neq w_{\mu a}$ which is a-accessible from w_μ. For every w_μ we have $(M', w_\mu) \models l \rightarrow \Diamond_a x$ and for every world $w''_{\mu a} \in W$ which is μa-accessible from w we have $(M', w''_{\mu a}) \models \neg x \vee \neg p$. Hence $(M', w) \models (C_1^+ \wedge C_2^+)$.

Suppose $C = \Box_\mu(l \rightarrow \Box_a \neg p)$. As $(M, w) \models C$ it follows that at every world $w_\mu \in W$ such that w_μ is μ-accessible from w either $V(w_\mu)(l) = 0$, or $V(w_\mu)(l) = 1$ and $V(w'_{\mu a})(p) = 0$ for every world $w'_{\mu a} \in W$ which is a-accessible from w_μ. Let M' denote the model obtained by extending M so that for every w_μ if $V(w_\mu)(l) = 0$ then for every a-accessible world $w'_{\mu a}$ we let $V(w'_{\mu a})(x) = 0$, and if $V(w_\mu)(l) = 1$ then for every $w'_{\mu a}$ we let $V(w'_{\mu a})(x) = 1$. Hence for every w_μ we have $(M', w_\mu) \models l \rightarrow \Box_a x$ and for every $w''_{\mu a} \in W$ which is μa-accessible from w we have $(M', w''_{\mu a}) \models \neg x \vee \neg p$. It follows that $(M', w) \models (C_1^+ \wedge C_2^+)$.

($\Leftarrow$): Suppose $(M, w) \models T^+_{mp}(\phi)$. Then $(M, w) \models T^+_{mp}(\phi) \cap T_{mp}(\phi)$. Let $C \in T_{mp}(\phi) \setminus T^+_{mp}(\phi)$. Then C must be of the form $\Box_\mu(l \rightarrow \circ_a \neg p)$, where $l \in \mathcal{L}$ and $p \in P$. Further $T^+_{mp}(\phi) \setminus T_{mp}(\phi)$ must contain the clauses $\Box_\mu(l \rightarrow \circ_a x)$ and $\Box_{\mu a}(\neg x \vee \neg p)$.

Suppose $C = \Box_\mu(l \rightarrow \Diamond_a \neg p)$ (respectively $C = \Box_\mu(l \rightarrow \Box_a \neg p)$). Then by assumption $(M, w) \models (l \rightarrow \Diamond_a x)$ (respectively $\Box_\mu(l \rightarrow \Box_a x)$) hence for every world $w_\mu \in W$ which is μ-accessible from w either $V(w_\mu)(l) = 0$ or there exists some world $w_{\mu a} \in W$ such that $V(w_{\mu a})(x) = 1$ (respectively for every world $w'_{\mu a} \in W$ which is a-accessible from w_μ we have $V(w'_{\mu a})(x) = 1$). In the former case $(M, w) \models C$. In the latter case we note that as $(M, w) \models \Box_{\mu a}(\neg x \vee \neg p)$ and $w_{\mu a}$ (respectively every $w'_{\mu a}$) is μa-accessible from w it follows that $V(w_{\mu a})(p) = 0$ (respectively $V(w'_{\mu a})(p) = 0$) and so $(M, w) \models C$. □

An identical proof can be used to show that each of the other positive separated normal forms preserves satisfiability.

We can now define our new proof system.

Definition 4.6.2. The proof system $\mathbf{K}^+_{mp}$-Res consists of the rules LRES, GEN1 and GEN3 defined as in Figure 4.3.

Similarly the proof system $\mathbf{K}^+_n$-Res consists of the rules IRES1, IRES2, LRES, GEN1 and GEN3 as defined in Figure 4.1, and the proof systems $\mathbf{K}^+_{ml}$-Res and $\mathbf{K}^+_{mc}$-Res consist of the rules LRES, GEN1 and GEN3 as defined in Figures 4.2 and 4.4 respectively.

Theorem 4.6.2. The proof system $\mathbf{K}^+_{mp}$-Res is complete and strongly sound.

Proof. Any proof system that p-simulates a complete proof system is complete and any proof system that is p-simulated by a strongly sound proof system is strongly sound. By Theorems 4.3.2 and 4.3.4 the proof system $\mathbf{K}_{mp}$-Res is strongly sound and complete, hence the theorem follows immediately from the fact that $\mathbf{K}^+_{mp}$-Res both p-simulates and is p-simulated by $\mathbf{K}_{mp}$-Res (Theorem 4.6.3). □

The strong soundness and the completeness of $\mathbf{K}^+_n$-Res, $\mathbf{K}_{ml}$-Res and $\mathbf{K}_{mc}$-Res follows in the same way.

Finally we prove in the following theorem that our new proof system $\mathbf{K}^+_{mp}$-Res is p-equivalent to $\mathbf{K}_{mp}$-Res.

Theorem 4.6.3. The proof systems $\mathbf{K}^+_{mp}$-Res and $\mathbf{K}_{mp}$-Res are p-equivalent.

Proof. As the rules of $\mathbf{K}_{mp}$-Res are a superset of those of $\mathbf{K}^+_{mp}$-Res and every set of SNF^+_{mp} clauses is in SNF_{mp} it follows trivially that $\mathbf{K}_{mp}$-Res p-simulates $\mathbf{K}^+_{mp}$-Res. Hence to prove that the two proof systems are p-equivalent it remains to show that $\mathbf{K}^+_{mp}$-Res p-simulates $\mathbf{K}_{mp}$-Res.

Let $\mathcal{C}$ be an unsatisfiable set of SNF_{mp} clauses, obtained by applying the translation function T_{mp} to some modal formula ϕ in NNF, and let $\mathcal{C}^+$ be the corresponding set of SNF^+_{mp} clauses obtained by applying T^+_{mp} to ϕ. Further let π be a $\mathbf{K}_{mp}$-Res refutation of $\mathcal{C}$. We will show that from π we can efficiently construct a $\mathbf{K}^+_{mp}$-Res refutation π^+ of $\mathcal{C}^+$ whose size is polynomial in that of π.

To show that we can construct such a refutation it suffices to show that any clause C in π which is inferred using some clause in $\mathcal{C} \setminus \mathcal{C}^+$ can be derived from $\mathcal{C}^+$ using only the rules of $\mathbf{K}^+_{mp}$-Res, and that this derivation has size linear in that of C. Every literal clause in $\mathcal{C}$ is in $\mathcal{C}^+$. In fact the only clauses that are in $\mathcal{C} \setminus \mathcal{C}^+$ are modal clauses of the form $\Box_\mu(l \to \circ_a \neg p)$, where $p \in P$. Further if $\Box_\mu(l \to \circ_a \neg p) \in \mathcal{C}$ then the SNF^+_{mp} clauses $\Box_\mu(l \to \circ_a x)$ and $\Box_{\mu a}(\neg x \vee \neg p)$ must be contained within $\mathcal{C}^+$, where x is an extension variable.

Suppose π contains some SNF_{mp} clause C which is inferred from some set of clauses containing at least one modal clause in $\mathcal{C} \setminus \mathcal{C}^+$. By definition no such clause can be inferred using LRES and so C must be inferred using either MRES, GEN1, GEN2 or GEN3.

Suppose C is inferred using MRES. Then we replace this instance of MRES in π with an instance of GEN1 as shown in Figure 4.5.

$$\text{MRES}\ \frac{\Box_\mu(l_1 \to \circ_a p) \quad \Box_\mu(l_2 \to \circ'_a \neg p)}{\Box_\mu(\neg l_1 \vee \neg l_2)} \quad \longmapsto \quad \text{GEN1}\ \frac{\Box_\mu(l_1 \to \circ_a p) \quad \Box_\mu(l_2 \to \circ'_a x) \quad \Box_{\mu a}(\neg x \vee \neg p)}{\Box_\mu(\neg l_1 \vee \neg l_2)}$$

where $l_1,\ l_2 \in \mathcal{L}$, $p \in P$ and x is an extension variable. Further $\circ'_a$ is the dual of $\circ_a$, i.e. if $\circ_a = \Box_a$ then $\circ'_a = \Diamond_a$ and if $\circ_a = \Diamond_a$ then $\circ'_a = \Box_a$.

Figure 4.5: Transformation of MRES

If C in inferred using GEN2 then we replace this instance of GEN2 with an instance of GEN3 as shown in Figure 4.6. Note that the transformation depends on the number of modal clauses in $\mathcal{C} \setminus \mathcal{C}^+$ which the instance of GEN2 being transformed is applied to.

If C in inferred using GEN1 then we replace this instance of GEN1 with several instances of LRES, followed by another instance of GEN1 as shown in Figure 4.7. The exact number of LRES steps required is the same as the number of clauses in $\mathcal{C} \setminus \mathcal{C}^+$ contained within the premises of the original instance of GEN1.

Similarly if C is inferred using GEN3 then we replace this instance of GEN3 with several instances of LRES, followed by another instance of GEN3. The set of clauses that GEN3 is applied to can be assumed to be identical to the set of clauses that GEN1 is applied to in Figure 4.7 with

Case 1: GEN2 $\dfrac{\Box_\mu(l_1 \to \Box_a p) \quad \Box_\mu(l_2 \to \Box_a \neg p) \quad \Box_\mu(l_3 \to \Diamond_a p')}{\Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)} \longmapsto$ GEN3 $\dfrac{\Box_\mu(l_1 \to \Box_a p) \quad \Box_\mu(l_2 \to \Box_a x) \quad \Box_\mu(l_3 \to \Diamond_a p') \quad \Box_{\mu a}(\neg x \vee \neg p)}{\Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)}$

Case 2: GEN2 $\dfrac{\Box_\mu(l_1 \to \Box_a p) \quad \Box_\mu(l_2 \to \Box_a \neg p) \quad \Box_\mu(l_3 \to \Diamond_a \neg p')}{\Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)} \longmapsto$ GEN3 $\dfrac{\Box_\mu(l_1 \to \Box_a p) \quad \Box_\mu(l_2 \to \Box_a x) \quad \Box_\mu(l_3 \to \Diamond_a x') \quad \Box_{\mu a}(\neg x \vee \neg p)}{\Box_\mu(\neg l_1 \vee \neg l_2 \vee \neg l_3)}$

where l_1, l_2 and l_3 are literals, p and p' are propositional variables and x and x' are extension variables.

Figure 4.6: Transformation of GEN2

the exception of the literal clause whose form depends on which of the modal clauses is a negative modal clause. This transformation is almost identical to that of GEN1 shown in Figure 4.7, however:

1. If the negative modal clause which GEN3 is applied to is of the form $\Box_\mu(l'_i \to \Diamond p_i)$ where $i > y$ then we assume without loss of generality that $i = z$ and replace the literal clause $\Box_{\mu a}\left(\bigvee_{i=1}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)$ with $\Box_{\mu a}\left(\bigvee_{i=1}^{y} p_i \vee \bigvee_{j=y+1}^{z-1} \neg p_j\right)$. We further replace each clause inferred using LRES with the corresponding new resolvent. Finally we change the application of GEN1 in the transformed derivation to an application of GEN3.

2. Whereas, if the negative modal clause which GEN3 is applied to is of the form $\Box_\mu(l'_i \to \Diamond \neg p_i)$ where $i \leq y$ then we assume without loss of generality that $i = y$ and replace the literal clause $\Box_{\mu a}\left(\bigvee_{i=1}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)$ with $\Box_{\mu a}\left(\bigvee_{i=1}^{y-1} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)$. We further replace each of the first $y - 1$ clauses inferred using LRES with the corresponding new resolvent and remove the yth application of LRES from the transformed derivation. Finally we change the of GEN1 in the transformed derivation to an application of GEN3.

Hence we take π^+ to be the $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}^+$ obtained by replacing each clause C which is inferred from some $C' \in \mathcal{C} \setminus \mathcal{C}^+$ with the corresponding $\mathbf{K}^+_{mp}$-Res derivation of C from $\mathcal{C}^+$. □

Identical proofs can be used to show that the proof systems $\mathbf{K}^+_n$-Res, $\mathbf{K}^+_{ml}$-Res and $\mathbf{K}^+_{mc}$-Res are p-equivalent to $\mathbf{K}_n$-Res, $\mathbf{K}_{ml}$-Res and $\mathbf{K}_{mc}$-Res respectively.

Remark 4.6.1. We henceforth refer to the family of all the original **K**-Res systems together with the systems $\mathbf{K}^+_n$-Res, $\mathbf{K}^+_{ml}$-Res, $\mathbf{K}^+_{mp}$-Res and $\mathbf{K}^+_{mc}$-Res, as the family of **K**-Res systems.

Corollary 4.6.1. The **K**-Res proof systems are all p-equivalent.

$$\text{GEN1}\ \frac{\begin{array}{c}\Box_\mu(l_1 \to \circ_a \neg p_1)\\ \vdots\\ \Box_\mu(l_y \to \circ_a \neg p_y)\\ \Box_\mu(l_{y+1} \to \circ_a p_{y+1})\\ \vdots\\ \Box_\mu(l_z \to \circ_a p_z)\\ \Box_{\mu a}\left(\bigvee_{i=1}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)\end{array}}{\Box_\mu\left(\bigvee_{j=1}^{z} \neg l_j\right)} \longmapsto \frac{\begin{array}{c}\vdots\\ \Box_{\mu a}(\bigvee_{i=1}^{y-1} \neg x_i \vee p_y \vee \bigvee_{j=y+1}^{z} \neg p_j)\\ \Box_{\mu a}(\neg x_y \vee \neg p_y)\\ \Box_{\mu a}(\bigvee_{i=1}^{y} \neg x_i \vee \bigvee_{j=y+1}^{z} \neg p_j)\\ \Box_\mu(l_1 \to \circ_a x_1)\\ \vdots\\ \Box_\mu(l_y \to \circ_a x_y)\\ \Box_\mu(l_{y+1} \to \circ_a p_{y+1})\\ \vdots\\ \Box_\mu(l_z \to \circ_a p_z)\end{array}}{\Box_\mu\left(\bigvee_{j=1}^{z} \neg l_j\right)}$$

$$\text{LRES}\ \frac{\begin{array}{c}\Box_{\mu a}\left(\bigvee_{i=1}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)\\ \Box_{\mu a}(\neg x_1 \vee \neg p_1)\end{array}}{\Box_{\mu a}\left(\neg x_1 \vee \bigvee_{i=2}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)}$$

$$\text{LRES}\ \frac{\begin{array}{c}\Box_{\mu a}\left(\neg x_1 \vee \bigvee_{i=2}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j\right)\\ \Box_{\mu a}(\neg x_2 \vee \neg p_2)\end{array}}{\Box_{\mu a}(\neg x_1 \vee \neg x_2 \vee \bigvee_{i=3}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j)}$$

$$\text{LRES}\ \frac{\begin{array}{c}\Box_{\mu a}(\neg x_1 \vee \neg x_2 \vee \bigvee_{i=3}^{y} p_i \vee \bigvee_{j=y+1}^{z} \neg p_j)\\ \vdots\end{array}}{\Box_{\mu a}(\bigvee_{i=1}^{y-1} \neg x_i \vee p_y \vee \bigvee_{j=y+1}^{z} \neg p_j)}$$

where $l_1, \ldots, l_z \in \mathcal{L}$, $p_1, \ldots, p_z \in P$ and $x_1, \ldots, x_y$ are extension variables. Further note that exactly one clause on the left is of the form $\Box_\mu(l_i \to \Diamond_a l'_i)$, where $l'_i = p_i$ if $i \leq y$ and $l'_i = \neg p_i$ if $i > y$. Hence the corresponding clause on the right is of the form $\Box_\mu(l_i \to \Diamond_a l''_i)$, where $l''_i = x_i$ if $i \leq y$ and $l''_i = p_i$ if $i > y$, and every other modal clause is a positive modal clause.

Figure 4.7: Transformation of GEN1

Corollary 4.6.2. The tree-like versions of all of the **K**-Res systems are p-equivalent.

Proof. This follows from Corollary 4.5.1 the proof of Theorem 4.6.3. □

Chapter 5

Comparing the family of K-Res proof systems with the proof system $\mathbf{RK}_\eta$

In this chapter we define one of the earliest proposed clausal resolution system for the multimodal logic $\mathbf{K}_n$, which was introduced by Enjalbert and Fariñas del Cerro in [34]. We then use p-simulations to compare the strength of this system to that of the **K**-Res systems discussed in Chapter 4.

5.1 The proof system $\mathbf{RK}_n$

In [33, 34] Enjalbert and Fariñas del Cerro proposed a clausal resolution systems for $\mathbf{K}_n$, called $\mathbf{RK}_n$. Like the **K**-Res systems, $\mathbf{RK}_n$ operates only on formulas that have been transformed into a specific normal form, however both the rules of $\mathbf{RK}_n$ and the normal form that it operates on ($\mathbf{RK}_n$ conjunctive normal form) are more complicated than those of the **K**-Res systems.

Definition 5.1.1 ([33])**.** A modal formula ϕ is in ***RK***$_n$ *conjunctive normal form* ($\mathbf{RK}_n$ CNF) if $\phi = \bigwedge_{i=1}^{N} C_i$, where each C_i is an ***RK***$_n$ *clause*. A formula C is an $\mathbf{RK}_n$ clause if:

$$C = l_1 \vee \cdots \vee l_m \vee \bigvee_{x=1}^{p} \Box_{a_x} D_x \vee \bigvee_{y=1}^{q} \Diamond_{a_y} A_y,$$

where each $l_i \in \mathcal{L}$, each D_x is an $\mathbf{RK}_n$ clause and each A_y is in $\mathbf{RK}_n$ CNF.

The proof system $\mathbf{RK}_n$ consists of three different types of rules, namely rules for computing resolvents (RFCR), simplification rules and inference rules. The RFCR consist of axioms, Σ rules, which are used to compute resolvents of a pair of clauses, and Γ rules which are used to compute

resolvents of a single clause. Note that we are able to compute resolvents of a single clause as an $\mathbf{RK}_n$ clause may contain an $\mathbf{RK}_n$ CNF within the scope of some $\Diamond_a$ and so can contain two complementary literals with the same "modal context". For example the complementary literals x and $\neg x$ both appear within the same modal context in the $\mathbf{RK}_n$ clause $\Diamond_a((x \vee y) \wedge (\neg x \vee z))$.

After computing a resolvent using the RFCR the simplification rules are used to simplify said resolvent. Only when a resolvent has been simplified as much as possible can it be inferred using an inference rule of $\mathbf{RK}_n$.

Definition 5.1.2 ([34])**.** The rules of $\mathbf{RK}_n$ are given in Figure 5.1.

Rules for Computing Resolvents (RFCR)

Axioms

A1: $\Sigma(p, \neg p) \to 0$ **A2**: $\Sigma(0, A) \to 0$

Σ rules

$\vee$: $\dfrac{\Sigma(A, B) \to C}{\Sigma(A \vee D, B \vee D') \to C \vee D \vee D'}$ $\Box\Box_a$: $\dfrac{\Sigma(A, B) \to C}{\Sigma(\Box_a A, \Box_a B) \to \Box_a C}$

$\Box\Diamond_a$: $\dfrac{\Sigma(A, B) \to C}{\Sigma(\Box_a A, \Diamond_a(B \wedge E)) \to \Diamond_a(B \wedge C \wedge E)}$

$\Diamond\Box_a$: $\dfrac{\Sigma(A, B) \to C}{\Sigma(\Diamond_a(A \wedge E), \Box_a B) \to \Diamond_a(A \wedge C \wedge E)}$

Γ rules

$\Box_a$: $\dfrac{\Gamma(A) \to B}{\Gamma(\Box_a A) \to \Box_a B}$ $\vee$: $\dfrac{\Gamma(A) \to B}{\Gamma(A \vee D) \to B \vee D}$

$\Diamond_a 1$: $\dfrac{\Sigma(A, B) \to C}{\Gamma(\Diamond_a(A \wedge B \wedge F)) \to \Diamond_a(A \wedge B \wedge C \wedge F)}$

$\Diamond_a 2$: $\dfrac{\Gamma(A) \to B}{\Gamma(\Diamond_a(A \wedge F)) \to \Diamond_a(A \wedge B \wedge F)}$

Simplification Rules

S1: $\Diamond_a 0 \approx 0$, **S2**: $0 \vee C \approx C$, **S3**: $0 \wedge E \approx 0$, **S4**: $A \vee A \vee C \approx A \vee C$.

Inference rules

R1: $\dfrac{C}{D}$ if $\Gamma(C) \Rightarrow D$, **R2**: $\dfrac{B, \quad C}{D}$ if $\Sigma(B, C) \Rightarrow D$.

where A, B, C, D, D' are $\mathbf{RK}_n$ clauses, E, F are $\mathbf{RK}_n$ CNFs and p is a propositional variable.

Figure 5.1: Rules for $\mathbf{RK}_n$

The simplification relation $\approx$ is defined to be the least equivalence relation generated by the

simplification rules. For every $\mathbf{RK}_n$ clause C there exists a unique $\mathbf{RK}_n$ clause C' which is such that $C' \approx C$ and no further simplification rules may be applied to C'. We call C' the *normal form* of C.

We say an $\mathbf{RK}_n$ clause D is a *resolvent* of two $\mathbf{RK}_n$ clauses B and C if by repeatedly applying the RFCR of $\mathbf{RK}_n$ to either **A1** or **A2** we can obtain $\Sigma(B,\, C) \rightarrow D'$ where D' is an $\mathbf{RK}_n$ clause and the normal form of D' is D. We further denote that D is a resolvent of B and C by $\Sigma(B,\, C) \Rightarrow D$. Similarly we say that D is a *resolvent* of a single $\mathbf{RK}_n$ clause C if by repeatedly applying the RFCR to either **A1** or **A2** we can obtain $\Gamma(C) \rightarrow D'$ where D' is an $\mathbf{RK}_n$ clause and the normal form of D' is D. This is denoted by $\Gamma(C) \Rightarrow D$.

We refer to any formula which is derived from some axiom using the RFCR and the simplification rules as a *formula for computing resolvents* (FFCR). Such a formula is either a Σ FFCR, that is a FFCR of the form $\Sigma(A,\, B) \rightarrow C$ or $\Sigma(A,\, B) \Rightarrow C$, or a Γ FFCR, that is a FFCR of the form $\Gamma(A) \rightarrow C$ or $\Gamma(A) \Rightarrow C$, where A, B and C are $\mathbf{RK}_n$ clauses. Further for any Σ FFCR we refer to A, B and C as the *1st assumed clause*, the *2nd assumed clause* and the *computed clause* respectively. Similarly, for any Γ FFCR we refer to A as the *assumed clause* and C as the *computed clause*.

Essentially, a Σ FFCR of the form $\Sigma(C_1, C_2) \rightarrow C_3$ says that given any model $M = (W, R_1, \ldots, R_n, V)$ and $w \in W$ such that $(M, w) \models C_1 \wedge C_2$ we have $(M, w) \models C_3$. Similarly a Γ FFRC of the form $\Gamma(C_1) \rightarrow C_2$ says that given any model $M = (W, R_1, \ldots, R_n, V)$ and $w \in W$ such that $(M, w) \models C_1$ we have $(M, w) \models C_2$.

The axioms **A1** and **A2** are both propositional inconsistencies. The first says that given a literal and its negation then we can compute 0, the second says that given 0 and any other $\mathbf{RK}_n$ clause we can compute 0.

The Σ $\vee$-rule says that if from the $\mathbf{RK}_n$ clauses A and B we can compute C, then from the $\mathbf{RK}_n$ clauses $A \vee D$ and $B \vee D'$ we can compute $C \vee D \vee D'$.

The Σ $\Box\Box_a$-rule states that if from two $\mathbf{RK}_n$ clauses A and B we can compute C then from $\Box_a A$ and $\Box_a B$ we can compute $\Box_a C$.

The Σ $\Box\Diamond_a$-rule says if from two $\mathbf{RK}_n$ clauses A and B we can compute C then from $\Box_a A$ and $\Diamond_a(B \wedge E)$, where E is some $\mathbf{RK}_n$ CNF, we can compute $\Diamond_a(B \wedge C \wedge E)$. Intuitively we can think of applying this rule as moving our computation to some accessible modal world. Note that we can choose not to add any E when applying this rule, obtaining a FFCR of the form $\Sigma(\Box_a A, \Diamond_a B) \rightarrow \Diamond_a(B \wedge C)$.

Similarly, the Σ $\Diamond\Box_a$-rule states that if from A and B we can compute C then from $\Diamond_a(A \wedge E)$ and $\Box_a B$ we can compute $\Diamond_a(A \wedge C \wedge E)$.

There is no Σ rule that prefixes both the 1st and the 2nd assumed clause by $\Diamond_a$. Such a rule would not be sound as two $\mathbf{RK}_n$ clauses of the form $\Diamond_a A$ and $\Diamond_a B$ do not have the same modal context, even if A and B do. However, the Γ $\Diamond_a 1$-rule transforms a Σ FFCR into a Γ FFCR by prefixing the conjunction of the 1st and 2nd assumed clauses with $\Diamond_a$. The rule states that if from A and B we can compute C then from $\Diamond_a(A \wedge B \wedge E)$ we can compute $\Diamond_a(A \wedge B \wedge C \wedge E)$, where E

is some $\mathbf{RK}_n$ CNF. This is sound as if both of the assumed clauses are true in the same a-accessible world then the computed clause must also be true in this world.

The remaining Γ rules are similar to the corresponding Σ rules. The Γ $\Diamond_a 2$-rule prefixes the assumed clause and the computed clause with $\Diamond_a$, effectively moving both into some a-accessible world. The Γ $\vee$-rule adds $\vee D$ to both the assumed clause and the computed clause, and the Γ $\Box_a$-rule allows us to prefix the assumed clause and computed clause with $\Box_a$.

The simplification rules may be applied to any subformula of a FFCR of the appropriate form. Any FFCR obtained by the application of a simplification rule is equivalent to the FFCR said rule was applied to. The purpose of the simplification rules is to convert the assumed clause(s) and the computed clause into their normal form. We can assume without loss of generality that simplification rules are only applied to a FFCR after all the necessary RCFC have been applied.

The inference rule **R1** allows us to infer a resolvent of a single $\mathbf{RK}_n$ clause, which has been computed using the RFCR and then fully simplified using the simplification rules. The rule **R2** allows us to do the same for a resolvent of a pair of $\mathbf{RK}_n$ clauses.

Example 5.1.1. We can use the rules of $\mathbf{RK}_n$ to infer the resolvent of the $\mathbf{RK}_n$ clauses $C_1 = l_1 \vee l_2$ and $C_2 = \neg l_1 \vee l_2$ as follows:

$$\begin{aligned}
&\Sigma(l_1,\ \neg l_1) \to 0, && \textbf{A1},\\
&\Sigma(l_1 \vee l_2,\ \neg l_1 \vee l_2) \to 0 \vee l_2 \vee l_2, && \Sigma\ \vee\text{-rule},\\
&\Sigma(l_1 \vee l_2,\ \neg l_1 \vee l_2) \Rightarrow l_2, && \textbf{S2},\ \textbf{S4},\\
&l_2, && \textbf{R2 on } C_1,\ C_2.
\end{aligned}$$

Example 5.1.2. Let $\Box\Box_\mu$ abbreviate the Σ rules $\Box\Box_{a_1}, \ldots, \Box\Box_{a_m}$. We can infer $\Box_\mu(\neg l_1 \vee \neg l_2)$ from $C_1 = \Box_\mu(\neg l_1 \vee \Box_a l)$ and $C_2 = \Box_\mu(\neg l_2 \vee \neg\Box_a l)$ as follows:

$$\begin{aligned}
&\Sigma(l,\ \neg l) \to 0, && \textbf{A1},\\
&\Sigma(\Box_a l,\ \Diamond_a \neg l) \to \Diamond_a(0 \wedge \neg l), && \Sigma\ \Box\Diamond_a\text{-rule},\\
&\Sigma(\neg l_1 \vee \Box_a l,\ \neg l_2 \vee \Diamond_a \neg l) \to \neg l_1 \vee \neg l_2 \vee \Diamond_a(0 \wedge \neg l), && \Sigma\ \vee\text{-rule},\\
&\Sigma(\Box_\mu(\neg l_1 \vee \Box_a l),\ \Box_\mu(\neg l_2 \vee \Diamond_a \neg l)) \to \Box_\mu(\neg l_1 \vee \neg l_2 \vee \Diamond_a(0 \wedge \neg l)), && \Sigma\ \Box\Box_\mu\text{-rules},\\
&\Sigma(\Box_\mu(\neg l_1 \vee \Box_a l),\ \Box_\mu(\neg l_2 \vee \Diamond_a \neg l)) \Rightarrow \Box_\mu(\neg l_1 \vee \neg l_2), && \textbf{S3},\ \textbf{S1},\ \textbf{S2},\\
&\Box_\mu(\neg l_1 \vee \neg l_2), && \textbf{R2 on } C_1,\ C_2.
\end{aligned}$$

Example 5.1.3. We can use the rules of $\mathbf{RK}_n$ to infer the $\mathbf{RK}_n$ clause $\Diamond_a(l_1 \wedge (\neg l_1 \vee l_2) \wedge l_2)$ from

$C_1 = \Diamond_a(l_1 \wedge (\neg l_1 \vee l_2))$ as follows:

$$\begin{aligned}
&\Sigma(l_1,\ \neg l_1) \to 0, && \textbf{A1},\\
&\Sigma(l_1,\ \neg l_1 \vee l_2) \to 0 \vee l_2, && \Sigma\ \vee\text{-rule},\\
&\Gamma(\Diamond_a(l_1 \wedge (\neg l_1 \vee l_2))) \to \Diamond_a(l_1 \wedge (\neg l_1 \vee l_2) \wedge (0 \vee l_2)), && \Gamma\ \Diamond_a 1\text{-rule},\\
&\Gamma(\Diamond_a(l_1 \wedge (\neg l_1 \vee l_2))) \Rightarrow \Diamond_a(l_1 \wedge (\neg l_1 \vee l_2) \wedge l_2), && \textbf{S2},\\
&\Diamond_a(l_1 \wedge (\neg l_1 \vee l_2) \wedge l_2), && \textbf{R1 on } C_1.
\end{aligned}$$

Example 5.1.4. We can use the rules of **RK**$_n$ to infer the **RK**$_n$ clause $\Diamond_a(l_2 \wedge l_1)$ from $C_1 = \Box_a l_1$ and $C_2 = \Diamond_a l_2$ as follows:

$$\begin{aligned}
&\Sigma(0,\ l_2) \to 0, && \textbf{A2},\\
&\Sigma(l_1,\ l_2) \to 0 \vee l_1, && \Sigma\ \vee\text{-rule},\\
&\Sigma(\Box_a l_1,\ \Diamond_a l_2) \to \Diamond_a(l_2 \wedge (0 \vee l_1)), && \Sigma\ \Box\Diamond_a\text{-rule},\\
&\Sigma(\Box_a l_1,\ \Diamond_a l_2) \Rightarrow \Diamond_a(l_2 \wedge l_1), && \textbf{S2},\\
&\Diamond_a(l_2 \wedge l_1), && \textbf{R2 on } C_1,\ C_2.
\end{aligned}$$

Theorem 5.1.1 ([34])**.** The proof system **RK**$_n$ is complete and strongly sound.

Definition 5.1.3. An ***RK***$_n$ *derivation* of some **RK**$_n$ clause C_m from some set of **RK**$_n$ clauses $\mathcal{C}$ is a sequence of formulas:

$$\pi = R_{(1,1)}, \ldots R_{(1,k_1)}, C_1, \ldots, C_{m-1}, R_{(m,1)}, \ldots, R_{(m,k_m)}, C_m,$$

were each C_i is an **RK**$_n$ clause in normal form and each $R_{(i,j)}$ is a FFCR. In particular $C_m = 0$. Further each C_i is either in $\mathcal{C}$, or is inferred by applying an inference rule to either some pair of clauses C_{i_1} and C_{i_2}, or a single clause C_{i_1}, where $i_1, i_2 < i$.

If $C_i \in \mathcal{C}$ then the sequence of FFCRs immediately proceeding C_i is empty. Otherwise, either $R_{(i,1)} =$ **A1** or $R_{(i,1)} =$ **A2**, and for each $j \in [k_i - 1]$ the FFCR $R_{(i,j+1)}$ must be the result of applying either a RFCR or a simplification rule to $R_{(i,j)}$. Finally if C_i is inferred from two **RK**$_n$ clauses C_{i_1} and C_{i_2} then $R_{(i,k_i)} = \Sigma(C_{i_1}, C_{i_2}) \Rightarrow C_i$, whereas if C_i is inferred from just C_{i_1} then $R_{(i,k_i)} = \Gamma(C_{i_1}) \Rightarrow C_i$.

An ***RK***$_n$ *refutation* of some unsatisfiable set of **RK**$_n$ clauses $\mathcal{C}$ is an **RK**$_n$ derivation of 0 from the set $\mathcal{C}$.

Definition 5.1.4. Let $\mathcal{C}$ be an **RK**$_n$ CNF and let π be an **RK**$_n$ refutation of $\mathcal{C}$. Further let C be an **RK**$_n$ clause in π and let $R_1, \ldots R_k$ be the sequence of FFCR used to infer C. We say that C is inferred by ***RK***$_n$ *weakening* (or just weakening) if either:

- $R_k = \Gamma(C_1) \Rightarrow C$, the first FFCR is $R_1 = \Sigma(0, A) \to 0$, where A is an **RK**$_n$ clause (i.e. R_1 is an instance of **A2**) and $R_2 \neq \Sigma(\Box_a 0, \Diamond_a(A \wedge E)) \to \Diamond_a(A \wedge E \wedge 0)$ (i.e. R_2 is obtained

using any other RFCR than the $\Sigma\,\Box\Diamond_a$-rule),

- Or, $R_k = \Sigma(C_1, C_2) \Rightarrow C$, R_1 is an instance of **A2** and no R_i is obtained using the Σ $\Box\Diamond_a$-rule.

Otherwise, we say that C is inferred by ***RK***$_n$ *resolution* (or just resolution).

Suppose some $\mathbf{RK}_n$ clause C is inferred by weakening and the first FFCR used to compute C is of the form $\Sigma(0, A) \rightarrow 0$, where A is some $\mathbf{RK}_n$ clause. Then if the final FFCR used to infer C is of the form $\Sigma(C_1, C_2) \Rightarrow C$ we say that C is inferred by weakening C_1 by C_2 less A. Whereas if the final FFCR used to compute C is of the form $\Gamma(C_1) \Rightarrow C$ then we say that C is inferred by weakening C_1 by itself less A.

Examples 5.1.1, 5.1.2, 5.1.3 and 5.1.4 are all resolution inferences. Hence we will now give two further examples of $\mathbf{RK}_n$ inferences that use weakening.

Example 5.1.5. Let $C_1 = \Diamond_a \neg l_1$ and $C_2 = \Box_a(l_1 \vee l_2)$. We can infer $C = \Diamond_a(\neg l_1 \wedge (\neg l_1 \vee l_2))$ by weakening C_1 by C_2 less l_1 as follows:

$$\begin{aligned}
&\Sigma(0,\ l_1) \rightarrow 0, && \textbf{A2},\\
&\Sigma(\neg l_1,\ l_1 \vee l_2) \rightarrow 0 \vee \neg l_1 \vee l_2, && \Sigma\ \vee\text{-rule},\\
&\Sigma(\Diamond_a \neg l_1,\ \Box_a(l_1 \vee l_2)) \rightarrow \Diamond_a(\neg l_1 \wedge (0 \vee \neg l_1 \vee l_2)), && \Sigma\ \Diamond\Box_a\text{-rule},\\
&\Sigma(\Diamond_a \neg l_1,\ \Box_a(l_1 \vee l_2)) \Rightarrow \Diamond_a(\neg l_1 \wedge (\neg l_1 \vee l_2)), && \textbf{S2},\\
&\Diamond_a(\neg l_1 \wedge (\neg l_1 \vee l_2)), && \textbf{R2 on } C_1,\ C_2.
\end{aligned}$$

Example 5.1.6. Let $C_1 = \Diamond_a((l_1 \vee l_2) \wedge l_3)$. We can infer $C = \Diamond_a((l_1 \vee l_2) \wedge l_3 \wedge (l_2 \vee l_3))$ from C_1 by weakening it by itself less l_1 as follows:

$$\begin{aligned}
&\Sigma(0,\ l_1) \rightarrow 0, && \textbf{A2},\\
&\Sigma(l_3,\ l_1 \vee l_2) \rightarrow 0 \vee l_2 \vee l_3, && \Sigma\ \vee\text{-rule},\\
&\Gamma(\Diamond_a((l_1 \vee l_2) \wedge l_3) \rightarrow \Diamond_a((l_1 \vee l_2) \wedge l_3 \wedge (0 \vee l_2 \vee l_3)), && \Gamma\ \Diamond_a 1\text{-rule},\\
&\Gamma(\Diamond_a((l_1 \vee l_2) \wedge l_3) \Rightarrow \Diamond_a((l_1 \vee l_2) \wedge l_3 \wedge (l_2 \vee l_3)), && \textbf{S2},\\
&\Diamond_a((l_1 \vee l_2) \wedge l_3 \wedge (l_2 \vee l_3)), && \textbf{R1 on } C_1.
\end{aligned}$$

Whilst it follows immediately from the strong soundness of $\mathbf{RK}_n$ that any $\mathbf{RK}_n$ clause C which is inferred from two $\mathbf{RK}_n$ clauses C_1 and C_2 is satisfied by a model (M, w) only if $(M, w) \models C_1 \wedge C_2$, for any $\mathbf{RK}_n$ clause which is inferred by weakening the following stronger statement holds.

Proposition 5.1.1. Let C_1 and C_2 be $\mathbf{RK}_n$ clauses, let $M = (W, R_1, \ldots, R_n, V)$ be a model and let $w \in W$ be such that $(M, w) \models C_1$. If some $\mathbf{RK}_n$ clause C is inferred by weakening C_1 by C_2 less some subclause A then $(M, w) \models C$.

Proof. Suppose $C_1 = C_2$. Then the proposition follows trivially from the strong soundness of **RK**$_n$.

Hence we assume that $C_1 \neq C_2$. The proof is by induction on the difference between the modal depth of C_2 and the modal depth of A. If the modal depth of C_2 is equal to that of A then C must be of the form $C_1 \vee A'$ where A' is an **RK**$_n$ clause such that $A \vee A' = C_2$. Hence $(M, w) \models C$.

Suppose A has modal depth strictly less than that of C_2. Then the sequence of FFCRs used to compute C must contain at least one FFCR to which some RFCR that adds a modal operator to the assumed clauses is applied.

As C is inferred from two **RK**$_n$ clauses the last such FFCR must be a Σ FFCR and so is of the form $\Sigma(C'_1, C'_2) \rightarrow C'$, where C'_1, C'_2 and C' are subclauses of C_1, C_2 and C respectively. Further C'_2 must have modal depth strictly less than that of C_2. Clearly we can weaken the normal form of C'_1 by the normal form of C'_2 less A to infer the normal form of C'. As every **RK**$_n$ clause is logically equivalent to its normal form it follows by the inductive hypothesis that if there exists some model $M' = (W', R'_1, \ldots, R'_n, V')$ and some $w' \in W'$ such that $(M', w') \models C'_1$ then $(M', w') \models C'$.

Let $\Sigma(C''_1, C''_2) \rightarrow C''$ be the FFCR obtained from $\Sigma(C'_1, C'_2) \rightarrow C'$. Further suppose there exists some model $M = (W, R_1, \ldots, R_n, V)$ and some world $w \in W$ such that $(M, w) \models C''_1$. Then by assumption either:

1. The FFCR $\Sigma(C''_1, C''_2) \rightarrow C''$ was obtained by applying the $\Sigma\,\Box\Box_a$-rule to $\Sigma(C'_1, C'_2) \rightarrow C'$. In which case $C''_1 = \Box_a C'_1$ and $C'' = \Box_a C'$. As $(M, w) \models C''_1$ it follows that either W contains no worlds which are a-accessible from w, or W contains at least one world that is a-accessible from w and every $w_1 \in W$ that is a-accessible from w is such that $(M, w_1) \models C'_1$. In the former case we trivially have $(M, w) \models C''$. In the latter case we recall that every (M, w_1) which satisfies C'_1 also satisfies C' and so $(M, w) \models C''$.

2. The FFCR $\Sigma(C''_1, C''_2) \rightarrow C''$ is obtained by applying the $\Sigma\,\Diamond\Box_a$-rule to $\Sigma(C'_1, C'_2) \rightarrow C'$. Then $C''_1 = \Diamond_a(C'_1 \wedge E)$ and $C'' = \Diamond_a(C'_1 \wedge C' \wedge E)$, where E is an **RK**$_n$ CNF. As $(M, w) \models C''_1$ it follows that there exists some $w_1 \in W$ such that w_1 is a-reachable from w and $(M, w_1) \models (C'_1 \wedge E)$. Recall that, by the inductive hypothesis we have that if some world in some model satisfies C'_1 then it must also satisfy C'. Hence $(M, w_1) \models C'$ and so $(M, w) \models C''$.

By assumption $\Sigma(C''_1, C''_2) \rightarrow C''$ is the last FFCR obtained using some modal RFCR, hence $\Sigma(C''_1, C''_2) \rightarrow C''$ must either be the final FFCR used to compute C which is obtained using a non-simplification RFCR, or the second to last FFCR used to compute C which is obtained using a non-simplification RFCR. Further if $\Sigma(C''_1, C''_2) \rightarrow C''$ is not the final such FFCR it must be followed by FFCR obtained using the $\Sigma\,\vee$-rule. In the former case, as every **RK**$_n$ clauses is logically equivalent to its normal form it follows that if $(M, w) \models C_1$ then $(M, w) \models C$. In the latter case we observe that the FFCR obtained using the $\Sigma\,\vee$-rule must be of the form $\Sigma(C''_1 \vee B_1, C''_2 \vee B_2) \rightarrow C'' \vee B$, where B_1, B_2 and B are **RK**$_n$ clauses. Clearly any world in any model which satisfies C''_1 (respectively C'') also satisfies $C''_1 \vee B_1$ (respectively $C'' \vee B$). Hence, as

C_1 and C are the normal forms of $C_1'' \vee B_1$ and $C'' \vee B$ respectively, it follows that if $(M, w) \models C_1$ then $(M, w) \models C$. □

5.2 Showing that $\mathbf{RK}_n$ p-simulates the family of K-Res proof systems

In this section we show that the proof system $\mathbf{RK}_n$ p-simulates each of the **K**-Res resolution systems. As each of the **K**-Res systems are p-equivalent to one another (Theorem 4.5.1) it is sufficient to show that any one of the **K**-Res systems is p-simulated by $\mathbf{RK}_n$.

Given any set of SNF^+_{mp} clauses we can easily construct an equivalent set of $\mathbf{RK}_n$ clauses by replacing every modal clause of the form $\Box_\mu(l \to \circ_a l')$ with the $\mathbf{RK}_n$ clause $\Box_\mu(\neg l \vee \circ_a l')$. Hence we choose to directly compare $\mathbf{RK}_n$ with $\mathbf{K}^+_{mp}$-Res.

Theorem 5.2.1. $\mathbf{K}^+_{mp}$-Res $\leq_p$ $\mathbf{RK}_n$.

Proof. Let $\mathcal{C}$ be an unsatisfiable set of SNF^+_{mp} clauses and let $\mathcal{C}'$ be the corresponding set of $\mathbf{RK}_n$ clauses. Further let $\pi = C_1, \ldots, C_k$ be a $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}$. We show that given π we can construct an $\mathbf{RK}_n$ refutation $\pi' = \pi'_1, \ldots, \pi'_k$ of $\mathcal{C}'$, where each π'_i is an $\mathbf{RK}_n$ derivation of either $C'_i = C_i$ if C_i is a literal clause, or $C'_i = \Box_\mu(\neg x_1 \vee \circ_a x_2)$ if C_i is a modal clause of the form $\Box_\mu(x_1 \to \circ_a x_2)$. Further each π'_i has size polynomial in that of C_i and so the size of π' is polynomial in that of π.

The construction of π'_i depends on how the clause C_i was inferred in π. For each i such that $C_i \in \mathcal{C}$ we simply let $\pi'_i = C'_i$.

(LRES) Suppose $C_i = \Box_\mu(D_1 \vee D_2)$, was inferred by applying LRES to two clauses:

$$C_{i_1} = \Box_\mu(D_1 \vee l) \text{ and } C_{i_2} = \Box_\mu(D_2 \vee \neg l).$$

Then we can infer $C'_i = C_i$ from $C'_{i_1} = C_{i_1}$ and $C'_{i_2} = C_{i_2}$ using $\mathbf{RK}_n$ as follows:

$$\begin{array}{ll}
\Sigma\,(l,\ \neg l) \to 0, & \textbf{A1}, \\
\Sigma\,(D_1 \vee l,\ D_2 \vee \neg l) \to D_1 \vee D_2 \vee 0, & \Sigma \vee\text{-rule}, \\
\Sigma\,(\Box_\mu\,(D_1 \vee l)\,,\ \Box_\mu\,(D_2 \vee \neg l)) \to \Box_\mu\,(D_1 \vee D_2 \vee 0)\,, & \Sigma\Box\Box_\mu\text{-rules}, \\
\Sigma\,(\Box_\mu\,(D_1 \vee l)\,,\ \Box_\mu\,(D_2 \vee \neg l)) \Rightarrow \Box_\mu\,(D_1 \vee D_2)\,, & \textbf{S2}, \\
\Box_\mu\,(D_1 \vee D_2)\,, & \textbf{R2 on } C'_{i_1},\ C'_{i_2},
\end{array}$$

where for any $\mu = a_1 \ldots a_m \in \mathcal{A}^*$ the $\Sigma\,\Box\Box_\mu$-rules abbreviates applying the Σ rules $\Box\Box_{a_1}, \ldots, \Box\Box_{a_m}$ successively. Hence we let π'_i denote the above derivation. Clearly $|\pi'_i|$ is linear in $|C'_i| = |C_i|$.

(GEN1): Suppose C_i is inferred by applying GEN1 to some set of $z + 2$ clauses. Then C_i must be of the form:

$$\Box_\mu \left(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l' \right),$$

and be inferred by applying GEN1 to z positive modal clauses $C_{i_1}, \ldots, C_{i_z}$ of the form:

$$C_{i_j} = \Box_\mu \left(l'_j \to \Box_a l_j \right),$$

for each $j \in [z]$, a negative modal clause:

$$C_{i_{z+1}} = \Box_\mu \left(l' \to \Diamond_a l \right)$$

and a literal clause:

$$C_{i_{z+2}} = \Box_\mu a \left(\bigvee_{x=1}^{z} \neg l_x \vee \neg l \right),$$

where $i_1, \ldots, i_{z+2} < i$. By assumption $C'_{i_j} = \Box_\mu \left(\neg l'_j \vee \Box_a l_j \right)$ for each $j \in [z]$, $C_{i_{z+1}} = \Box_\mu \left(\neg l' \vee \Diamond_a l \right)$ and $C'_{i_{z+2}} = C_{i_{z+2}}$.

To derive $C'_i = C_i$ from $C'_{i_1}, \ldots, C'_{i_{z+2}}$ using $\mathbf{RK}_n$ we proceed as follows. First for each $k \in [z]$ we successively apply the steps below:

$$\Sigma(l_k,\ \neg l_k) \to 0, \qquad \textbf{A1},$$

$$\Sigma\left(l_k,\ \textstyle\bigvee_{x=k}^{z} \neg l_x \vee \neg l\right) \to \bigvee_{x=k+1}^{z} \neg l_x \vee \neg l \vee 0, \qquad \Sigma \vee\text{-rule},$$

$$\Sigma\left(\Box_a l_k,\ \Box_a \left(\textstyle\bigvee_{x=k}^{z} \neg l_x \vee \neg l\right)\right) \to \Box_a \left(\bigvee_{x=k+1}^{z} \neg l_x \vee \neg l \vee 0\right), \qquad \Sigma\Box\Box_a\text{-rule},$$

$$\Sigma\left(\neg l'_k \vee \Box_a l_k,\ \Box_a \left(\textstyle\bigvee_{x=k}^{z} \neg l_x \vee \neg l\right) \vee \bigvee_{x=1}^{k-1} \neg l'_x\right) \to \qquad \Sigma \vee\text{-rule},$$
$$\Box_a \left(\textstyle\bigvee_{x=k+1}^{z} \neg l_x \vee \neg l \vee 0\right) \vee \bigvee_{x=1}^{k} \neg l'_x,$$

$$\Sigma\left(\Box_\mu \left(\neg l'_k \vee \Box_a l_k\right),\ \Box_\mu \left(\Box_a \left(\textstyle\bigvee_{x=k}^{z} \neg l_x \vee \neg l\right) \vee \bigvee_{x=1}^{k-1} \neg l'_x\right)\right) \to \qquad \Sigma\Box\Box_\mu\text{-rules},$$
$$\Box_\mu \left(\Box_a \left(\textstyle\bigvee_{x=k+1}^{z} \neg l_x \vee \neg l \vee 0\right) \vee \bigvee_{x=1}^{k} \neg l'_x\right),$$

$$\Sigma\left(\Box_\mu \left(\neg l'_k \vee \Box_a l_k\right),\ \Box_\mu \left(\Box_a \left(\textstyle\bigvee_{x=k}^{z} \neg l_x \vee \neg l\right) \vee \bigvee_{x=1}^{k-1} \neg l'_x\right)\right) \Rightarrow \qquad \textbf{S2},$$
$$\Box_\mu \left(\Box_a \left(\textstyle\bigvee_{x=k+1}^{z} \neg l_x \vee \neg l\right) \vee \bigvee_{x=1}^{k} \neg l'_x\right),$$

$$A_k = \Box_\mu \left(\Box_a \left(\textstyle\bigvee_{x=k+1}^{z} \neg l_x \vee \neg l\right) \vee \bigvee_{x=1}^{k} \neg l'_x\right), \qquad \textbf{R2 on } \star.$$

where if $k = 1$ then $\star$ is C_{i_k} and $C_{i_{z+2}}$ and if $k > 1$ then $\star$ is C_{i_k} and A_{k-1}. Clearly each of these z inferences has size linear in $|C'_i| = |C_i|$. Further note that the clause:

$$A_z = \Box_\mu \left(\Box_a \neg l \vee \bigvee_{x=1}^{z} \neg l'_x \right).$$

Hence we complete our $\mathbf{RK}_n$ derivation of C'_i by resolving A_z with $C'_{i_{z+1}}$ as follows:

$\Sigma\,(l,\ \neg l) \to 0$, **A1**,

$\Sigma\,(\Diamond_a l,\ \Box_a \neg l) \to \Diamond_a\,(l \wedge 0)$, Σ $\Diamond\Box_a$-rule,

$\Sigma\left(\neg l' \vee \Diamond_a l,\ \Box_a \neg l \vee \bigvee_{x=1}^{z} \neg l'_x\right) \to \Diamond_a\,(l \wedge 0) \vee \bigvee_{x=1}^{z} \neg l'_x \vee \neg l'$, Σ $\vee$-rule,

$\Sigma\left(\Box_\mu\left(\neg l' \vee \Diamond_a l\right),\ \Box_\mu\left(\Box_a \neg l \vee \bigvee_{x=1}^{z} \neg l'_x\right)\right) \to \Box_\mu\left(\Diamond_a\,(l \wedge 0) \vee \bigvee_{x=1}^{z} \neg l'_x \vee \neg l'\right)$, $\Sigma\,\Box\Box_\mu$-rules,

$\Sigma\left(\Box_\mu\left(\neg l' \vee \Diamond_a l\right),\ \Box_\mu\left(\Box_a \neg l \vee \bigvee_{x=1}^{z} \neg l'_x\right)\right) \Rightarrow \Box_\mu\left(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l'\right)$, **S3**, **S1** and **S2**,

$\Box_\mu\left(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l'\right)$, **R2** on $C_{i_{z+1}}$, A_z.

Clearly the size of this final inference is also linear in $|C'_i|$. Hence if we take π'_i to be the above derivation then as π'_i consists of $z+1$ inferences of size linear in $|C_i|$ it follow that π'_i has size polynomial in $|C_i|$.

(GEN3): Suppose C_i is inferred by applying GEN3 to some set of $z+2$ clauses. Then C_i must be of the form:

$$\Box_\mu\left(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l'\right)$$

and be inferred by applying GEN3 to z positive modal clauses $C_{i_1}, \ldots, C_{i_z}$ of the form:

$$C_{i_j} = \Box_\mu\left(l'_j \to \Box_a \neg l_j\right)$$

for each $j \in [z]$, a negative modal clause:

$$C_{i_{z+1}} = \Box_\mu\left(l' \to \Diamond_a l\right)$$

and a literal clause:

$$C_{i_{z+2}} = \Box_{\mu a}\left(\bigvee_{x=1}^{z} l_x\right),$$

where $i_1, \ldots, i_{z+2} < i$. Further by assumption, $C'_{i_j} = \Box_\mu\left(\neg l'_j \vee \Box_a l_j\right)$ for each $j \in [z]$, $C_{i_{z+1}} = \Box_\mu\left(\neg l' \vee \Diamond_a l\right)$ and $C'_{i_{z+2}} = C_{i_{z+2}}$.

For each $k \in [z]$ we construct an inference of some A_k by following exactly the same method as for GEN1. The final $\mathbf{RK}_n$ clause A_z which we obtain is of the form:

$$\Box_\mu\left(\Box_a 0 \vee \bigvee_{x=1}^{z} \neg l'_x\right).$$

As in the case for GEN1 each of these z inferences has size linear in $|C'_i|$ and so $|C_i|$ as $C'_i = C_i$.

We then proceed to infer C'_i from $C_{i_{z+1}}$ and A_z as follows:

$\Sigma\,(0,\ \ l) \to 0,$ **A2**,

$\Sigma\,(\Box_a 0,\ \ \Diamond_a l) \to \Diamond_a\,(l \wedge 0)\,,$ Σ $\Box\Diamond_a$-rule,

$\Sigma\,(\Box_a 0 \vee \bigvee_{x=1}^{z} \neg l'_x,\ \ \neg l' \vee \Diamond_a l) \to \Diamond_a\,(l \wedge 0) \vee \bigvee_{x=1}^{z} \neg l'_x \vee \neg l',$ Σ $\vee$-rule,

$\Sigma\,\big(\Box_\mu\,(\Box_a 0 \vee \bigvee_{x=1}^{z} \neg l'_x)\,,\ \ \Box_\mu\,\big(\neg l' \vee \Diamond_a l\big)\big) \to$ Σ $\Box\Box_\mu$-rules,
$\Box_\mu\,(\Diamond_a\,(l \wedge 0) \vee \bigvee_{x=1}^{z} \neg l'_x \vee \neg l')\,,$

$\Sigma\,\big(\Box_\mu\,(\Box_a 0 \vee \bigvee_{x=1}^{z} \neg l'_x)\,,\ \ \Box_\mu\,\big(\neg l' \vee \Diamond_a l\big)\big) \Rightarrow \Box_\mu\,(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l')\,,$ **S3**, **S1** and **S2**,

$\Box_\mu\,(\bigvee_{x=1}^{z} \neg l'_x \vee \neg l')\,,$ **R2** on A_z, $C_{i_{z+1}}$.

Clearly the size of this final inference is linear in $|C'_i|$. Hence if we take π'_i to be the whole derivation then $|\pi'_i|$ is polynomial in $|C_i|$.

□

Corollary 5.2.1. The proof system **RK**$_n$ p-simulates each of the **K**-Res systems defined in Chapter 4.

Proof. This follows immediately from Theorem 5.2.1 and the p-equivalence of the family of **K**-Res systems (Theorems 4.5.1 and 4.6.3). □

Chapter 6

Feasible interpolation for modal resolution systems

In this chapter we present a lower bound technique for the family of **K**-Res modal resolution systems defined in Chapter 4. This technique is a modified version of the successful propositional lower bound technique, feasible interpolation [53, 75].

Feasible interpolation (Definition 6.1.1) has been used to show lower bounds for propositional resolution and the propositional proof system cutting planes [75]. The technique is based on Craig's interpolation theorem [29]. We say a logic L has Craig interpolation (or just interpolation) if given any true implication $A \to B$ (or equivalently a false conjunction $A \wedge \neg B$) there exists a formula C over the shared variables of A and B such that $A \to C$ and $C \to B$ are both true. In particular we call C an *interpolant* of A and B.

Informally, we say a proof system P has feasible interpolation if given any proof of any formula of the form $A \to B$ we can construct a circuit C which interpolates (i.e. is an interpolant of) A and B and has size polynomial in the size of the proof of $A \to B$. Clearly if a proof system has feasible interpolation and there exists a family of formulas for which there cannot exist small interpolating circuits, then every proof of these formulas must be large. Hence feasible interpolation allows us to obtain proof size lower bounds indirectly via circuit lower bounds.

A *circuit* is a finite directed acyclic graph (dag) where each leaf vertex is labelled by either a propositional variable or a constant ($\mathbf{0}$ or $\mathbf{1}$). Further each internal vertex is labelled by some *gate*. Each of the gates must be an element of some basis set B of Boolean functions (i.e. functions from $\{0,1\}^m$ to $\{0,1\}$ for some $m \in \mathbb{N}$). If not specified we take $B = \{\neg, \vee, \wedge, \mathbf{0}, \mathbf{1}\}$. Finally, each vertex with no out edges is called an output vertex. A circuit with n input vertices and m output vertices calculates a function $f : \{0,1\}^n \to \{0,1\}^m$. The *size* of a circuit is the number of vertices

it contains.

We further note that any circuit where each vertex has at most 1 out edge is in fact a propositional formula.

A number of modal logics, including the basic monomodal logic **K**, have been shown to posses the interpolation property [37, 77]. Further in [52] it was shown that the interpolation property transfers from the monomodal logic **K** to the multimodal logic $\mathbf{K}_n$.

In this chapter we address the natural question of whether or not some variant of feasible interpolation can be used to obtain lower bounds for modal resolution systems. In [45, 46] Hrubeš gave a positive answer to the analogous question regarding Frege systems for certain modal logics (including **K**). Further, in [49] Jeřábek extended Hrubeš' lower bound (and lower bound proving technique) to extended Frege systems for certain modal logics with infinite branching. In [18] it was shown that the sequent calculus for several normal modal logics, including **K**, also has a (weaker) type of feasible interpolation. An exponential lower bound for these proof systems was further shown under certain cryptographic assumptions.

6.1 Feasible interpolation for propositional logic

Definition 6.1.1 ([75])**.** Let $\bar{p}, \bar{q}$ and $\bar{r}$ be disjoint sets of propositional variables, and let $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ be propositional CNFs over the variables in $\bar{p}, \bar{q}$ and $\bar{p}, \bar{r}$ respectively. We say a circuit $C(\bar{p})$ *interpolates* the formula $A(\bar{p}, \bar{q}) \wedge B(\bar{p}, \bar{r})$ if given any $\{0, 1\}$ assignment $\bar{\alpha}$ to the variables in $\bar{p}$ we have:

$$C(\bar{\alpha}) = \begin{cases} 0 & \text{only if } A(\bar{\alpha}, \bar{q}) \text{ is unsatisfiable,} \\ 1 & \text{only if } B(\bar{\alpha}, \bar{r}) \text{ is unsatisfiable.} \end{cases}$$

Let Q be a propositional proof system. We say Q has *feasible interpolation* if given any unsatisfiable formula $A(\bar{p}, \bar{q}) \wedge B(\bar{p}, \bar{r})$ of the above form, and any Q refutation π of this formula we can construct a circuit $C(\bar{p})$ which interpolates $A(\bar{p}, \bar{q}) \wedge B(\bar{p}, \bar{r})$ and has size polynomial in the length of π.

Further, if given that the $\bar{p}$ variables occur only positively in $A(\bar{p}, \bar{q})$, we can construct a circuit as above but with the additional requirement of having a monotone basis, $\{\vee, \wedge, \mathbf{0}, \mathbf{1}\}$, we say Q has *monotone feasible interpolation*.

It follows from the above definition that if a proof system Q has feasible interpolation then any family of formulas of the form $A(\bar{p}, \bar{q}) \wedge B(\bar{p}, \bar{r})$ that has only exponential sized interpolating circuits requires exponential sized Q refutations. Similarly if Q has monotone feasible interpolation and every monotone circuit that interpolates a family of formulas has exponential size then this family of formulas must require exponential size Q refutations.

There are currently no known superpolynomial circuit lower bounds, however there is a known superpolynomial monotone circuit lower bound [2, 78]. Hence in order to prove exponential proof size lower bounds we must prove that a proof system has not only feasible interpolation but also

monotone feasible interpolation. In [75] it was shown that propositional resolution has monotone feasible interpolation, leading to a new exponential lower bound for propositional resolution.

6.2 Feasible interpolation for modal resolution systems

In this section we prove that whilst the proof system $\mathbf{K}^+_{mp}$-Res does not admit the full version of feasible interpolation given in Definition 6.1.1, it does admit a weaker version of feasible interpolation. We begin by considering a simple example which rules out the possibility of $\mathbf{K}^+_{mp}$-Res admitting feasible interpolation.

Example 6.2.1. Consider the modal formula:

$$\phi = (\Box_a \Box_a l_1 \wedge \Diamond_a \Diamond_a \neg l_1) \vee (\Box_a \Box_a l_2 \wedge \Diamond_a \Diamond_a \neg l_2),$$

and the corresponding set of SNF^+_{mp} clauses:

$$\left\{ \begin{array}{c} x_0,\, \neg x_0 \vee x_1 \vee x_2,\; x_1 \to \Box_a x_3,\, \Box_a(x_3 \to \Box_a l_1), \\ x_1 \to \Diamond_a x_4,\, \Box_a(x_4 \to \Diamond_a x_5),\, \Box_{aa}(\neg x_5 \vee \neg l_1),\, x_2 \to \Box_a x_6, \\ \Box_a(x_6 \to \Box_a l_2),\, x_2 \to \Diamond_a x_7,\, \Box_a(x_7 \to \Diamond_a x_8),\, \Box_{aa}(\neg x_8 \vee \neg l_2) \end{array} \right\}$$

It is clear that ϕ is an unsatisfiable formula. Hence the above set of clauses must also be unsatisfiable and there must exist a $\mathbf{K}^+_{mp}$-Res refutation said set of clauses. Further:

$$\bar{p} = \{x_1,\, x_2,\, l_1,\, l_2\},\; \bar{q} = \{x_0,\, x_3,\, x_6\} \text{ and } \bar{r} = \{x_4,\, x_5,\, x_7,\, x_8\}$$

are disjoint sets of propositional variables. Now consider the assignment $\bar{\alpha}$ which maps $l_1 \mapsto 0,\; l_2 \mapsto 0,\; x_1 \mapsto 1,\; x_2 \mapsto 1$. Neither of the disjoint sets:

$$A(\bar{\alpha}, \bar{q}) = \{x_0,\; \neg x_0 \vee 1 \vee 1,\; 1 \to \Box_a x_3,\; \Box_a(x_3 \to \Box_a 0),\; 1 \to \Box_a x_6,\; \Box_a(x_6 \to \Box_a 0)\},$$

and:

$$B(\bar{\alpha}, \bar{r}) = \left\{ \begin{array}{c} 1 \to \Diamond_a x_4,\; \Box_a(x_4 \to \Diamond_a x_5),\; \Box_{aa}(\neg x_5 \vee 1), \\ 1 \to \Diamond_a x_7,\; \Box_a(x_7 \to \Diamond_a x_8),\; \Box_{aa}(\neg x_8 \vee 1) \end{array} \right\},$$

is unsatisfiable. To see this note that the model $M_1 = (W_1, R_1, V_1)$ where $W_1 = w_1$, $R_1 = \emptyset$ and $V_1(w_1)(x_0) = 1$ is such that $(M_1, w_1) \models A(\bar{\alpha}, \bar{q})$ and the model $M_2 = (W_2, R_2, V_2)$ where $W_2 = \{w_2, w_2', w_2''\}$, $R_2 = \{(w_2, w_2'), (w_2', w_2'')\}$ and $V_2(w_2')(x_4) = V_2(w_2')(x_7) = 1$ and $V_2(w_2'')(x_5) = V_2(w_2'')(x_8) = 1$ is such that $(M_2, w_2) \models B(\bar{\alpha}, \bar{r})$.

The above example demonstrates the existence of unsatisfiable sets of SNF^+_{mp} clauses which can be split into two disjoint subsets of clauses which are both satisfiable. Hence neither $\mathbf{K}^+_{mp}$-Res,

nor indeed any proof system that operates on formulas in SNF^+_{mp} (or similarly SNF, SNF_{ml} or SNF^+_{mc}) admits feasible interpolation. This is essentially due to the fact that $\Box_a 0$ is satisfiable.

In the following definition we introduced a weaker notion of feasible interpolation called *modal feasible interpolation*, which as we will see in Theorem 6.2.1, is admitted by $\mathbf{K}^+_{mp}$-Res. The idea behind this weaker form of interpolation is to add some clauses to the sets $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ so as to ensure that at least one of these sets is always unsatisfiable. We then say that a proof system admits modal feasible interpolation if given any refutation π of $A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})$ there exists a circuit that interpolates the newly constructed supersets of $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$, and has size polynomial in the length of π.

Definition 6.2.1. Let Q be a modal proof system. We say Q has *modal feasible interpolation* if given any Q refutation of some set of SNF^+_{mp} clauses $A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})$, where $\bar{p}$, $\bar{q}$ and $\bar{r}$ are disjoint sets of propositional variables, and $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ are disjoint sets of SNF^+_{mp} clauses, we can construct a circuit $C(\bar{p})$ such that the following holds:

1. $C(\bar{p})$ interpolates $A'(\bar{p}, \bar{q}) \supseteq A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r}) \supseteq B(\bar{p}, \bar{r})$ where:

$$\begin{aligned} A'(\bar{p}, \bar{q}) = A(\bar{p}, \bar{q}) \cup \{\Box_\mu(l \to \Diamond_a l') \mid \Box_\mu(l \to \Box_a l') \in A(\bar{p}, \bar{q})\} \\ \cup \{\Box_\mu \Diamond_a q' \mid \Box_\mu(l_2 \to \Diamond_a l) \in A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})\}, \\ B'(\bar{p}, \bar{r}) = B(\bar{p}, \bar{r}) \cup \{\Box_\mu(l \to \Diamond_a l') \mid \Box_\mu(l \to \Box_a l') \in B(\bar{p}, \bar{r})\} \\ \cup \{\Box_\mu \Diamond_a r' \mid \Box_\mu(l_2 \to \Diamond_a l) \in A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})\}, \end{aligned}$$

   further, q is a new $\bar{q}$ variable and r is a new $\bar{r}$ variable.

2. $C(\bar{p})$ has size polynomial in the length of the original Q refutation of $A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})$.

Further, if given that the $\bar{p}$ variables occur only positively in $A(\bar{p}, \bar{q})$ we can construct a circuit as above but with the additional requirement of having a monotone basis, $\{\vee, \wedge, \mathbf{0}, \mathbf{1}\}$, we say Q has *modal monotone feasible interpolation*.

We will now prove that $\mathbf{K}^+_{mp}$-Res has modal feasible interpolation. The proof of this is similar to the proof that propositional resolution has feasible interpolation given by Pudlák in [75].

The first half of the proof consists of showing that any $\mathbf{K}^+_{mp}$-Res refutation π of two disjoint sets of SNF^+_{mp} clauses $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ can be transformed into either a refutation of $A'(\bar{\alpha}, \bar{q})$ or a $\mathbf{K}^+_{mp}$-Res refutation of $B'(\bar{\alpha}, \bar{r})$. As every clause with mixed variables (i.e. $\bar{q}$ and $\bar{r}$ variables) must have been inferred by some resolution step, this essentially just consists of replacing steps where a $\bar{p}$ variable is resolved on with some clause which does not contain that variable. This can always be done by as we have replaced all $\bar{p}$ variables with either 0 or 1.

The second half of the proof consists of showing that from this refutation we can construct a circuit $C(\bar{p})$ which interpolates $A'(\bar{p}, \bar{q})$ and $B'(\bar{p}, \bar{r})$. Crucially the size of this circuit is equal to the length of the original refutation π.

Theorem 6.2.1. $\mathbf{K}^+_{mp}$-Res has modal feasible interpolation.

Proof. Let $A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})$ be an unsatisfiable set of SNF^+_{mp} clauses, where $\bar{p}$, $\bar{q}$ and $\bar{r}$ are disjoint sets of propositional variables. Further let π be a $\mathbf{K}^+_{mp}$-Res refutation of this set of clauses. Given an assignment $\bar{\alpha}$ to the variables in $\bar{p}$ we will show that we can construct a $\mathbf{K}^+_{mp}$-Res refutation of either the set of clauses $A'(\bar{\alpha}, \bar{q})$ or the set of clauses $B'(\bar{\alpha}, \bar{r})$, both defined as in Definition 6.2.1.

Note that given any $\mathbf{K}^+_{mp}$-Res refutation in which the same initial clause is taken as a premise for several different inference steps, if we require that a new copy of every such initial clause is used for each inference then this causes at most a linear increase in the size of the refutation. Hence we assume without loss of generality that each appearance of an initial clause in π is used in at most one inference.

We call a clause in π a *q-clause* (respectively an *r-clause*) if it contains only $\bar{p}$ and $\bar{q}$ variables (respectively $\bar{p}$ and $\bar{r}$ variables). If a clause contains only $\bar{p}$ variables or is empty we call it a q-clause (respectively an r-clause) if it was derived from q-clauses (respectively r-clauses).

To construct our new refutation we transform each clause in π by replacing it with some subclause of itself. Recall that every modal clause in π must be initial, whereas literal clauses may be either initial or non-initial. If a literal clause is initial then the transformation leaves it as it is. If a literal clause is non-initial then the transformation depends on which rule of $\mathbf{K}^+_{mp}$-Res was used to infer it. The transformation of modal clauses depends on which clauses they were used to infer.

Suppose π contains a literal clause which is inferred by an instance of LRES.

$$\dfrac{\begin{array}{ll}\Box_\mu(D \vee l) & (L_1)\\ \Box_\mu(D' \vee \neg l) & (L_2)\end{array}}{\Box_\mu(D \vee D') \quad (L_3)}$$

If L_1 is a non-initial clause and has already been transformed then we let $\Box_\mu E$ be the clause that L_1 was replaced with, otherwise we let $\Box_\mu E = \Box_\mu(D \vee l)$. We similarly define $\Box_\mu E'$ in terms of L_2. Suppose $l \in \bar{p}$. If $l \mapsto 0$ then we replace L_3 with $\Box_\mu E$. If $l \mapsto 1$ then we replace L_3 with $\Box_\mu E'$. Suppose $l \in \bar{q}$ (respectively $l \in \bar{r}$), if one of E and E' is an r-clause (respectively a q-clause) then by definition it cannot contain l and so we replace L_3 with with either $\Box_\mu E$ or $\Box_\mu E'$ accordingly. Otherwise we either use LRES to resolve l in $\Box_\mu E$ with $\neg l$ in $\Box_\mu E'$ and replace L_3 with the result, or if one of the literal clauses $\Box_\mu E$ and $\Box_\mu E'$ does not contain l then we replace L_3 with this clause.

Suppose π contains a literal clause which is inferred by an instance of GEN1.

$$\dfrac{\begin{array}{ll}\Box_\mu(l'_1 \to \Box_a l_1) & (L_1)\\ \vdots & \vdots\\ \Box_\mu(l'_z \to \Box_a l_z) & (L_z)\\ \Box_\mu(l'_{z+1} \to \Diamond_a l_{z+1}) & (L_{z+1})\\ \Box_{\mu a}\left(\bigvee_{i=1}^{z+1} \neg l_i\right) & (L_{z+2})\end{array}}{\Box_\mu\left(\bigvee_{i=1}^{z+1} \neg l'_i\right) \quad (L_{z+3})}$$

If the clause L_{z+2} has previously been transformed then we let $\Box_{\mu a} E$ be the clause it was replaced with. If L_{z+2} has not been transformed then we let $\Box_{\mu a} E = L_{z+2}$. Note that in either case the set

of literals in $\Box_{\mu a}E$ must be a subset of the literals in L_{z+2}.

Suppose $\Box_{\mu a}E$ is a q-clause, then each variable in $\Box_{\mu a}E$ (i.e. each l_i where $i \in [z+1]$) must be either a $\bar{p}$ variable or a $\bar{q}$ variable. If for some $i \in [z+1]$ the variable l_i is mapped to 0 by $\bar{\alpha}$ and the corresponding modal clause L_i is an r-clause then we replace L_{z+3} with the clause $\Box_\mu(l'_i \to \Diamond_a l_i) \in B'(\bar{p}, \bar{r})$. Otherwise either some variable l_i is mapped to 0 by $\bar{\alpha}$ and the corresponding modal clause L_i is a q-clause, or no variable l_i is mapped to 0. In the former case we replace L_{z+3} with the clause $\Box_\mu(l'_i \to \Diamond_a l_i) \in A'(\bar{p}, \bar{q})$. In the latter case either:

1. Every variable in $\Box_{\mu a}E$ is in $\bar{p}$. By assunption this is the case only if either E is empty or every literal in $\Box_{\mu a}E$ is mapped to 1 under the assignment $\bar{\alpha}$. Hence in this case we replace L_{z+1} with $\Box_\mu \Diamond_a q' \in A'(\bar{p}, \bar{q})$ and L_{z+3} with the clause $\Box_\mu 0$. Note that this clause can be inferred by applying GEN3 to $\Box_{\mu a}E \equiv \Box_{\mu a}0$ and $\Box_\mu \Diamond_a q'$.

2. Or, there exists some $i \in [z+1]$ such that l_i is in $\bar{q}$ and $\neg l_i$ appears in $\Box_{\mu a}E$. If l_{z+1} is such a variable then we replace L_{z+3} with the clause obtained by applying GEN1 to $\Box_\mu(l'_{z+1} \to \Diamond_a l_{z+1})$, the set of modal clauses $\{L_j \mid j \in [z],\ \neg l_j \in E \text{ and } l_j \in \bar{q}\}$ and $\Box_{\mu a}E$. Otherwise we replace L_i with the clause $\Box_\mu(l'_i \to \Diamond_a l_i)$ and replace L_{z+3} with the clause obtained by applying GEN1 to $\Box_\mu(l'_i \to \Diamond_a l_i)$, the set of modal clauses $\{L_j \mid j \in [z+1] \setminus \{i\},\ \neg l_j \in E \text{ and } l_j \in \bar{q}\}$ and $\Box_{\mu a}E$.

If $\Box_{\mu a}E$ is an r-clause then the transformation is dual.

Suppose π contains a literal clause which is inferred by an application of GEN3.

$$\dfrac{\begin{array}{cc} \Box_\mu(l'_1 \to \Box_a l_1) & (L_1) \\ \vdots & \vdots \\ \Box_\mu(l'_z \to \Box_a l_z) & (L_z) \\ \Box_\mu(l'_{z+1} \to \Diamond_a l_{z+1}) & (L_{z+1}) \\ \Box_{\mu a}\left(\bigvee_{i=1}^{z+1} \neg l_i\right) & (L_{z+2}) \end{array}}{\Box_\mu\left(\bigvee_{i=1}^{z} \neg l'_i\right) \quad (L_{z+3})}$$

The transformation is identical to that of GEN1.

As every clause in π is replaced by a subclause of itself it follows that our transformed sequence of clauses ends with the empty clause. Further, if we replace every $\bar{p}$ variable with its value under the assignment $\bar{\alpha}$ then our transformed sequence of clauses becomes a $\mathbf{K}^+_{mp}$-Res refutation of $A'(\bar{\alpha}, \bar{q}) \cup B'(\bar{\alpha}, \bar{r})$. Our transformation is such that q-clauses can only be inferred from sets of q-clauses and r-clauses can only be inferred from sets of r-clauses. Hence if the final clause in our transformed proof is a q-clause then our new refutation must contain be a subrefutation of $A'(\bar{\alpha}, \bar{q})$, similarly if the final clause is an r-clause then the refutation must contain a subrefutation of $B'(\bar{\alpha}, \bar{r})$.

Now we construct a circuit $C(\bar{p})$ such that $C(\bar{\alpha}) = 0$ only if the set of clauses $A'(\bar{\alpha}, \bar{q})$ is unsatisfiable and $C(\bar{\alpha}) = 1$ only if the set of clauses $B'(\bar{\alpha}, \bar{q})$ is unsatisfiable. More specifically, we construct the circuit so that the value computed by the gate labelling each vertex is 0 if and

only if the corresponding clause becomes a q-clause in the transformed proof. Similarly the value computed by the gate labelling each vertex is 1 if and only if the corresponding clause becomes a r-clause in the transformed proof.

We take $C(\bar{p})$ to have the same underlying structure as π. Hence the initial clauses correspond to the leaf vertices of the circuit and the internal vertices correspond to clauses inferred by the applying some rule of $\mathbf{K}^+_{mp}$-Res to clauses represented by its parent vertices. We label leaf vertices corresponding to q-clauses with the constant gate **0** and leaf vertices corresponding to r-clauses with the constant gate **1**. The gate we put on each internal vertex depends on the rule used to infer the corresponding clause.

Suppose a vertex u of $C(\bar{p})$ corresponds to a clause L_3 in π which is inferred by applying LRES to two clauses L_1 and L_2 which correspond to the vertices u_1 and u_2 respectively. Let l be the variable that was resolved on. If $l \in \bar{q}$ then we put a $\vee$ gate on u. If $l \in \bar{r}$ then we put a $\wedge$ gate on u. If $l \in \bar{p}$ we put a selector gate on u, defined:

$$\mathbf{sel}(l, x_1, x_2) = (\neg l \wedge x_1) \vee (l \wedge x_2)\,,$$

where x_1 is the value of the gate on u_1 and x_2 is the value of the gate on u_2. This gate returns the value of x_1 whenever $l \mapsto 0$ and x_2 whenever $l \mapsto 1$, this corresponds to our transformation where L_3 is replaced by L_1 if $l \mapsto 0$ and L_2 if $l \mapsto 1$.

Suppose u corresponds to a clause inferred by an application of GEN1 or GEN3. For each $i \in [z+2]$ let x_i be the gate corresponding to the clause L_i. If each resolvent l_i in these clauses is either a $\bar{q}$ variable or an $\bar{r}$ variable then we first remove from $C(\bar{p})$ the vertices labelled by the gates $x_1, \ldots, x_{z+1}$. As each of these vertices corresponds to some modal (and therefore initial) clause, every such vertex is a leaf and so $C(\bar{p})$ remains a circuit even after removing these vertices. We then put a no operation gate on u. Hence the gate on u returns the same value as the gate x_{z+2}, which corresponds to the literal clause L_{z+2}, as desired.

If at least one $l_i \in \bar{p}$ then we let $\{p_1, \ldots, p_m\} \subseteq \{l_1, \ldots, l_{z+1}\}$ be the subset of $\bar{p}$ variables resolved on and let $x'_1, \ldots, x'_m$ be the vertices corresponding to the associated modal clauses. We further put a m-ary selector gate, defined:

$$\mathbf{sel}_m\left(p_1, \ldots, p_m, x'_1, \ldots, x'_m, x_{z+2}\right) = \bigvee_{i=1}^{m} \left(\neg p_i \wedge x'_i\right) \vee \left(\bigwedge_{j=1}^{m} p_j \wedge x_{z+2}\right),$$

on u. Finally we remove from $C(\bar{p})$ all parent vertices of u that are labelled by gates which are not taken as inputs to our m-ary selector gate. That is we remove the vertices corresponding to modal q-clauses and r-clauses, noting once again that every such clause is initial. Our m-ary selector gate returns the value of x_{z+2} if every $p_i \mapsto 1$, it returns 1 if there exists some i such that the value returned by x'_i is 1 and $p_i \mapsto 0$, and it returns 0 if there exists some i such that $p_i \mapsto 0$ but for every such i the value of x_i is 0. In each case the output of the selector gate corresponds to the clause L_{z+2} is replaced by in our transformation.

Clearly the size of $C(\bar{p})$ is less than or equal to that of π. Further we can easily convert $C(\bar{p})$ into a circuit with basis $\{\wedge, \vee, \neg, \mathbf{0}, \mathbf{1}\}$ whose size is linear in that of $C(\bar{p})$. □

Corollary 6.2.1. Each proof system in the family of **K**-Res systems has modal feasible interpolation.

Proof. By Corollary 4.6.1 each of the **K**-Res systems is p-simulated by $\mathbf{K}^+_{mp}$-Res. Hence we can transform any $\mathbf{K}^+_{mc}$-Res refutation into a $\mathbf{K}^+_{mp}$-Res refutation of the corresponding set of SNF^+_{mp} clauses and so that the corollary holds follows trivially from Theorem 6.2.1. □

6.3 Monotone feasible interpolation for modal resolution systems

In this section we prove that $\mathbf{K}^+_{mp}$-Res has modal monotone feasible interpolation. As in the case for propositional resolution, it is this property which allows us to prove unconditional lower bounds for $\mathbf{K}^+_{mp}$-Res.

Theorem 6.3.1. $\mathbf{K}^+_{mp}$-Res has modal monotone feasible interpolation.

Proof. Let $A(\bar{p}, \bar{q}) \cup B(\bar{p}, \bar{r})$ be an unsatisfiable set of SNF^+_{mp} clauses, where $\bar{p}, \bar{q}$ and $\bar{r}$ are disjoint sets of propositional variables. Further let π be a $\mathbf{K}^+_{mp}$-Res refutation of this set of clauses. Suppose each of the $\bar{p}$ variables appears only positively in $A(\bar{p}, \bar{q})$. Then we can modify the circuit $C(\bar{p})$ constructed in the proof of Theorem 6.2.1 to obtain a monotone circuit $C^{mon}(\bar{p})$ such that $C^{mon}(\bar{\alpha}) = 0$ only if the set of clauses $A'(\bar{\alpha}, \bar{q})$ is unsatisfiable and $C^{mon}(\bar{\alpha}) = 1$ only if $B'(\bar{\alpha}, \bar{q})$ is unsatisfiable. To do this we must replace all selector gates **sel** and $\mathbf{sel}_m$ with monotone gates.

We replace every $\mathbf{sel}(l, x_1, x_2)$ gate in $C(\bar{p})$ with the monotone ternary gate $(l \vee x_1) \wedge x_2$. This gate differs from $\mathbf{sel}(l, x_1, x_2)$ on only one input, namely $l = 0$, $x_1 = 1$ and $x_2 = 0$. In this case L_1 is an r-clause, L_2 is a q-clause and the variable being resolved on is in $\bar{p}$. By assumption q-clauses contain only positive $\bar{p}$ variables hence L_2 cannot contain $\neg l$ and so in the transformation of our clauses we can replace L_3 with the q-clause L_2 without any issues.

We further replace every $\mathbf{sel}_m(p_1, \ldots, p_m, x'_1, \ldots, x'_m, x_{z+2})$ gate in $C(\bar{p})$ with the monotone gate:

$$(p_1 \vee x'_1) \wedge \cdots \wedge (p_m \vee x'_m) \wedge x_{z+2}.$$

The output of this gate differs from that of $\mathbf{sel}_m(p_1, \ldots, p_m, x'_1, \ldots, x'_m, x_{z+2})$ in two cases. If for some $i \neq j$ we have $p_i = 0$, $p_j = 0$, $x'_i = 1$ and $x'_j = 0$, then our monotone gate returns 0 whereas $\mathbf{sel}_m$ returns 1. Hence we change the transformation so that L_{z+3} is replaced by the modal clause containing p_i, which must be an r-clause as $x'_i = 1$. The other case is when $p_i = 0$, $x'_i = 1$ and for all $j \neq i$ either $p_j \neq 1$ or $x'_j \neq 1$, and $x_{z+2} = 0$. In this case $\mathbf{sel}_m$ returns 1, whereas our monotone gate returns 0. However, as the literal clause corresponding to x_{z+2} is a q-clause and so cannot contain any $\neg p_i$, instead of replacing L_{z+3} with the literal clause corresponding to x'_i we proceed in one of two ways. If $l_{z+1} \in \bar{q}$ and $\neg l_{z+1}$ appears in L_{z+2} then we replace L_{z+3} with the q-clause obtained by applying GEN1 to L_{z+2} and the set of modal clauses $\{L_j \mid j \in [z+1]$ and $\neg l_j$ appears in $L_{z+2}\}$. Otherwise we replace it with the q-clause obtained by

applying GEN3 to L_{z+2}, the set of modal clauses $\{L_j \mid j \in [z+1] \text{ and } \neg l_j \text{ appears in } L_{z+2}\}$ and $\Box_\mu \Diamond_a q'$.

Hence we have constructed a monotone circuit $C^{mon}(\bar{p})$ which interpolates $A'(\bar{\alpha}, \bar{q})$ and $B'(\bar{\alpha}, \bar{r})$, and whose size is polynomial in that of π. We can further transform this circuit into a monotone circuit over the basis $\{\wedge, \vee, \mathbf{0}, \mathbf{1}\}$ with at most a linear increase in size. □

Corollary 6.3.1. Each of the **K**-Res systems has modal monotone feasible interpolation.

Proof. This follows immediately from Theorem 6.3.1 and the fact that each of these proof systems is p-simulated by $\mathbf{K}^+_{mp}$-Res (Corollary 4.6.1). □

6.4 Lower bound

In this section we use the fact that $\mathbf{K}^+_{mp}$-Res has weak monotone feasible interpolation to prove an exponential lower bound for $\mathbf{K}^+_{mp}$-Res. The family of formulas for which we obtain this lower bound encode a well known graph theoretic result.

We say that a graph G has a *clique* of size k if there exists some k-subset (i.e. some subset of size k) of the vertices of G such that each vertex in this set is adjacent to every other vertex in this set. We say that G is *k-colourable* if the vertices of G can be partitioned into k disjoint subsets such that each pair of vertices in each subset are non-adjacent. It is a well known fact that any graph containing a clique of size $k+1$ is not k-colourable.

The statement "if a graph has a clique of size $k+1$ then it is not k-colourable" can be formulated as the propositional formula:

$$Clique^{k+1}_n(\bar{p}, \bar{q}) \rightarrow (\neg Colour^k_n(\bar{p}, \bar{r})),$$

where:

$$Clique^{k+1}_n(\bar{p}, \bar{q}) = \bigwedge_j \bigvee_i q_{ij} \wedge \bigwedge_{i, j_1 \neq j_2} (\neg q_{ij_1} \vee \neg q_{ij_2}) \wedge \bigwedge_{i_1 \neq i_2, j_1, j_2} ((q_{i_1 j_1} \wedge q_{i_2 j_2}) \rightarrow p_{i_1 i_2}),$$

$$\text{and} \qquad Colour^k_n(\bar{p}, \bar{r}) = \bigwedge_i \bigvee_j r_{ij} \wedge \bigwedge_{i_1 \neq i_2, j} (p_{i_1 i_2} \rightarrow (\neg r_{i_1 j} \vee \neg r_{i_2 j})),$$

for $i, i_1, i_2 \in [n]$ and $j, j_1, j_2 \in [k+1]$. In the above we interpret each variable q_{ij} as denoting that the ith vertex of some graph of size n is the jth element of some clique of size at most $k+1$. We further interpret each $p_{i_1 i_2}$ variable as denoting that there is an edge between the i_1th and i_2th vertex and each variable r_{ij} as denoting that the ith vertex of the graph is coloured by the jth colour in some k-colouring.

It is the above propositional formulas (henceforth referred to as the clique-colour formulas) that Pudlák used to show an exponential lower bound for propositional resolution via feasible interpolation in [75]. Clearly if we restrict $\mathbf{K}^+_{mp}$-Res to propositional formulas then it becomes the

propositional resolution system and so these formulas trivially give an exponential lower bound for $\mathbf{K}^+_{mp}$-Res. However this lower bound is of little interest as it offers no insight into the modal aspect of $\mathbf{K}^+_{mp}$-Res. The lower bound that we prove uses a modal version of these formulas first proposed by Hrubeš in [45, 46]. More specifically we will prove our lower bound using the formulas:

$$Clique^{k+1}_n(\Box\bar{p}, \bar{q}) \rightarrow \Box(\neg Colour^k_n(\bar{p}, \bar{r})),$$

where $Clique^{k+1}_n(\Box\bar{p}, \bar{q})$ denotes the formula obtained by replacing each $p_{i_1 i_2}$ in $Clique^{k+1}_n(\bar{p}, \bar{q})$ with $\Box p_{i_1 i_2}$. We call these formulas the *modal clique-colour formulas*.

In [45, 46] Hrubeš obtained an exponential lower bound on the size of $\mathbf{K}_n$-Frege (see Definition 10.1.3) proofs required for the modal clique-colour formulas. To do this he proved that $\mathbf{K}_n$-Frege admits a certain type of feasible interpolation. In particular he showed that for any given propositional formulas $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ the modal formula $A(\Box\bar{p}, \bar{q}) \rightarrow \Box B(\bar{p}, \bar{r})$ is a $\mathbf{K}_n$ tautology. Further any monotone circuit which interpolates $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ has size polynomial in that of the number of K axioms required to prove $A(\Box\bar{p}, \bar{q}) \rightarrow \Box B(\bar{p}, \bar{r})$. As this type of monotone interpolation can only be applied to modal formulas of the form $A(\Box\bar{p}, \bar{q}) \rightarrow \Box B(\bar{p}, \bar{r})$, it is in some sense less general than our notion of modal monotone feasible interpolation.

We will see in Chapter 10 that $\mathbf{K}_n$-Frege p-simulates $\mathbf{K}^+_{mp}$-Res (Proposition 10.1.1). Hence it follows trivially that the modal clique-colour formulas also require exponential sized $\mathbf{K}^+_{mp}$-Res refutations. We use the fact that $\mathbf{K}^+_{mp}$-Res has weak monotone feasible interpolation to give an alternative direct proof (i.e. a proof not reliant on the fact that $\mathbf{K}_n$-Frege p-simulates $\mathbf{K}^+_{mp}$-Res) of this lower bound.

To prove our lower bound we require the following well known result from circuit complexity.

Theorem 6.4.1 ([2])**.** Any monotone circuit over the basis $\{\vee, \wedge, \mathbf{0}, \mathbf{1}\}$ which decides whether or not a graph of size n has a clique of size $\sqrt{n}$ has size $2^{\Omega(n^{1/4})}$.

Theorem 6.4.2. Let:

$$\phi^k_n = Clique^{k+1}_n(\Box\bar{p}, \bar{q}) \rightarrow \Box(\neg Colour^k_n(\bar{p}, \bar{r}))$$

If we take $k = \sqrt{n}$ then every $\mathbf{K}^+_{mp}$-Res proof of ϕ^k_n has size $2^{\Omega(n^{1/4})}$.

Proof. As $\mathbf{K}^+_{mp}$-Res is a refutational proof system to prove ϕ^k_n we must refute its negation:

$$\neg\left(Clique^{k+1}_n(\Box\bar{p}, \bar{q}) \rightarrow \Box(\neg Colour^k_n(\bar{p}, \bar{r}))\right) \equiv Clique^{k+1}_n(\Box\bar{p}, \bar{q}) \wedge \Diamond Colour^k_n(\bar{p}, \bar{r}).$$

Hence we begin by letting $A(\bar{p}, \bar{q})$ and $B(\bar{p}, \bar{r})$ denote the sets of SNF^+_{mp} clauses corresponding to $Clique^{k+1}_n(\Box\bar{p}, \bar{q})$ and $\Diamond Colour^k_n(\bar{p}, \bar{r})$ respectively. That is, we let:

$$A(\bar{p},\bar{q}) = \{x\} \cup \{(x \to \Diamond x')\} \cup \{(\neg x \vee q_{1j} \vee \cdots \vee q_{nj}) \mid j \in [k+1]\} \cup$$
$$\{(\neg x \vee \neg q_{ij_1} \vee \neg q_{ij_2}) \mid i \in [n],\, j_1 \neq j_2 \in [k+1]\} \cup$$
$$\{(\neg x \vee \neg q_{i_1 j_1} \vee \neg q_{i_2 j_2} \vee x_{i_1 i_2}), (x_{i_1 i_2} \to \Box p_{i_1 i_2}) \mid i_1 \neq i_2 \in [n],\, j_1, j_2 \in [k+1]\},$$

and

$$B(\bar{p},\bar{r}) = \big\{\Box(\neg x' \vee r_{i1} \vee \cdots \vee r_{ik}) \mid i \in [n]\big\} \cup$$
$$\{\Box(\neg x' \vee \neg p_{i_1 i_2} \vee \neg r_{i_1 j} \vee \neg r_{i_2 j}) \mid i_1 \neq i_2 \in [n],\, j \in [k]\},$$

where $x_{i_1 i_2}, x$ are new $\bar{q}$ variables and x' is a new $\bar{p}$ variable. We further let π be a $\mathbf{K}^+_{mp}$-Res refutation of $A(\bar{p},\bar{q}) \cup B(\bar{p},\bar{r})$.

Clearly the $\bar{p}$ variables appear only positively in $A(\bar{p},\bar{q})$. Hence it follows by Theorem 6.3.1 that there exists a monotone circuit $C^{mon}(\bar{p})$ whose size is polynomial in that of the refutation π, and which interpolates:

$$A'(\bar{p},\bar{q}) = A(\bar{p},\bar{q}) \cup \{(x_{i_1 i_2} \to \Diamond p_{i_1 i_2}) \mid i_1 \neq i_2 \in [n],\, j_1, j_2 \in [k+1]\} \cup \{\Diamond q'\},$$
$$\text{and} \quad B'(\bar{p},\bar{r}) = B(\bar{p},\bar{r}) \cup \{\Diamond r'\},$$

where q' is a new $\bar{q}$ variable and r' is a new $\bar{r}$ variable.

Let $\bar{\alpha}$ be an assignment to the $\bar{p}$ variables and let G_α be the graph consisting of n vertices and having an edge between the $i_1 th$ and $i_2 th$ vertices if and only if $p_{i_1 i_2} \mapsto 1$. It is not hard to see that this graph has a clique of size $k+1$ if and only if $A'(\bar{\alpha},\bar{q})$ is satisfiable and a k-colouring if and only if $B'(\bar{\alpha},\bar{r})$ is satisfiable[1]. Finally we note that a graph cannot contain a $k+1$ clique if its k-colourable and cannot be k-colourable if it contains a $k+1$ clique, and so $A'(\bar{\alpha},\bar{q})$ is satisfiable if and only if $B'(\bar{\alpha},\bar{r})$ is unsatisfiable.

Hence $C^{mon}(\bar{p})$ is a monotone circuit which decides whether or not a graph has a clique of size $k+1$. By Theorem 6.4.1 every such circuit has exponential size and so as the size of $C^{mon}(\bar{p})$ is polynomial in $|\pi|$ it follows that π must also have exponential size. □

Corollary 6.4.1. The modal clique-colour formulas require proofs of size $2^{\Omega(n^{1/4})}$ in each of the **K**-Res systems.

Proof. As each of the **K**-Res systems is p-equivalent to $\mathbf{K}^+_{mp}$-Res (Corollary 4.6.1) the corollary follows immediately from Theorem 6.4.2. □

We finish by noting that whilst our lower bound is on a truly modal family of formulas (as demonstrated by Hrubeš proof that any $\mathbf{K}_n$-Frege proof of these formulas contains an exponential number of modal axioms), it is not a lower bound on the number of modal resolution steps (i.e. applications of GEN1 and GEN3) required in any $\mathbf{K}^+_{mp}$-Res proof of these formulas. In fact even the

[1]Note that the analogous statement also holds for $A(\bar{\alpha},\bar{q})$ but not for $B(\bar{\alpha},\bar{r})$ as this set of clauses can be satisfied for any assignment by taking a model with only one world and an empty accessibility relation.

non-direct proof of our lower bound does not give a lower bound for the number of modal resolution steps in such a refutation as even LRES, a seemingly propositional $\mathbf{K}^{+}_{mp}$-Res rule requires modal axioms if we wish to simulate it using $\mathbf{K}_n$-Frege.

Chapter 7

The size-width lower bound technique

Among the most successful lower bound techniques for propositional resolution is the size-width technique introduced by Ben-Sasson and Wigderson in [9]. In this chapter we address the natural question of whether or not a similar technique can be used to obtain lower bounds for modal resolution systems.

The size width-lower bound proving technique works as follows. For both tree-like and dag-like resolution there exists a fundamental relationship between the size of refutations and their width (i.e. the maximum number of literals in any clause in the refutation). This relationship can be exploited to obtain proof size lower bounds indirectly by proving lower bounds for their width. Indeed a number of exponential size lower bounds have been shown for tree-like and dag-like propositional resolution using this technique [9].

In this chapter we will show that analogous relationships do not hold between the size and width of either $\mathbf{RK}_n$ resolution proofs or proofs in any of the **K**-Res resolution systems defined in Chapter 4.

7.1 Width

Throughout this chapter we make use of *big omega notation* . That is, given two functions f and g we write $f(n) = \Omega(g(n))$ if and only if f is asymptotically bounded below by g. More formally, we write $f(n) = \Omega(g(n))$ if and only if there exists some constant $k > 0$ and some n_0 such that for every $n > n_0$ we have $f(n) \geq kg(n)$.

Definition 7.1.1 ([9])**.** The *width* of a propositional clause C is the number of literals it contains (denoted $w(C)$).
The *width* of a propositional CNF ϕ is the maximum width of any clause in the conjunction (denoted

$w(\phi)$).
The *width* of a propositional resolution proof π is the maximum width of any clause in the proof (denoted $w(\pi)$).

Let $S(\phi)$ denote the minimum size of any resolution refutation of the CNF ϕ, let $S_T(\phi)$ denote the minimum size of any tree-like refutation of ϕ and let $w(\phi \vdash 0)$ denote the minimum width of any resolution refutation of ϕ.

There exists a fundamental relationship between the size and width of propositional (tree-like) resolution refutations, namely that if a formula has polynomial size (tree-like) resolution refutations then it has constant width (tree-like) resolution refutations (Theorems 7.1.1 and 7.1.2). This relationship was first formalised by Ben-Sasson and Wigderson in [9].

Theorem 7.1.1 ([9]). $S_T(\phi) \geq 2^{w(\phi \vdash 0) - w(\phi)}$.

Theorem 7.1.2 ([9]). $S(\phi) \geq \exp\left(\Omega\left(\frac{(w(\phi \vdash 0) - w(\phi))^2}{n}\right)\right)$, where n denotes the number of variables in ϕ.

Theorem 7.1.1 states that, in tree-like resolution, every refutation of a CNF ϕ has size exponential in the difference between the minimum proof width required to refute ϕ and the width of ϕ. Whereas, Theorem 7.1.2 states that, in the full dag-like version of resolution, every refutation of a CNF ϕ has size exponential in the square of the difference between the minimum proof width required to refute ϕ and the width of ϕ over the total number of variables in ϕ.

Hence using Theorem 7.1.1 exponential proof size lower bounds for tree-like resolution can be obtained indirectly via linear width lower bounds and using Theorem 7.1.2 exponential proof size lower bounds for dag-like resolution can be obtained indirectly via linear width lower bounds. However neither theorem can be used to obtain proof size lower bounds for any CNF with $w(\phi \vdash 0) - w(\phi) = O(1)$. Further the formula for dag-like resolution refutations cannot be used to obtain proof size lower bounds for any CNF with $(w(\phi \vdash 0) - w(\phi))^2 = O(n)$.

Hence to show that these theorems do not hold for any of the modal resolution systems considered in this thesis we need to show that for each system there exist modal formulas with small initial width, a small number of variables, polynomial size refutations and which cannot be refuted with less than linear width.

7.2 Width for modal resolution systems

In this chapter we rule out the possibility of proving exponential proof size lower bounds via linear width lower bounds for each of the modal resolution systems defined in Chapters 4 and 5. We do this by giving a counterexample (i.e. a formula which has small initial width, a small number of variables and only linear width refutations, but can be refuted with polynomial size) for each of these systems.

In the following definition we extend the notion of width for propositional CNFs, propositional clauses and propositional resolution refutations to $\mathbf{RK}_n$ CNFs, $\mathbf{RK}_n$ clauses, $\mathbf{RK}_n$ refutations and **K**-Res refutations.

Definition 7.2.1. The *width* of an $\mathbf{RK}_n$ clause C is the number of literals it contains and is denoted $w(C)$.
The *width* of an $\mathbf{RK}_n$ CNF ϕ (or equivalently a set of $\mathbf{RK}_n$ clauses $\mathcal{C}$) is the maximum width of any $\mathbf{RK}_n$ clause in ϕ (equivalently any $C \in \mathcal{C}$) and is denoted $w(\phi)$ (equivalently $w(\mathcal{C})$).
The *width* of an $\mathbf{RK}_n$ (respectively a **K**-Res) resolution proof π is the maximum width of any clause in π and is denoted $w(\pi)$.

It is easy to see that the above definition also extends to SNF^+_{mp} clauses, sets of SNF^+_{mp} clauses and SNF^+_{mp} refutations.

Finally we remark that whilst our definition of the width of an $\mathbf{RK}_n$ clause is independent of the number of modal operators contained within it, a natural alternative way to extend the definition of propositional width would be to count the number of modal operators and the number of literals contained within an $\mathbf{RK}_n$ clause. However the formulas we use to show that superpolynomial proof size lower bounds cannot be obtained from linear width lower bounds for both the **K**-Res systems of Chapter 4 and the proof system $\mathbf{RK}_n$ defined in Chapter 5 contain only a constant number of modal operators. Hence even if we took this alternative definition of width these formulas would still be counterexamples for each of the modal resolution systems we consider.

7.3 A counterexample for the K-Res proof systems

In this section we show that Theorems 7.1.1 and 7.1.2 cannot hold for any of the **K**-Res proof systems defined in Chapter 4. To do this we construct a family of modal formulas which have only linear width **K**-Res refutations. We further show that this family of formulas have polynomial size **K**-Res refutations.

Consider the following unsatisfiable modal formula:

$$\psi_m = \bigwedge_{i=1}^{m} \Box l_i \wedge \Diamond \left(\bigvee_{i=1}^{m} \neg l_i \right).$$

Each of the **K**-Res systems can of course be used to refute ψ_m (or more specifically a set of clauses which is satisfiability equivalent to ψ_m). However, the set of clauses obtained by transforming ψ_m (in particular the subformula $\Diamond (\bigvee_{i=1}^{m} \neg l_i)$) into any of the normal forms defined in Chapter 4 contains a clause of width $m+1$. For example if we transform ψ_m into SNF^+_{mp} then some SNF^+_{mp} clause $\Box (\neg x \vee \bigvee_{i=1}^{m} \neg l_i)$, where x is an extension variable, must be contained within the resulting set of clauses. Hence ψ_m cannot be used to show that Theorems 7.1.1 and 7.1.2 do not hold for the **K**-Res systems.

However the formula:

$$\phi_m = \bigwedge_{i=1}^{m} \Box l_i \wedge \Box(\neg l_1 \vee l'_1) \wedge \bigwedge_{i=2}^{m-1} \Box(\neg l_i \vee l'_i \vee \neg l'_{i-1}) \wedge \Diamond(\neg l_m \vee \neg l'_{m-1}),$$

which can be obtained from ψ_m by adding extension variables $l'_1, \ldots, l'_{m-1}$, is unsatisfiable and has width 3. Hence we can be use ϕ_m as a counterexample for Theorems 7.1.1 and 7.1.2.

Our proof that every **K**-Res refutation of ϕ_m has linear width relies on the fact that the rules of the **K**-Res systems allow literals to be resolved with modal literals only in very specific circumstances. Namely, only when every literal in a literal clause can be resolved with a modal literal. Hence in each of the **K**-Res systems all l'_i variables must be resolved out of a clause before any l_i variable in the clause can be resolved on. The formulas ϕ_m are designed so that whenever an l'_i variable is resolved out of a clause the resolvent obtained contains the variables l_i and l_{i+1}. It is a direct result of this that every **K**-Res refutation of ϕ_m contains a bottleneck clause which contains every l_i variable and so has width m.

In the following theorem we formally show that the set of SNF^+_{mp} clauses obtained by applying the translation function T^+_{mp} to ϕ_m has only linear width $\mathbf{K}^+_{mp}$-Res refutations. This rules out the possibility of proving that either of the Theorems 7.1.1 and 7.1.2 hold for $\mathbf{K}^+_{mp}$-Res.

Theorem 7.3.1. Every $\mathbf{K}^+_{mp}$-Res refutation of the set of SNF^+_{mp} clauses obtained by applying T^+_{mp} to ϕ_m has width $\Omega(m)$. There also exist $\mathbf{K}^+_{mp}$-Res refutations of $T^+_{mp}(\phi_m)$ with size $O(m)$.

Proof. The set of SNF^+_{mp} clauses obtained by applying T^+_{mp} to ϕ_m is:

$$\mathcal{C}_m = \{x\} \cup \bigcup_{i=1}^{m-1} \{x \to \Box l_i\} \cup \{x \to \Diamond x_m\} \cup \{\Box(\neg x_1 \vee \neg l_1 \vee l'_1)\} \cup \bigcup_{j=2}^{m-1} \{\Box(\neg x_j \vee \neg l_j \vee l'_j \vee \neg l'_{j-1})\} \cup \{\Box(\neg x_m \vee \neg l_m \vee \neg l'_{m-1})\}.$$

We will first show that every $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}_m$ has width $2m$.

By definition the inference rule GEN1 (respectively GEN3) can only be applied to a set of SNF^+_{mp} clauses $\mathcal{C}'$ if said set contains exactly one literal clause C with width $z \in \mathbb{N}$ and z (respectively $z+1$) modal clauses. In particular if $C = \Box_{\mu a}(\neg y_1 \vee \cdots \vee \neg y_z)$ then for each $i \in [z]$ some modal clause of the form $\Box_\mu(y'_i \to \circ y_i)$ (respectively $\Box_\mu(y'_i \to \Box y_i)$) must be contained within $\mathcal{C}'$. Hence as each literal clause in $\mathcal{C}_m$ contains some variable l'_i and no l'_i appears in any modal clause in $\mathcal{C}_m$ it follows that neither GEN1 nor GEN3 can be applied to any subset of $\mathcal{C}_m$.

We will now show that every literal clause C which can be derived from $\mathcal{C}_m$ using only LRES inferences is either of the form:

$$\Box \bigvee_{i \in [m]} (\neg x_i \vee \neg l_i) \quad \text{or} \quad \Box \left(\bigvee_{i \in \{a,\ldots,b\}} (\neg x_i \vee \neg l_i) \vee A \right),$$

where $1 \leq a, b \leq m$ such that $\{a, \ldots, b\} \subset [m]$. Further:

$$A = \begin{cases} l'_b \vee \neg l'_{a-1} & \text{if } a > 1 \text{ and } b < m, \\ \neg l'_{a-1} & \text{if } a > 1 \text{ and } b = m, \\ l'_b & \text{if } a = 1 \text{ and } b < m. \end{cases}$$

Note that as $\{a, \ldots, b\} \subset [m]$ it is not possible for $a = 1$ and $b = m$.

Recall from Definition 3.2.2 that the length of a proof is the number of lines it contains. We prove the above claim by induction on the length of the LRES derivation used to obtain C. If the derivation has length 1 then $C \in \mathcal{C}_m$ and so the claim follows trivially.

Suppose C is obtained using an LRES derivation of length $z > 1$. Then C must be inferred from two clauses C_1 and C_2 which have LRES derivations of length z_1 and z_2 respectively, where $z_1 < z$ and $z_2 < z$. It follows by the inductive hypothesis that C_1 and C_2 are both of the form stated in our claim. Further as LRES is applied to C_1 and C_2 they must contain complementary literals. Hence:

$$C_1 = \square \left(\bigvee_{i_1 \in \{a_1, \ldots, b_1\}} (\neg x_{i_1} \vee \neg l_{i_1}) \vee A_1 \right) \text{ and } C_2 = \square \left(\bigvee_{i_2 \in \{a_2, \ldots, b_2\}} (\neg x_{i_2} \vee \neg l_{i_2}) \vee A_2 \right),$$

where A_1 and A_2 are non-empty subclauses of $\neg l'_{a_1-1} \vee l'_{b_1}$ and $\neg l'_{a_2-1} \vee l'_{b_2}$ respectively, and $1 \leq a_1, a_2, b_1, b_2 \leq m$ such that $\{a_1, \ldots, b_1\} \subset [m]$ and $\{a_2, \ldots, b_2\} \subset [m]$. Further the variable resolved on to infer C must be either l'_{b_1} or l'_{b_2}. We assume without loss of generality that the pivot variable is l'_{b_1}. Hence $a_2 - 1 = b_1$ and:

$$C = \begin{cases} \square \left(\bigvee_{i \in \{a_1, \ldots, b_2\}} (\neg x_i \vee \neg l_i) \right) & \text{if } a_1 = 1 \text{ and } b_2 = m, \\ \square \left(\bigvee_{i \in \{a_1, \ldots, b_2\}} (\neg x_i \vee \neg l_i) \vee l'_{b_2} \right) & \text{if } a_1 = 1 \text{ and } b_2 < m, \\ \square \left(\bigvee_{i \in \{a_1, \ldots, b_2\}} (\neg x_i \vee \neg l_i) \vee \neg l'_{a_1-1} \right) & \text{if } a_1 > 1 \text{ and } b_2 = m, \\ \square \left(\bigvee_{i \in \{a_1, \ldots, b_2\}} (\neg x_i \vee \neg l_i) \vee \neg l'_{a_1-1} \vee l'_{b_2} \right) & \text{if } a_1 > 1 \text{ and } b_2 < m. \end{cases}$$

This concludes the proof of our claim.

Clearly the empty clause cannot be derived using LRES alone. Further the only clause that can be derived from $\mathcal{C}_m$ using LRES to which either GEN1 or GEN3 can be applied is $\square \left(\bigvee_{i \in [m]} (\neg x_i \vee \neg l_i) \right)$. Hence every $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}_m$ must contain this clause. As this clause has width $2m$ it follows that every $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}_m$ also has width $2m$.

We will now prove that $\mathcal{C}_m$ has polynomial size $\mathbf{K}^+_{mp}$-Res refutations. We have already shown that every $\mathbf{K}^+_{mp}$-Res refutation of $\mathcal{C}_m$ contains the clause $\square \left(\bigvee_{i \in [m]} (\neg x_i \vee \neg l_i) \right)$. Further said clause can be derived using m applications of LRES as follows:

$$\text{LRES}\;\dfrac{\text{LRES}\;\dfrac{\text{LRES}\;\dfrac{\vdots}{\Box\left(\bigvee_{i\in[m-2]}(\neg x_i\vee\neg l_i)\vee l'_{m-2}\right)}\qquad B_{m-1}}{\Box\left(\bigvee_{i\in[m-1]}(\neg x_i\vee\neg l_i)\vee l'_{m-1}\right)}\qquad\Box(\neg x_m\vee\neg l_m\vee\neg l'_{m-1})}{\Box\left(\bigvee_{i\in[m]}(\neg x_i\vee\neg l_i)\right)}$$

where $B_j=\Box(\neg x_j\vee\neg l_j\vee l'_j\vee\neg l'_{j-1})$. This derivation contains $2m$ clauses of size at most linear in m and so has size at most polynomial in m. We can complete our refutation of $\mathcal{C}_m$ by applying GEN1 to:

$$\left\{\Box\left(\bigvee_{i\in[m]}(\neg x_i\vee\neg l_i)\right)\right\}\cup\left\{C'\in\mathcal{C}_m\mid C'\text{ is a modal clause}\right\},$$

to infer $\neg x$ and then resolving $\neg x$ with $x\in\mathcal{C}_m$ to infer 0. The whole refutation has size at most polynomial in m. □

An identical proof could be used to show that the corresponding translation of ϕ_m for any other **K**-Res system is a counterexample for that proof system. However, in the next corollary we instead use the p-equivalence of each of the **K**-Res systems to show that ϕ_m is a counterexample for every such system.

Corollary 7.3.1. For each of the **K**-Res proof systems ϕ_m has refutations of size $O(m)$, however every refutation of ϕ_m has width $\Omega(m)$.

Proof. The proofs of Theorems 4.5.1 and Theorem 4.6.3 are such that each of the **K**-Res systems p-simulate each other whilst preserving width. Hence the corollary follows immediately from Theorem 7.3.1. □

We conclude this subsection by remarking that neither ϕ_m nor $T^+_{mp}(\phi_m)$ is a counterexample for the proof system $\mathbf{RK}_n$. To see this consider the following examples.

Example 7.3.1. The following derivation is an $\mathbf{RK}_n$ refutation of $T^+_{mp}(\phi_m)$ which has width 4.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\pi_1\qquad\pi_2}{\neg x\vee\Box l'_2}\qquad\pi_3}{\neg x\vee\Box l'_3}\qquad\pi_4}{\vdots}}{\neg x\vee\Box l'_{m-2}}\qquad\pi_{m-1}}{\neg x\vee\Box l'_{m-1}}\qquad\pi_m}{\neg x\vee\Box\neg x_m}\qquad\neg x\vee\Diamond x_m}{\neg x}\qquad x}{0}$$

where π_1 denotes the $\mathbf{RK}_n$ derivation:

$$\dfrac{\dfrac{\neg x \vee \Box l_1 \qquad \Box(\neg x_1 \vee \neg l_1 \vee l'_1)}{\neg x \vee \Box(\neg x_1 \vee l'_1)} \qquad \neg x \vee \Box x_1}{\neg x \vee \Box l'_1}$$

For each $i \in \{2, \ldots, m-1\}$ π_i denotes the $\mathbf{RK}_n$ derivation:

$$\dfrac{\dfrac{\neg x \vee \Box l_i \qquad \Box(\neg x_i \vee \neg l_i \vee l'_i \vee \neg l'_{i-1})}{\neg x \vee \Box(\neg x_i \vee l'_i \vee \neg l'_{i-1})} \qquad \neg x \vee \Box x_i}{\neg x \vee \Box(l'_i \vee \neg l'_{i-1})}$$

and π_m denotes the $\mathbf{RK}_n$ derivation:

$$\dfrac{\neg x \vee \Box l_m \qquad \Box(\neg x_m \vee \neg l_m \vee \neg l'_{m-1})}{\neg x \vee \Box(\neg x_m \vee \neg l'_{m-1})}$$

Note that we have omitted the FFCRs for each inference from the above refutation.

Example 7.3.2. An $\mathbf{RK}_n$ refutation of ϕ_m with width 3 can be constructed from the $\mathbf{RK}_n$ refutation of $T^+_{mp}(\phi_m)$ given in Example 7.3.1 as follows. First we remove the final two resolution steps from the refutation then we replace π_1 with:

$$\dfrac{\Box l_1 \qquad \Box(\neg l_1 \vee l'_1)}{\Box l'_1}$$

replace each π_i with:

$$\dfrac{\Box l_i \qquad \Box(\neg l_i \vee l'_i \vee \neg l'_{i-1})}{\Box(l'_i \vee \neg l'_{i-1})}$$

and replace π_m with:

$$\dfrac{\Box l_m \qquad \Diamond(\neg l_m \vee \neg l'_{m-1})}{\Diamond((\neg l_m \vee \neg l'_{m-1}) \wedge l'_{m-1})}$$

Remark 7.3.1. It is an immediate consequence of the existence of the $\mathbf{RK}_n$ refutation of $T^+_{mp}(\phi_m)$ given in Example 7.3.1 that any p-simulation of either tree-like or dag-like $\mathbf{RK}_n$ by any tree-like or dag-like **K**-Res system cannot be width preserving.

7.4 A counterexample for $\mathbf{RK}_n$

In this subsection we rule out the possibility of proving that there exists a relationship, analogous to that of propositional resolution (Theorems 7.1.1 and 7.1.2), between the size and width of $\mathbf{RK}_n$ resolution refutations. To do this we define an $\mathbf{RK}_n$ CNF which has polynomial size refutations, but for which there do not exist sub-linear width $\mathbf{RK}_n$ refutations.

To find such a formula we exploit the fact that whenever we use $\mathbf{RK}_n$ to resolve on a pivot within the scope of some diamond operator either the subformula contained within this diamond operator disappears entirely, or the width of the formula increases. That is, if we resolve on some pivot in $\Diamond E$, where E is an $\mathbf{RK}_n$ CNF then the resolvent obtained either no longer contains $\Diamond E$ or, it contains some subformula of the form $\Diamond(E \vee A)$, where A is an $\mathbf{RK}_n$ clause. Hence for

our counterexample we construct an unsatisfiable $\mathbf{RK}_n$ CNF which requires a large number of resolution steps on pivots within diamond operators.

Consider the $\mathbf{RK}_n$ CNF:

$$\gamma_m = \Diamond\Box\neg x_{\geq 1} \wedge \bigwedge_{i=1}^{m} \Box\Diamond\left(x_{\geq 1} \vee \neg x_i\right) \wedge \Box\left(\bigvee_{i=1}^{m} \Box x_i\right).$$

To see that γ_m is unsatisfiable suppose there exists some world w_0 in some model M such that $(M, w_0) \models \gamma_m$. The first $\mathbf{RK}_n$ clause in γ_m states that there exists a world w_1 which is accessible from w_0 and that at every world accessible from w_1 the variable $x_{\geq 1}$ is false. The next m $\mathbf{RK}_n$ clauses say that from every world accessible from w_0 there exists an accessible world in which $x_{\geq 1}$ is true whenever x_i is true. Hence as (M, w_0) satisfies γ_m the model M must contain a submodel that is either as shown in Figure 7.1 or can be obtained from the model in Figure 7.1 by identifying worlds.

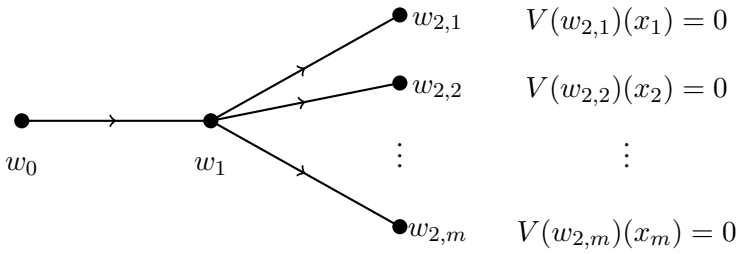

Figure 7.1: A model for $\Diamond\Box\neg x_{\geq 1} \wedge \bigwedge_{i=1}^{m} \Box\Diamond(x_{\geq 1} \vee \neg x_i)$

The remaining m $\mathbf{RK}_n$ clauses in γ_m say that for some $i \in [m]$ the variable x_i is true at every world of distance two from w_0. In particular, x_i is true at every $w_{2,j}$, where $j \in [m]$. This is not possible in the model shown in Figure 7.1, nor any model obtained from this model by identifying worlds, and so γ_m must be unsatisfiable.

Every resolution inference that can be applied to γ_m involves resolving on some pivot within the scope of a diamond operator. However γ_m has initial width m and so proving that every $\mathbf{RK}_n$ refutation of γ_m has linear width is both trivial and does not rule out the possibility of some relationship as in Theorem 7.1.1 or Theorem 7.1.2.

Hence we instead consider the following $\mathbf{RK}_n$ CNF, which is satisfiability equivalent to γ_m but has constant width:

$$\theta_m = \Diamond\Box\neg x_{\geq 1} \wedge \bigwedge_{i=1}^{m} \Box\Diamond\left(x_{\geq 1} \vee \neg x_i\right) \wedge \Box\left(\Box x_1 \vee x'_1\right) \wedge \bigwedge_{i=2}^{m-1} \Box\left(\neg x'_{i-1} \vee \Box x_i \vee x'_i\right) \wedge \Box\left(\neg x'_{m-1} \vee \Box x_m\right).$$

To see that θ_m is satisfiability equivalent to γ_m note that each x'_i essentially abbreviates the clause $\bigvee_{j=i+1}^{m} \Box x_j$.

The family of formulas θ_m has been designed so that every $\mathbf{RK}_n$ refutation of θ_m requires a large number of variables within $\Diamond$ operators to be resolved on. Every time such a variable resolved on either the subformula of the form $\Diamond A$ which it is contained within collapses to 0 or it is replaced by a wider clause. Hence by proving that in every $\mathbf{RK}_n$ refutation of θ_m either $\Diamond\Box\neg x_{\geq 1}$ or some clause of the form $\Box\Diamond(x_{\geq 1} \vee \neg x_i)$ has a linear number of descendants, each of which is wider than the descendant it was inferred from, we prove that θ_m has only linear width $\mathbf{RK}_n$ refutations.

A key property of the formula θ_m is that removing any single $\mathbf{RK}_n$ clause from it results in a satisfiable $\mathbf{RK}_n$ CNF.

Definition 7.4.1. We say that an $\mathbf{RK}_n$ CNF ϕ is *minimally unsatisfiable* if removing any single $\mathbf{RK}_n$ clause from ϕ results in a satisfiable $\mathbf{RK}_n$ CNF.

To see that θ_m is minimally unsatisfiable we first note that if we remove $\Diamond\Box\neg x_{\geq 1}$ from θ_m then the model $M = (W, R, V)$, where:

$$W = \{w_0, w_1, w_2\}, \quad R = \{(w_0, w_1), (w_1, w_2)\},$$

and V is such that:

$$V(w_2)(x_{\geq 1}) = 1 \quad \text{and} \quad V(w_2)(x_i) = 1,$$

for all $i \in [m]$, then M satisfies the resulting $\mathbf{RK}_n$ CNF at the world w_0. Similarly if we remove some $\Box\Diamond(x_{\geq 1} \vee \neg x_k)$ from θ_m then the resulting $\mathbf{RK}_n$ CNF is satisfied by (M', w_0), where $M' = (W, R, V')$ and V' is such that:

$$V'(w_2)(x_{\geq 1}) = 0, \;\; V'(w_2)(x_k) = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise,} \end{cases} \text{ and } \; V'(w_1)(x'_i) = \begin{cases} 1 & \text{if } i < k, \\ 0 & \text{otherwise.} \end{cases}$$

Finally the $\mathbf{RK}_n$ CNF obtained by removing some $\mathbf{RK}_n$ clause containing $\Box x_k$ is satisfied by (M'', w_0), where $M'' = (W, R, V'')$ and V'' is such that:

$$V''(w_2)(x_{\geq 1}) = 0, \;\; V''(w_2)(x_i) = 0 \text{ for all } i \in [m], \;\; \text{and } \; V''(w_1)(x'_i) = \begin{cases} 1 & \text{if } i < k, \\ 0 & \text{otherwise.} \end{cases}$$

In the following theorem we give the main result of this section.

Theorem 7.4.1. The family of formulas θ_m are such that the minimum width required to refute θ_m using $\mathbf{RK}_n$ is $\Omega(m)$, whereas the minimum size required to refute θ_m using $\mathbf{RK}_n$ is $O(m)$.

The proof of this theorem (given at the end of Subsection 7.4.3) consists of two parts. We first show in Subsection 7.4.2 that every $\mathbf{RK}_n$ refutation of θ_m has linear width. We then show in Subsection 7.4.3 that despite this, there exist polynomial size $\mathbf{RK}_n$ refutations of θ_m. In fact, in Subsection 7.4.2 we only prove that every $\mathbf{RK}_n$ refutation of θ_m which contains no inferences of a certain form has width at least linear in m. Hence for this result to be sufficient to conclude that

every $\mathbf{RK}_n$ refutation of θ_m has width m we first prove in Subsection 7.4.1 that we can assume without loss of generality that no $\mathbf{RK}_n$ refutation contains this type of inference.

7.4.1 Removing propositional weakening from $\mathbf{RK}_n$ refutations

Recall from Section 4.1 that the proof system $\mathbf{RK}_n$ admits two types of inferences, weakening inferences and resolution inferences.

Definition 7.4.2. We say that an $\mathbf{RK}_n$ clause C is inferred by *propositional weakening* if it is inferred by weakening some $\mathbf{RK}_n$ clause C_1 by some $\mathbf{RK}_n$ clause C_2 less A, where A has the same modal depth as C_2.

In Subsection 7.4.2 we prove that every $\mathbf{RK}_n$ refutation of θ_m which contains no propositional weakening inferences has width $\Omega(m)$. Hence for this result to be sufficient to conclude that every $\mathbf{RK}_n$ refutation of θ_m has width $\Omega(m)$ we require the following proposition.

Proposition 7.4.1. Let $\mathcal{C}$ be an unsatisfiable set of $\mathbf{RK}_n$ clauses. Given any $\mathbf{RK}_n$ refutation π of $\mathcal{C}$ we can construct a new refutation π' of $\mathcal{C}$ which contains no propositional weakening steps. Further $|\pi'| \leq |\pi|$ and $w(\pi') \leq w(\pi)$.

Proof. The proof is by induction on the number of $\mathbf{RK}_n$ clauses in π which are inferred using propositional weakening. If π contains no $\mathbf{RK}_n$ clauses which are inferred using propositional weakening then we let $\pi' = \pi$.

Suppose π contains $r > 0$ propositional weakening inferences. Let C be the last $\mathbf{RK}_n$ clause in π which is inferred using propositional weakening. Further suppose C_1 is weakened by some C_2 less A to obtain C. By definition the modal depth of A is equal to that of C_2 and so C must be of the form $B_1 \vee B_2 \vee B_3$, where B_1, B_2 and B_3 are $\mathbf{RK}_n$ clauses such that $C_1 = B_1 \vee B_3$ and C_2 is the normal form of $A \vee B_2 \vee B_3$.

Further the refutation π must be of the form $\pi_1, R_1, \ldots, R_z, C, \pi_2$, where $R_1, \ldots, R_z$ is the sequence of FFCRs used to weaken C_1 by C_2 less A and each π_i is an $\mathbf{RK}_n$ derivation. Suppose:

$$\pi_2 = R_{1,1}, \ldots, R_{1,z_1}, D_1, R_{2,1}, \ldots, R_{2,z_2}, D_2 \ldots, R_{s,z_s}, \ldots, R_{s,z_s}, D_s,$$

where each D_i is an $\mathbf{RK}_n$ clause and each $R_{i,1}, \ldots, R_{i,z_i}$ is the sequence of FFCRs used to infer D_i. We assume without loss of generality that each D_i is a descendant of C.

To complete our induction we make use of the following claim.

Claim. Given π_2 we can construct an $\mathbf{RK}_n$ derivation:

$$\pi_3 = R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, R'_{2,1}, \ldots, R'_{2,z_2}, D'_2 \ldots, R'_{s,z_s}, \ldots, R'_{s,z_s}, D'_s$$

which contains no propositional weakening inferences and is such that $|\pi_3| \leq |\pi_2|$ and $w(\pi_3) \leq w(\pi_2)$. Further $\pi' = \pi_1, \pi_3$ is a refutation of $\mathcal{C}$ and if D_i is inferred from some pair of $\mathbf{RK}_n$ clauses

G_1 and G_2 then either $D'_i = G'_1$, $D'_i = G'_2$ or D'_i is inferred from G'_1 and G'_2, where:

$$G'_1 = \begin{cases} G_1 & \text{if } G_1 \text{ is in } \pi_1, \\ C_1 & \text{if } G_1 = C, \\ D'_j & \text{if } G_1 = D_j \text{ for some } j \in [s]. \end{cases} \quad \text{and} \quad G'_2 = \begin{cases} G_2 & \text{if } G_2 \text{ is in } \pi_1, \\ C_1 & \text{if } G_2 = C, \\ D'_j & \text{if } G_2 = D_j \text{ for some } j \in [s]. \end{cases}$$

Similarly if D_i is inferred from a single $\mathbf{RK}_n$ clause G_1 then either $D'_i = G'_1$ or D'_i is inferred from G'_1, where G'_1 is defined as above.

It follows from the above claim that π' contains $r - 1$ propositional resolution inferences and that $|\pi'| = |\pi_1| + |\pi_3| \leq |\pi|$ and $w(\pi') = \max\{w(\pi_1), w(\pi_2)\} \leq w(\pi)$. Hence the lemma follows by the inductive hypothesis. All that remains is to give a proof of the claim.

Proof of Claim. The proof is by induction on (a) the number of descendants of C in π and (b) the number of children of C in π. For base case (a) we suppose that C has only one descendant. By assumption this descendant must be 0. As only one pivot can be resolved on in a single $\mathbf{RK}_n$ inference it follows that either $B_2 = B_3 = 0$ or $B_1 = B_2 = 0$ (it cannot be the case that $B_1 = B_3 = 0$ as then $C_1 = 0$). In either case $C = C_1$ and so we let $\pi_3 = \pi_2$.

Suppose C has $s > 1$ descendants. Then C must be used to infer $D_1 \neq 0$. Further, by assumption D_1 cannot be inferred using propositional weakening. Suppose D_1 is the only child of C in π. The construction of π_3 depends on whether D_1 is inferred by resolution or non-propositional weakening.

Suppose D_1 is inferred by resolution on pair of $\mathbf{RK}_n$ clauses (respectively a single $\mathbf{RK}_n$ clause). Then some pivot x_1 in C must be resolved with some pivot x_2 in some $\mathbf{RK}_n$ clause G (respectively some other pivot x_2 in C). Recall $C = B_1 \vee B_2 \vee B_3$. Hence x_1 (respectively x_1 and x_2) must be contained within some B_i. If $i \in \{1, 3\}$ then we let D'_1 be the $\mathbf{RK}_n$ clause obtained by resolving x_1 in $C_1 = B_1 \vee B_3$ with x_2 in G (respectively x_1 in C_1 with x_2 in C_1) and let $R'_{1,1}, \ldots, R'_{1,z_1}$ be the associated sequence of FFCRs. Otherwise we let $D'_1 = C_1$ and let $R'_{1,1}, \ldots, R'_{1,z_1}$ be empty.

We define an $\mathbf{RK}_n$ derivation π_{D_1} of D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2\}$) as follows:

1. If x_1 is in B_1 (respectively if x_1 and x_2 are both in B_1) then we can derive D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2\}$) by weakening D'_1 by C_2 less A. Hence we let π_{D_1} denote this derivation.

2. If the x_1 is in B_2 or B_3 (respectively if x_1 and x_2 are both in B_1 or are both in B_3) then we can derive D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2\}$) by resolving x_1 in C_2 with x_2 in G (respectively by resolving together x_1 and x_2 in C_2) and then weakening D'_1 by the resultant $\mathbf{RK}_n$ clause less A. Hence we let π_{D_1} denote this derivation.

In either case π_{D_1} contains precisely one propositional weakening inference and this inference is used to infer D_1. Hence if we let π'_2 be the $\mathbf{RK}_n$ derivation obtained from π_2 by removing $R_{1,1}, \ldots, R_{1,z_1}, D_1$ then:

$$\pi_1, R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi_{D_1}, \pi'_2$$

is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$. Further the last $\mathbf{RK}_n$ clause in this refutation which is inferred using propositional weakening is D_1, and D_1 has strictly less than s descendants. Hence it follows by inductive hypothesis (a) that π_2' can be transformed into some π_2'' such that:

$$\pi_1, R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi_{D_1}, \pi''_2$$

is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$ and all other conditions of our claim are satisfied. However as the only $\mathbf{RK}_n$ clause in π_{D_1} which is used to infer any $\mathbf{RK}_n$ clause is π_2' is D, it follows that no $\mathbf{RK}_n$ clause in π_2'' is a descendant of any $\mathbf{RK}_n$ clause in π_{D_1}. Hence:

$$\pi_1, R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi''_2$$

is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$ and so we let $\pi_3 = R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi''_2$. It follows that $|\pi_3| \leq |\pi_2|$ and $w(\pi_3) \leq w(\pi_2)$.

Now suppose D_1 is inferred from C and some $\mathbf{RK}_n$ clause G (respectively from only C) by non-propositional weakening. Then either C is weakened by some G less some A_1, or some G is weakened by C less some A_1 (respectively C is weakened by itself less some A_1). As D_1 is not inferred by propositional weakening, the modal depth of A_1 must be strictly less than that of C. The last non-simplification rule used to obtain the sequence of FFCRs $R_{1,1}, \ldots, R_{1,z_1}$ must be the $\Sigma\,\mathsf{V}$-rule (respectively the $\Gamma\,\mathsf{V}$-rule). Further this RFCR must be used to add $B_{i_1} \vee B_{i_2}$ to an the assumed clause in some $R_{1,i}$, where $i_1, i_2 \in [3]$ and $i_1 \neq i_2$. If $\{i_1, i_2\} = \{1, 3\}$ then we let $D'_1 = C_1$ and let $R'_{1,1}, \ldots, R'_{1,z_1}$ be empty. Otherwise, if D_1 was inferred by weakening C by G less A_1 then we let D'_1 be the $\mathbf{RK}_n$ clause obtained by weakening C_1 by G less A_1, and if D_1 was inferred by weakening C by G less A_1 then we let D'_1 be the clause obtained by weakening G by C_1 less A_1 (respectively by weakening C_1 by itself less A_1). In either case we further let $R'_{1,1}, \ldots, R'_{1,z_1}$ be the corresponding sequence of FFCRs.

As in the case where D_1 is inferred by resolution we proceed to define an $\mathbf{RK}_n$ derivation of D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2, C_1\}$). This is done as follows:

1. If $i_1, i_2 = \{2, 3\}$ and D_1 was inferred by weakening C by G (respectively itself) less A_1 in π then we can derive D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2, C_1\}$) by weakening C_2 by G less A_1 (respectively by weakening C_1 by G less A_1) and then weakening D'_1 by the inferred clause less A_1. Hence we let π_{D_1} denote this derivation.

2. If $i_1, i_2 = \{2, 3\}$ and D_1 was inferred by weakening G by C less A_1 then we can derive D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2, C_1\}$) by weakening D'_1 by C_2 less A. Hence we let π_{D_1} denote this derivation.

3. Otherwise if $i_1, i_2 = \{1, 3\}$ and D_1 was inferred by weakening $H_1 \in \{C, G\}$ by $H_2 \in (\{C, G\} \backslash \{H_1\})$ less A_1 then we can derive D_1 from $\{D'_1, C_2, G\}$ (respectively $\{D'_1, C_2, C_1\}$) by weakening C_2 by G less A_1 (respectively by weakening C_1 by G less A_1) and then weakening D'_1 by the inferred clause less A_1. We let π_{D_1} denote this derivation.

We can then proceed to construct π_3 as in the case where D_1 was inferred by weakening.

Finally suppose C has $k > 1$ children. Let the sequence of FFCRs $R'_{1,1}, \ldots, R'_{1,z_1}$, the $\mathbf{RK}_n$ clause D'_1 and the derivations π_{D_1} and π'_2 be defined as above. Then the $\mathbf{RK}_n$ derivation:

$$\pi_1, R_1, \ldots, R_z, C, R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi_{D_1}, \pi'_2,$$

is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$. As in the case where C has only one descendant it follows by inductive hypothesis (a) that there exists some $\mathbf{RK}_n$ derivation π''_2 which contains no propositional weakening inferences and is such that $|\pi''_2| \leq |\pi'_2|$, $w(\pi''_2) \leq w(\pi'_2)$ and:

$$\pi_1, R_1, \ldots, R_z, C, R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi''_2,$$

is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$. Further C has at most $k-1$ children in this $\mathbf{RK}_n$ refutation. Hence by inductive hypothesis (b) π''_2 can be transformed into some π_3 such that $\pi' = \pi_1, \pi_3$ is an $\mathbf{RK}_n$ refutation of $\mathcal{C}$ and the other conditions of the claim are met. In particular $|\pi_3| \leq |\pi''_2|$ and $w(\pi_3) \leq w(\pi''_2)$. Hence as $|\pi''_2| \leq |\pi'_2|$ and $w(\pi''_2) \leq w(\pi'_2)$ we have $|\pi_3| \leq |\pi'''_2| \leq |\pi_2|$ and $w(\pi_3) \leq w(\pi'''_2) \leq w(\pi_2)$ respectively, where $\pi'''_2 = R'_{1,1}, \ldots, R'_{1,z_1}, D'_1, \pi''_2$. □

7.4.2 Proving that θ_m requires large width $\mathbf{RK}_n$ refutations

In this subsection we prove that every $\mathbf{RK}_n$ refutation of θ_m which contains no propositional weakening steps has width $\Omega(m)$ (Theorem 7.4.2).

This proof consists of two parts. We first show that every $\mathbf{RK}_n$ refutation of θ_m in which the $\mathbf{RK}_n$ clause $\Diamond\Box\neg x_{\geq 1}$ has at least m descendants must have width at least m. The second part of the proof is to show that every $\mathbf{RK}_n$ refutation of θ_m in which the clause $\Diamond\Box\neg x_{\geq 1}$ has less than m descendants also has width at least m. The first half of the proof uses only facts about the structure of descendants of $\Diamond\Box\neg x_{\geq 1}$. Whereas the second half requires that we show a lower bound on the number of *essential* clauses in such a refutation. Intuitively, a clause is essential to an $\mathbf{RK}_n$ refutation only if it contributes to the contradiction exposed in the refutation.

Definition 7.4.3. Let $\mathcal{C}$ be an unsatisfiable set of $\mathbf{RK}_n$ clauses and let π be some $\mathbf{RK}_n$ refutation of $\mathcal{C}$. We say that a clause C_1 in π is *essential* to π if there exists some path of $\mathbf{RK}_n$ clauses $C_1, \ldots, C_z$, through π, where each C_i is inferred from C_{i-1} (and possibly some other $\mathbf{RK}_n$ clause). Further C_i is either inferred by resolution or by weakening C_{i-1} (either propositionally or non-propositionally) by some other $\mathbf{RK}_n$ clause.

In the following lemma we show that every $\mathbf{RK}_n$ clause contained within a minimally unsatisfiable $\mathbf{RK}_n$ CNF $\mathcal{C}$ is essential to every $\mathbf{RK}_n$ refutation of $\mathcal{C}$.

Lemma 7.4.1. Let $\mathcal{C}$ be an unsatisfiable $\mathbf{RK}_n$ CNF and let π be an $\mathbf{RK}_n$ refutation of $\mathcal{C}$. If there exists some subset $\mathcal{D}$ of $\mathcal{C}$ such that $\mathcal{C} \setminus \mathcal{D}$ is satisfiable then some $C_i \in \mathcal{D}$ must be essential to π.

Further if we let $\mathcal{D}'$ denote the subset of $\mathcal{D}$ consisting only of clauses that are essential to π then $(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$ is unsatisfiable.

Proof. Clearly some $C_i \in \mathcal{D}$ must be used in the refutation π as otherwise π would be a refutation of the set of $\mathbf{RK}_n$ clauses $\mathcal{C} \setminus \mathcal{D}$, contradicting our assumption that this set is satisfiable.

For each $i \in [z]$ let $\mathcal{E}_i = \{E_{i,1}, \ldots, E_{i,y}\}$ be the set of all $\mathbf{RK}_n$ clauses in π which are inferred using $C_i \in \mathcal{D}$. Further let:

$$\mathcal{E} = \left\{ E \in \bigcup_{i \in [z]} \mathcal{E}_i \mid E \text{ is essential to } \pi \right\}.$$

By definition, every $\mathbf{RK}_n$ clause C which is essential to π must either be the empty clause or be used to infer some $\mathbf{RK}_n$ clause C' which is also essential to π. Hence if we let:

$$\mathcal{D}'' = \{C_i \in \mathcal{D} \mid \text{some } E_{i,j} \in \mathcal{E} \text{ is inferred by resolving on } C_i\} \cup \{C_i \in \mathcal{D} \mid \text{some } E_{i,j} \in \mathcal{E} \text{ is inferred by weakening } C_i \text{ by some } C\},$$

and:

$$\mathcal{D}' = \begin{cases} \mathcal{D}'' \cup \{0\} & \text{if } 0 \in \mathcal{D} \text{ and } 0 \text{ is used in } \pi, \\ \mathcal{D}'' & \text{otherwise,} \end{cases}$$

then $\mathcal{D}'$ is the set of all $\mathbf{RK}_n$ clauses in $\mathcal{D}$ which are essential to π.

We proceed by induction on the number of $\mathbf{RK}_n$ descendants of $\mathcal{D}$ in π. Suppose $\mathcal{D}$ has no descendants. Then as π contains some $C_i \in \mathcal{D}$ we must have $0 \in \mathcal{D}$ and $\pi = 0$. Hence $\mathcal{D}' = \{0\}$ and so $((\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}') \supseteq \{0\}$ is unsatisfiable.

Suppose $\mathcal{D}$ has $k > 0$ descendants. If $\mathcal{D}'$ contains 0 then $((\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}') \supseteq \{0\}$ and so $((\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}')$ is clearly unsatisfiable. Hence we further suppose that $0 \notin \mathcal{D}'$. It follows that π contains a subrefutation π' of:

$$(\mathcal{C} \setminus \mathcal{D}) \cup \bigcup_{i \in [z]} \mathcal{E}_i,$$

and so this set of $\mathbf{RK}_n$ clauses must be unsatisfiable. It follows from the definition of each $\mathcal{E}_i$ that if some $\mathbf{RK}_n$ clause is a descendant of some $E_{i,j} \in \mathcal{E}_i$ then it must also be a descendant of C_i. Hence the number of descendants of $\bigcup_{i \in [z]} \mathcal{E}_i$ in π', which we denote as k', is at most equal to k. Further $k' = k$ only if every clause in $\bigcup_{i \in [z]} \mathcal{E}_i$ is a descendant of some other clause in $\bigcup_{i \in [z]} \mathcal{E}_i$, however this cannot be true as if it were then the refutation would contain a cycle. Hence $k' < k$ and so it follows by the inductive hypothesis that some $E_{i,j} \in \bigcup_{i \in [z]} \mathcal{E}_i$ is essential to π' and that the set:

$$(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{E},$$

is unsatisfiable.

We will now prove that if some model $M = (W, R, V)$ and some world $w \in W$ are such that $(M, w) \models (\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$ then it must also be the case that $(M, w) \models C_j$, for every $\mathbf{RK}_n$ clause C_j which is essential to π. We prove this by induction on the length of the sub-derivation π_j of C_j

contained within π. If π_j has length 1 then $C_j \in \mathcal{C}$. Further as C_j is essential to π it follows by definition that $C_j \in \mathcal{D}$ only if $C_j \in \mathcal{D}'$, and so if $(M, w) \models (\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$ then $(M, w) \models C_j$.

Suppose π_j has length at least 2 and that $(M, w) \models (\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$. There are two cases:

(a) If C_j is inferred by resolution then, as C_j is essential to π, each of the $\mathbf{RK}_n$ clauses it is inferred from must also be essential to π. Hence it follows by the inductive hypothesis that both of these $\mathbf{RK}_n$ clauses are satisfied by (M, w) and so, by the strong soundness of $\mathbf{RK}_n$, we have $(M, w) \models C_j$.

(b) If C_j is inferred by weakening some C'_j by some C''_j (where, possibly $C'_j = C''_j$) then C'_j must be essential to π. Hence it follows by the inductive hypothesis that $(M, w) \models C'_j$ and then by Proposition 5.1.1 that $(M, w) \models C_j$.

As every $E_{i,j} \in \mathcal{E}$ is essential to π it follows from the above that every model which satisfies $(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$ must also satisfy $\mathcal{E}$. As:

$$(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}' \cup \mathcal{E} \supseteq (\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{E},$$

and $(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{E}$ is unsatisfiable the set $(\mathcal{C} \setminus \mathcal{D}) \cup \mathcal{D}'$ must also be unsatisfiable. Note that as $\mathcal{C} \setminus \mathcal{D}$ is assumed to be satisfiable this means that $\mathcal{D}'$ must be non-empty (i.e. there exists some $C \in \mathcal{D}$ which is essential to π). □

Remark 7.4.1. As θ_m is minimally unsatisfiable it follows immediately from the above lemma that every $\mathbf{RK}_n$ clause in θ_m is essential to every $\mathbf{RK}_n$ refutation of θ_m.

We will shortly give the main theorem of the section. However first we give two lemmas. The statements of these two lemmas both assert useful facts about the form which $\mathbf{RK}_n$ refutations of θ_m must take.

Lemma 7.4.2. Let π be an $\mathbf{RK}_n$ refutation of θ_m. If π contains no propositional weakening inferences then every descendant of the $\mathbf{RK}_n$ clause $\Diamond\Box\neg x_{\geq 1}$ contained within π is either the empty clause 0 or is of the form $\Diamond E$, where E is an $\mathbf{RK}_n$ CNF. In particular if C is inferred from some $\Diamond E$ then $C = \Diamond(E \wedge D)$.

Proof. It follows from the definition of the rules of $\mathbf{RK}_n$ that the last non-simplification RFCR used when resolving together two $\mathbf{RK}_n$ clauses of the form $\Box B_1$ and $\Box B_2$ (respectively when resolving on two pivots within a single $\mathbf{RK}_n$ clause of the form $\Box B_1$) must be the Σ $\Box\Box$-rule (respectively the Γ $\Box$-rule). Hence any such resolvent must be of the form $\Box B_3$. Further when weakening some $\mathbf{RK}_n$ clause of the form $\Box B_1$ by some $\mathbf{RK}_n$ clause $\Box B_2$ less A, where A has modal depth strictly less than that of $\Box B_2$ the last non-simplification RFCR used must be either the Σ $\Box\Box$-rule or the Γ $\Box$-rule. Hence the inferred $\mathbf{RK}_n$ clause must be of the form $\Box B_3$.

Similarly the last non-simplification RFCR used when either resolving together any two $\mathbf{RK}_n$ clauses of the form $\Diamond E$ (where by definition E is an $\mathbf{RK}_n$ CNF) and $\Box B$ respectively, or resolving on two pivots within a single $\mathbf{RK}_n$ clause of the form $\Diamond E$, must be either the Σ $\Box\Diamond$-rule, the Σ

$\Diamond\Box$-rule, the $\Gamma\ \Diamond 1$-rule or the $\Gamma\ \Diamond 2$-rule. Hence such an inference results in either the empty clause (if the simplification rule **S1** can be applied) or an $\mathbf{RK}_n$ clause of the form $\Diamond(E_1 \wedge D)$, where D is an $\mathbf{RK}_n$ clause. The same is true when non-propositionally weakening some $\mathbf{RK}_n$ clause of the form $\Diamond E_1$ by some $\mathbf{RK}_n$ clause $\Box B$ less A.

Finally we note that two $\mathbf{RK}_n$ clauses of the form $\Diamond E_1$ and $\Diamond E_2$ cannot be resolved together, and that no $\mathbf{RK}_n$ clause can be weakened non-propositionally by some $\mathbf{RK}_n$ clause of the form $\Diamond E_1$.

The $\mathbf{RK}_n$ CNF θ_m contains only $\mathbf{RK}_n$ clauses of the form $\Box B_1$ and $\Diamond E_1$. Hence in every $\mathbf{RK}_n$ refutation of θ_m which does not contain any propositional weakening inferences it must be the case that every descendant D of $\Diamond\Box\neg x_{\geq 1}$ is either the empty clause or is of the form $\Diamond E$. Further if $D = \Diamond E$ and is inferred from some other descendant $D' = \Diamond E'$ of $\Diamond\Box\neg x_{\geq 1}$ then $E = E' \wedge C$ for some $\mathbf{RK}_n$ clause C. □

Lemma 7.4.3. Let π be an $\mathbf{RK}_n$ refutation of θ_m. If π contains no propositional weakening steps then we can construct a new $\mathbf{RK}_n$ refutation π' of θ_m in which the $\mathbf{RK}_n$ clause $\Diamond\Box\neg x_{\geq 1}$, and each of its descendants (excluding the empty clause), are used as a premise of exactly one inference. Further $w(\pi') \leq w(\pi)$ and $|\pi'| \leq |\pi|$.

Proof. By Lemma 7.4.2 every descendant of $\Diamond\Box\neg x_{\geq 1}$ which is not the empty clause must be of the form $\Diamond E$, where E is an $\mathbf{RK}_n$ CNF. As none of the RFCR for $\mathbf{RK}_n$ can be used to simultaneously add a $\Diamond$ operator to both the 1st and 2nd assumed clause it follows that no two $\mathbf{RK}_n$ clauses of this form can be resolved together. Hence no two descendants of $\Box\Diamond\neg x_{\geq 1}$ can be resolved with one another. Similarly as no $\mathbf{RK}_n$ clause of the form $\Diamond E_1$ can be weakened by any clause of the form $\Diamond E_2$ less D, where D is a subclause of E_1 it follows that $\Diamond_a E_1$ cannot be weakened non-propositionally by $\Diamond E_2$. Hence as, by assumption π also contains no propositional weakening inferences this means that no clause of the form $\Diamond E_1$ can be weakened by any clause of the form $\Diamond E_2$. That is, no descendant of $\Box\Diamond\neg x_{\geq 1}$ can be weakened by any other resolvent of $\Box\Diamond\neg x_{\geq 1}$.

It follows that if some $\mathbf{RK}_n$ refutation π of θ_m contains two $\mathbf{RK}_n$ clauses inferred using $\Diamond\Box\neg x_{\geq 1}$ then at most one of these $\mathbf{RK}_n$ clauses is an ancestor of 0. Any $\mathbf{RK}_n$ clause that is not an ancestor of 0, along with all of its descendants, can be removed from π to obtain a new refutation of θ_m whose size and width are upped bounded by $|\pi|$ and $w(\pi)$ respectively. Hence we let π' be the $\mathbf{RK}_n$ refutation of θ_m obtained by doing and so $\Diamond\Box\neg x_{\geq 1}$ is taken as a premise in exactly[1] one inference in π'.

Similarly, if some descendant of $\Diamond\Box\neg x_{\geq 1}$ is used to infer two $\mathbf{RK}_n$ clauses then at most one of these can be an ancestor of 0 and so we can remove one of these clauses from π to obtain a new $\mathbf{RK}_n$ refutation of θ_m where every descendant of $\Diamond\Box\neg x_{\geq 1}$ is taken as a premise for exactly one inference. Further the size and width of this new refutation are upper bounded by $|\pi|$ and $w(\pi)$ respectively. □

We will now prove the main theorem of this subsection.

[1]That $\Diamond\Box\neg x_{\geq 1}$ is taken as a premise in *exactly* one resolution step as opposed to at *most* one follows from the fact that θ_m is minimally unsatisfiable.

Theorem 7.4.2. Every $\mathbf{RK}_n$ refutation of θ_m has width linear in m.

Proof. Let π be an $\mathbf{RK}_n$ refutation of θ_m. By Proposition 7.4.1 we can assume without loss of generality that π contains no propositional weakening steps. Further by Lemma 7.4.3 we can assume without loss of generality that the $\mathbf{RK}_n$ clause $\Diamond\Box\neg x_{\geq 1}$ and each of its descendants are used as the premise of exactly one inference in π.

Suppose $\Diamond\Box\neg x_{\geq 1}$ has $f(m) = \Omega(m)$ descendants in π. Let $D_0 = \Diamond\Box x_{\geq 1}$ and let $D_1, \ldots, D_{f(m)}$ be the $f(m)$ descendants of $\Diamond\Box\neg x_{\geq 1}$ contained within π, indexed by the order in which they are derived. As θ_m is minimally unsatisfiable the $\mathbf{RK}_n$ clause $\Diamond\Box\neg x_{\geq 1}$ must be an ancestor of 0. Further as $\Diamond\Box\neg x_{\geq 1}$ and each of its descendants are used as the premise of exactly one inference in π it follows that each D_i is inferred from D_{i-1} (and possibly also some other $\mathbf{RK}_n$ clause), and that $D_{f(m)} = 0$.

By Lemma 7.4.2 each of the first $f(m) - 1$ descendants of $\Diamond\Box\neg x_{\geq 1}$ are of the form $\Diamond E$. For each $i \in \{0, \ldots, f(m)\}$ let $D_i = \Diamond E_i$. By definition the final non-simplification RFCR used to derive the resolvent of any clause of the form $\Diamond E_i$ is either the Σ $\Box\Diamond$-rule, the Σ $\Diamond\Box$-rule, the Γ $\Diamond 1$-rule or the Γ $\Diamond 2$-rule. Each of these rules ensures that the resolvent computed is of the form $\Diamond(E_{i-1} \wedge C_i)$. Hence whenever $w(C_i) \geq 1$ we have $w(D_i) \geq w(D_{i-1}) + 1$. Further $w(C_i) < 1$ if and only if $C_i = 0$, however if $C_i = 0$ then $D_i = 0$ and so for all $i < f(m)$ we have $w(C_i) > 0$. Hence as $w(D_0) = 1$ we have $w(D_{f(m)-1}) \geq f(m)$ and so $w(\pi) \geq f(m) = \Omega(m)$.

Now suppose $\Diamond\Box\neg x_{\geq 1}$ has $g(m) \neq \Omega(m)$ descendants in π. If $g(m) \geq cm$ for any m then it follows from the above argument that $w(\pi) \geq cm$. Hence we assume without loss of generality that $g(m) < m$. Further let $D_0 = \Diamond\Box\neg x_{\geq 1}$ and let $D_1, \ldots, D_{g(m)}$ be the $g(m)$ descendants of $\Diamond\Box\neg x_{\geq 1}$. Then as in the previous case $D_{g(m)} = 0$ and for each $i \in [g(m)]$ the $\mathbf{RK}_n$ clause D_i must be inferred from D_{i-1}. For each i if D_i is inferred using D_{i-1} and some other $\mathbf{RK}_n$ clause then let this $\mathbf{RK}_n$ clause be denoted by A_i. Further let $g'(m) \leq g(m)$ be the number of such $\mathbf{RK}_n$ clauses.

Recall from Remark 7.4.1 that every initial $\mathbf{RK}_n$ clause in θ_m is essential to π. Let θ'_m denote the $\mathbf{RK}_n$ CNF:

$$\bigwedge_{i=1}^{m} \Box\Diamond\left(x_{\geq 1} \vee \neg x_i\right) \wedge \Box\left(\Box x_1 \vee x'_1\right) \wedge \bigwedge_{i=2}^{m-1} \Box\left(\neg x'_{i-1} \vee \Box x_i \vee x'_i\right) \wedge \Box\left(\neg x'_{m-1} \vee \Box x_m\right),$$

which is obtained from θ_m by removing $\Diamond\Box\neg x_{\geq 1}$. For each A_i there exists some sub-derivation π_{A_i} of A_i from θ'_m. The only $\mathbf{RK}_n$ clause in θ_m which can be essential to π without being essential to some A_i is $\Diamond\Box\neg x_{\geq 1}$. It follows that each of the $\mathbf{RK}_n$ clauses in θ'_m is essential to some π_{A_i}. For each i let r_i denote the number of initial clauses which are essential π_{A_i}. Then $\sum_{j=1}^{g'(m)} r_j \geq 2m$.

Claim. $w(A_i) \geq r_i$.

As each $\mathbf{RK}_n$ clause in θ'_m contains a unique literal of the form x_i or $\neg x_i$ this claim follows immediately from Lemma 7.4.4, which is given at the end of this section. Further $w(D_i) \geq w(D_{i-1}) + w(\pi_{A_i}) - 1$ for every i such that A_i is defined and $w(D_i) \geq 2 \times w(D_{i-1}) - 1$ for every

other $i \in [g(m)]$. Hence $w(\pi) \geq 2m - g'(m)$. Further as $g'(m) \leq g(m)$ and we have assumed that $g(m) < m$ it follows that $w(\pi) > m$. □

We conclude the subsection by proving that every $\mathbf{RK}_n$ clause that can be derived from θ'_m is of a certain form, and crucially for the claim in the above proof, that if at least k clauses in θ'_m are essential to the derivation of any such clause then this clause has width at least k.

Lemma 7.4.4. Let θ'_m denote the $\mathbf{RK}_n$ CNF which is obtained from θ_m by removing $\Diamond\Box\neg x_{\geq 1}$. Then every $\mathbf{RK}_n$ clause C which is derivable from θ'_m must be of the form:

$$\Box\left(\bigvee_{i=1}^{z} \Diamond A_i \vee A' \vee \bigvee_{i_1 \in I_1} x'_{i_1} \vee \bigvee_{i_2 \in I_2} \neg x'_{i_2}\right),$$

where $z \in \mathbb{N}$, each $I_j \subseteq [m]$, each A_{i_1} is an $\mathbf{RK}_n$ clause and A' is an $\mathbf{RK}_n$ clause. In particular each $A_{i_1} = \bigwedge_{j_1 \in [z_{i_1}]} A^k_{i_1,j_1}$, where $z_{i_1} \in \mathbb{N}$, $k \in [m]$ and for each $j_1 \in [z_{i_1}]$ either:

- $A^k_{i_1,j_1} = x_{\geq 1}$,
- $A^k_{i_1,j_1} = x_{\geq 1} \vee \bigvee_{j \in J} x_j$,
- $A^k_{i_1,j_1} = x_{\geq 1} \vee \neg x_k \vee \bigvee_{j \in J} x_j$,
- $A^k_{i_1,j_1} = \neg x_k \vee \bigvee_{j \in J} x_j$,
- $A^k_{i_1,j_1} = x_{\geq 1} \vee \neg x_k$,
- or, $A^k_{i_1,j_1} = \bigvee_{j \in J} x_j$,

where J is a non-empty subset of $[m]$. Further A' is a subformula of:

$$\bigvee_{I \subseteq [m]} \left(\Box \bigvee_{j \in I} x_j\right).$$

We refer to $\mathbf{RK}_n$ clauses of the same form as C as *well-structured clauses*.

Furthermore if π is an $\mathbf{RK}_n$ derivation of some $\mathbf{RK}_n$ clause C from θ'_m and some initial $\mathbf{RK}_n$ clause which contains a variable x_i, where $i \in m$, is essential to π then the variable x_i must also appear in C.

Proof. For each $k \in [m]$ we refer to the $A^k_{i_1,j_1}$'s as k-diamond clauses. We further refer to A' as a box clause.

In the proof we will make use of the following facts, each of which follows immediately from the definition of a well-structured clause.

Fact 1. If $\Box C_1$ and $\Box C_2$ are well-structured clauses then so is the normal form of $\Box(C_1 \vee C_2)$.

Fact 2. If A' and B' are box clauses then so is the normal form of $A' \vee B'$.

Fact 3. If $A^k_{i,j}$ is a k-diamond clause then so is every subclause of $A^k_{i,j}$ apart from $\neg x_k$.

Fact 4. If A^k_{i,j_1} is a k-diamond clause then so are the normal forms of $A^k_{i,j_1} \vee \neg x_k$ and $A^k_{i,j_1} \vee A^k_{i,j_2}$, where A^k_{i,j_2} is a k-diamond clause.

Let r denote the length of π. The proof is by induction on r. Suppose $r = 1$. Then $C \in \theta'_m$. It is clear from inspection that every $\mathbf{RK}_n$ clause in θ'_m is of the desired form. Further the the only clause which is essential to π is C.

Now suppose $r > 1$. Then either C is inferred from a single $\mathbf{RK}_n$ clause C_1, or C is inferred from a pair of $\mathbf{RK}_n$ clauses C_1 and C_2. Further π must contain a sub-derivation π_i, of C_i for each i. Each of these sub-derivations has length strictly less than r and so it follows by the inductive hypothesis that C_1 and C_2 are well-structured clauses. That is:

$$C_1 = \Box\left(\bigvee_{i_1 \in I_1} \Diamond A_{i_1} \vee A' \vee \bigvee_{i_2 \in I_2} x'_{i_2} \vee \bigvee_{i_3 \in I_3} \neg x'_{i_3}\right),$$

$$\text{and} \quad C_2 = \Box\left(\bigvee_{i'_1 \in I'_1} \Diamond B_{i'_1} \vee B' \vee \bigvee_{i'_2 \in I'_2} x'_{i'_2} \vee \bigvee_{i'_3 \in I'_3} \neg x'_{i'_3}\right),$$

where $I_1 = [z_1]$ for some $z_1 \in \mathbb{N}$ (respectively $I'_1 = [z'_1]$ for some $z'_1 \in \mathbb{N}$), I_2 and I_3 (respectively I'_2 and I'_3) are subsets of $[m]$, each A_{i_1} (respectively B_{i_3}) is a conjunction of k-diamond clauses for some $k \in [m]$ and A' (respectively B') is a box clause. Further every x_i variable which appears in an $\mathbf{RK}_n$ clause in θ'_m which is essential to π_1 (respectively π_2) must also appear in C_1 (respectively C_2).

Suppose C is inferred from just C_1. Then by definition, every $\mathbf{RK}_n$ clause in θ'_m which is essential to π must also be essential to π_1. Hence it suffices to show that C is a well-structured clause and that every literal x_i or $\neg x_i$ which appears in C_1 also appears in C. We have two cases:

1. C is inferred by resolution. It follows from the rules of $\mathbf{RK}_n$ that C is inferred by resolving on two pivots that occur within the same diamond operator. Hence we must resolve on some variable x_k which appears positively within some k-diamond clause $A^k_{i_1,j_1} = x_k \vee D_1$ and negatively within some k-diamond clause $A^k_{i_1,j_2} = \neg x_k \vee D_2$, where D_1 is either a k-diamond clause or $\neg x_k$, and D_2 is a k-diamond clause. The corresponding sequence of FFCR must begin as follows:

$$\begin{aligned}
&\Sigma(x_k,\ \neg x_k) \to 0, && \mathbf{A1},\\
&\Sigma(A^k_{i_1,j_1},\ A^k_{i_1,j_2}) \to D_1 \vee D_2, && \Sigma\ \vee\text{-rule},\ \mathbf{S2},\\
&\Gamma(\Diamond A_{i_1}) \to \Diamond\left((D_1 \vee D_2) \wedge A_{i_1}\right), && \Gamma\ \Diamond\text{1-rule},\\
&\Gamma(C_1) \to \Box\Big(\Diamond\left((D_1 \vee D_2) \wedge A_{i_1}\right) \vee && \Gamma\ \vee\text{-rule and}\\
&\qquad \textstyle\bigvee_{i_4 \in I_4} \Diamond A_{i_4} \vee A' \vee \bigvee_{i_2 \in I_2} x'_{i_2} \vee \bigvee_{i_3 \in I_3} \neg x'_{i_3}\Big), && \Gamma\ \Box\text{-rule},
\end{aligned}$$

   where $I_4 = I_1 \setminus \{i_1\}$. The inference is completed by applying some simplification rules[2] followed by the inference rule $\mathbf{R1}$.

[2]Note that in the above sequence of FFCR for notational convenience we have applied a simplification rule before the last non-simplification RFCR, contrary to our usual convention.

As D_2 is a k-diamond clause and D_1 is either a k-diamond clause or $\neg x_k$ it follows by Fact 4 that the normal form of $D_1 \vee D_2$ is also a k-diamond clause. Hence C must be a well-structured clause. Further, every literal in C_1 also appears in the last computed clause in our above partial sequence of FFCR. Note that the simplification rule **S3** is the only rule of $\mathbf{RK}_n$ which can be used to remove literals from an $\mathbf{RK}_n$ clause[3]. The normal form of $(D_1 \vee D_2) \wedge A_{i_1}$ is cannot be 0 and so **S3** cannot be used in any sequence of FFCR which is used to compute C. Hence every literal in C_1 must also appear in C.

2. Suppose C is inferred by weakening. This must be done within some diamond operator. Hence C_1 must be weakened by itself less some $\mathbf{RK}_n$ clause D. Further as C is inferred by single clause weakening D must be such that for some diamond clause $A^k_{i_1,j_1}$ we have $D \vee D' = A^k_{i_1,j_1}$, for some $\mathbf{RK}_n$ clause D'. The associated sequence of FFCRs must begin as follows:

$$\begin{aligned}
&\Sigma(0,\ D) \to 0, && \mathbf{A2},\\
&\Sigma(A^k_{i_1,j_2},\ D \vee D') \to A^k_{i_1,j_2} \vee D', && \Sigma \vee\text{-rule, } \mathbf{S2},\\
&\Gamma(\Diamond A_{i_1}) \to \Diamond\left((A^k_{i_1,j_2} \vee D') \wedge A_{i_1}\right), && \Gamma\ \Diamond\text{1-rule},\\
&\Gamma(C_1) \to \Box\Big(\Diamond\left((A^k_{i_1,j_2} \vee D') \wedge A_{i_1}\right) \vee && \Gamma \vee\text{-rule and}\\
&\qquad \textstyle\bigvee_{i_4 \in I_4} \Diamond A_{i_4} \vee A' \vee \bigvee_{i_2 \in I_2} x'_{i_2} \vee \bigvee_{i_3 \in I_3} \neg x'_{i_3}\Big), && \Gamma\ \Box\text{-rule},
\end{aligned}$$

where $j_2 \neq j_1$ and $I_4 = I_1 \setminus \{i_1\}$. The sequence can be completed by applying some simplification rules followed by the inference rule $\mathbf{R1}$.

As D' is a subclause of the k-diamond clause $A^k_{i_1,j_1}$ and $A^k_{i_1,j_2}$ is a k-diamond clause it follows by Facts 3 and 4 that the normal form of $A_{k,j_2} \vee D'$ is also a k-diamond clause. Hence C is a well-structured clause. Further by the same reasoning as in case 1 above, every literal which appears in C_1 must also appear in C.

Suppose C is inferred from a pair of $\mathbf{RK}_n$ clauses C_1 and C_2. Then we have two cases:

1. C is inferred by resolution. Then every $\mathbf{RK}_n$ clause in θ'_m which is essential to C must also be essential to either C_1 or C_2. Hence it suffices to show that C is well-structured and that every x_i variable which appears in either C_1 or C_2 appears in C.

   We consider three subcases:

   (a) C is inferred by resolving some box operator with some diamond operator. We assume without loss of generality that the box operator resolved on is in C_1 and the diamond operator is in C_2. It follows that the box operator is contained within A' and that the diamond operator is within some $B_{i'_1}$. Hence the associated sequence of FFCRs must

[3]The rule **S4** removes only repetitions of literals.

begin as follows:

$$\begin{aligned}
&\Sigma(0,\ D) \to 0, && \mathbf{A2},\\
&\Sigma(\textstyle\bigvee_{i\in I} x_i,\ D \vee D') \to \bigvee_{i\in I} x_i \vee D', && \Sigma\ \vee\text{-rule},\\
&\Sigma\left(\Box \textstyle\bigvee_{i\in I} x_i,\ \Diamond B_{i'_1}\right) \to \Diamond\left(B_{i'_1} \wedge \left(\textstyle\bigvee_{i\in I} x_i \vee D'\right)\right), && \Sigma\,\Box\Diamond\text{-rule},\\
&\Sigma\,(C_1, C_2) \to \Box\Big(\Diamond\left(B_{i'_1} \wedge \left(\textstyle\bigvee_{i\in I} x_i \vee D'\right)\right) \vee \bigvee_{i_1\in I_1} \Diamond A_{i_1} \vee && \Sigma\ \vee\text{-rule, } \mathbf{S4}\\
&\qquad \textstyle\bigvee_{i'_4\in I'_4} \Diamond B_{i'_4} \vee A'' \vee B' \vee \bigvee_{i_4\in I_4} x'_{i_4} \vee \bigvee_{i_5\in I_5} \neg x'_{i_5}\Big), && \text{and } \Sigma\,\Box\Box\text{-rule},
\end{aligned}$$

where $I'_4 = I'_1 \setminus \{i'_1\}$, $I_4 = (I_2 \cup I'_2)$ and $I_5 = (I_3 \cup I'_3)$. Further A'' is such that $A' = A'' \vee \Box\left(\bigvee_{i\in I} x_i\right)$ and D and D' are such that $D \vee D' = B^k_{i_1,j_1}$ for some j_1. The inference can be completed by applying all possible simplification rules followed by the inference rule $\mathbf{R2}$.

The subformula A'' is a box clause and so it follows by Fact 2 that the normal form of $A'' \vee B'$ is also a box clause. Further by Fact 3 either $D' = \neg x_k$ or D' is a k-diamond clause. In either case the normal form of $\bigvee_{i\in I} x_i \vee D'$ must be a k-diamond clause. Hence C is a well-structured clause. Finally we note that every literal in C_1 and every literal in C_2 is contained within the final computed to clause of the above sequence of FFCRs. As the computed clause contains no subclause of the form $0 \wedge E$, no sequence of FFCRs which can be used to infer C contains a FFCR obtained using $\mathbf{S3}$. It follows that C contains every literal in C_1 and every literal in C_2.

(b) C is inferred by resolving on some x_k. We assume without loss of generality that x_k appears positively in C_1 and negatively in C_2. Then C must be inferred by resolving some k-diamond clause $B^k_{i'_1,j_1} = \neg x_k \vee D_1$ with some subformula of A' of the form $x_k \vee \bigvee_{i\in I} x_i$. The sequence of FFCRs used to infer C must begin as follows:

$$\begin{aligned}
&\Sigma(x_k,\ \neg x_k) \to 0, && \mathbf{A1},\\
&\Sigma\left(x_k \vee \textstyle\bigvee_{i\in I} x_i,\ B^k_{i'_1,j_1}\right) \to \bigvee_{i\in I} x_i \vee D_1, && \Sigma\ \vee\text{-rule},\\
&\Sigma\left(\Box\left(x_k \vee \textstyle\bigvee_{j\in I} x_j\right),\ \Diamond B_{i'_1}\right) \to \Diamond\left(\left(\textstyle\bigvee_{i\in I} x_i \vee D_1\right) \wedge B_{i'}\right), && \Sigma\,\Box\Diamond\text{-rule},\\
&\Sigma(C_1,\ C_2) \to \Box\Big(\Diamond\left(\left(\textstyle\bigvee_{i\in I} x_i \vee D_1\right) \wedge B_{i'_1}\right) \vee \bigvee_{i_1\in I_1} A_{i_1} && \Gamma\ \vee\text{-rule, } \mathbf{S4}\\
&\qquad \vee \textstyle\bigvee_{i'_4\in I'_4} B_{i'_4} \vee A'' \vee B' \vee \bigvee_{i_4\in I_4} x'_{i_4} \vee \bigvee_{i_5\in I_5} \neg x'_{i_5}\Big), && \text{and } \Sigma\,\Box\Box\text{-rule}
\end{aligned}$$

where $I'_4 = I'_1 \setminus \{i'_1\}$, $I_4 = (I_2 \cup I'_2)$ and $I_5 = (I_3 \cup I'_3)$. Further A'' is such that $A' = A'' \vee \Box\left(\bigvee_{i\in I} x_i\right)$. The sequence of FFCR can then be completed by applying further simplification rules to the computed clause followed by the inference rule $\mathbf{R2}$.

As D_1 is a subclause of a k-diamond clause and $D_1 \neq \neg x_k$ it follows by Fact 3 that D_1 is a k-diamond clause. Hence $\bigvee_{i\in I} x_i \vee D_1$ must also be a k-diamond clause. Further as A'' and B' are box clauses it follows by Fact 2 that the normal form of $A'' \vee B'$

must also be a box clause. Hence C is a well-structured clause. The above computed clause contains every literal in C_2 and every literal in C_1 except x_k, however as C_2 contains $\neg x_k$ it follows that the computed clause contains every variable in C_1 and every variable in C_2. Further, as C is the normal form of the final computed clause and the simplification rules used to obtain C cannot include $\mathbf{S3}$, it follows that C contains every variable in C_1 and every variable in C_2.

(c) C is inferred by resolving on some variable x'_i. We assume without loss of generality that x'_i appears positively in C_1 and negatively in C_2. Then the sequence of FFCRs used to infer C must begin as follows:

$$\begin{aligned}
&\Sigma(x'_i,\ \neg x'_i) \to 0, && \mathbf{A1},\\
&\Sigma(C'_1,\ C'_2) \to \textstyle\bigvee_{i_1 \in I_1} A_{i_1} \vee \bigvee_{i'_1 \in I'_1} B_{i'_1} \vee && \Sigma\ \vee\text{-rule},\ \mathbf{S4}\\
&\qquad\qquad A' \vee B' \vee \textstyle\bigvee_{i_4 \in I_4} x'_{i_4} \vee \bigvee_{i_5 \in I_5} \neg x'_{i_5},\\
&\Sigma(C_1,\ C_2) \to \Box\Big(\textstyle\bigvee_{i_1 \in I_1} \Diamond A_{i_1} \vee \bigvee_{i'_1 \in I'_1} \Diamond B_{i'_1} \vee && \Sigma\ \Box\Box\text{-rule}\\
&\qquad\qquad A' \vee B' \vee \textstyle\bigvee_{i_4 \in I_4} x'_{i_4} \vee \bigvee_{i_5 \in I_5} \neg x'_{i_5} \Big),
\end{aligned}$$

where $I_4 = (I_2 \cup I'_2) \setminus \{i\}$, $I_5 = (I_3 \cup I'_3) \setminus \{i\}$ and C_1 and C'_2 are such that $C_1 = \Box C'_1$ and $C_2 = \Box C'_2$ respectively. The sequence of FFCR can then be completed by applying further simplification rules followed by the inference rule $\mathbf{R2}$.

By Fact 2 the normal form of $A' \vee B'$ is a box clause and so the normal form of the final computed clause must be a well-structured clause. Further the simplification rule $\mathbf{S3}$ cannot be used to derived any of the FFCRs used to obtain C. Hence as the final computed clause contains every x_i variable which is contained by in either C_1 or C_2 it follows that C also contains every such variable.

2. C is inferred by weakening. We assume without loss of generality that C_1 is weakened by C_2 less some D. Note that it follows from the definition of weakening that D cannot be nested within any diamond operator in C_2. Further by Definition 7.4.3 every $\mathbf{RK}_n$ clause in θ'_m which is essential to π must also be essential to π_1 and so we only need to show that C is a well-structured clause and that every x_i variable in C_1 is also in C. There are three subcases:

(a) D has modal depth 1. The sequence of FFCRs used to infer C must begin as follows:

$$\begin{aligned}
&\Sigma(0,\ D) \to 0, && \mathbf{A1},\\
&\Sigma\,(C_1,\ C_2) \to \Box\,(C'_1 \vee C''_2) && \Sigma\ \vee\text{-rule}, \Sigma\ \Box\Box\text{-rule},
\end{aligned}$$

where C'_1 is such that $\Box C'_1 = C_1$ and C''_2 is such that $C_2 = \Box(D \vee C''_2)$. The sequence of FFCR can then be completed by applying further simplification rules followed by the inference rule $\mathbf{R2}$.

As C_1 and C_2 are well-structured clauses it follows by Fact 1 that the normal form of $\Box(C_1' \vee C_2'')$ is also a well-structured clause. Further as $\Box(C_1' \vee C_2'')$ contains no subclause of the form $\Diamond(0 \wedge E)$, where E is an $\mathbf{RK}_n$ clause, the simplification rule **S3** cannot be used to obtain any FFCR used to obtain C. Hence as $\Box(C_1' \vee C_2'')$ contains every literal which appears in C_1 it follows that C must also contain every such literal.

(b) D has modal depth 0 and is added to some k-diamond clause $A^k_{i_1,j_1}$. As D cannot be nested within any $\Diamond$ operator in C_2 the $\mathbf{RK}_n$ clause D must be a subformula of the box clause B'. That is, D must be such that $\Box(D \vee D') = \Box \bigvee_{j \in I} x_j$ for some D'. Hence the sequence of FFCRs used to infer C must begin as follows:

$$\begin{aligned}
&\Sigma(0,\ D) \to 0, && \mathbf{A1},\\
&\Sigma(A^k_{i_1,j_1},\ D \vee D') \to A^k_{i_1,j_1} \vee D' && \Sigma\ \vee\text{-rule}, \mathbf{S2},\\
&\Sigma(\Diamond A_{i_1},\ \Box(D \vee D')) \to \Diamond(A_{i_1} \wedge (A^k_{i_1,j_1} \vee D')) && \Sigma\ \Diamond\Box\text{-rule},\\
&\Sigma(C_1,\ C_2) \to \Box\Big(\Diamond(A_{i_1} \wedge (A^k_{i_1,j_1} \vee D')) \vee \textstyle\bigvee_{i_4 \in I_4} \Diamond A_{i_4} \vee && \Sigma\ \vee\text{-rule},\ \mathbf{S4}\\
&\qquad \textstyle\bigvee_{i_1 \in I_1'} \Diamond B'_{i_1} \vee A' \vee B'' \vee \bigvee_{i_5 \in I_5} x'_{i_5} \vee \bigvee_{i_6 \in I_6} \neg x'_{i_6}\Big), && \text{and } \Sigma\ \Box\Box\text{-rule},
\end{aligned}$$

where $I_4 = I_1 \setminus \{i_1\}$, $I_5 = I_2 \cup I_2'$ and $I_6 = I_3 \cup I_3'$. Further B'' is such that $\Box(D \vee D') \vee B'' = B'$. The sequence of FFCR can then be completed by applying further simplification rules followed by the inference rule $\mathbf{R2}$.

The subformula B'' is a box clause and so it follows by Fact 2 that the normal form of $A' \vee B''$ is also a box clause. Further as D' is a disjunction of literals of the form x_i and $A^k_{i_1,j_1}$ is a k-diamond clause the normal form of $A^k_{i_1,j_1} \wedge D'$ must also be a k-diamond clause. Hence C must be a well-structured clause. Every literal in C_1 appears in the final computed clause in the above sequence of FFCRs, and so as this clause also contains no subfomula of the form $\Diamond(0 \wedge E)$, it follows that every literal in C_1 must also appear in C.

(c) D has depth 0 and is added to the box clause A'. As in case (b) D must be such that $\Box(D \vee D') = \Box \bigvee_{j' \in I'} x_{j'}$ for some D'. Hence the sequence of FFCRs used to infer C must begin as follows:

$$\begin{aligned}
&\Sigma(0,\ D) \to 0, && \mathbf{A1},\\
&\Sigma(\textstyle\bigvee_{j \in I} x_j,\ D \vee D') \to \bigvee_{j \in I} x_j \vee D' && \Sigma\ \vee\text{-rule}, \mathbf{S2},\\
&\Sigma(\Box \textstyle\bigvee_{j \in I} x_j,\ \Box(D \vee D')) \to \Box(\bigvee_{j \in I} x_j \vee D') && \Sigma\ \Box\Box\text{-rule},\\
&\Sigma(C_1,\ C_2) \to \Box\Big((\textstyle\bigvee_{j \in I} x_j \vee D') \vee \bigvee_{i_1 \in I_1} \Diamond A_{i_1} \vee && \Sigma\ \vee\text{-rule},\ \mathbf{S4}\\
&\qquad \textstyle\bigvee_{i_1 \in I_1'} \Diamond B'_{i_1} \vee A'' \vee B'' \vee \bigvee_{i_4 \in I_4} x'_{i_4} \vee \bigvee_{i_5 \in I_5} \neg x'_{i_5}\Big), && \text{and } \Sigma\ \Box\Box\text{-rule},
\end{aligned}$$

where $I_4 = I_2 \cup I_2'$ and $I_5 = I_3 \cup I_3'$. Further A'' is such that $\bigvee_{j \in I} x_j \vee A'' = A'$ and B'' is such that $\Box(D \vee D') \vee B'' = B'$. We can complete the sequence of FFCRs by

applying as many simplification rules as possible and the applying the inference rule **R2**.

The subformulas $\Box\left(\bigvee_{j\in I} x_j \vee D'\right)$, A'' and B'' are all box clauses. Hence it follows by Fact 2 that the normal form of $\Box\left(\bigvee_{j\in I} x_j \vee D'\right) \vee A'' \vee B''$ is a box clause. Hence C must be a well-structured clause. Further every literal in C_1 is in the final computed clause in the above sequence of FFCRs. Hence, as this clause contains no subformulas of the form $\Diamond(0 \wedge E)$ it follows that C must also contain very literal in C_1. □

7.4.3 Proving that θ_m does not require large size refutations

In this subsection we show that θ_m has polynomial size $\mathbf{RK}_n$ refutations and then give the proof of the main theorem of the chapter (Theorem 7.4.1).

Theorem 7.4.3. There exist polynomial size $\mathbf{RK}_n$ refutations of θ_m.

Proof. We prove our theorem by constructing a polynomial size $\mathbf{RK}_n$ refutation of θ_m.

Let π_1 denote the following $\mathbf{RK}_n$ derivation of the $\mathbf{RK}_n$ clause $\Box\left(\bigvee_{i=1}^m \Box x_i\right)$ from θ_m.

$$\mathbf{R2}\ \dfrac{\mathbf{R2}\ \dfrac{\mathbf{R2}\ \dfrac{\Box(\Box x_1 \vee x_2') \qquad \Box(\neg x_2' \vee \Box x_2 \vee x_3')}{\Box(\Box x_1 \vee \Box x_2 \vee x_3')} \qquad \Box(\neg x_3' \vee \Box x_4 \vee x_4')}{\vdots} \qquad \Box(\neg x_m' \vee \Box x_m)}{\Box\left(\bigvee_{i=1}^m \Box x_i\right)}$$

Further let π_2 denote the below $\mathbf{RK}_n$ derivation of $\Box\left(\bigvee_{i=1}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)$ from the $\mathbf{RK}_n$ CNF $\theta_m \wedge \Box\left(\bigvee_{i=1}^m \Box x_i\right)$.

$$\begin{array}{l}\mathbf{R2}\\ \mathbf{R2}\end{array}\ \dfrac{\mathbf{R2}\ \dfrac{\mathbf{R2}\ \dfrac{\Box\left(\bigvee_{i=1}^m \Box x_i\right) \qquad \Box\Diamond(x_{\geq 1} \vee \neg x_1)}{\Box\left(\Diamond\left((x_{\geq 1} \vee \neg x_1) \wedge x_{\geq 1}\right) \vee \bigvee_{i=2}^m \Box x_i\right)} \qquad \Box\Diamond(x_{\geq 1} \vee \neg x_2)}{\vdots} \qquad \Box\Diamond(x_{\geq 1} \vee \neg x_m)}{\Box\left(\bigvee_{i=1}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)}$$

Finally let π_3 denote the following $\mathbf{RK}_n$ refutation of $\theta_m \wedge \Box\left(\bigvee_{i=1}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)$.

$$\begin{array}{l}\\ \mathbf{R1}\\ \mathbf{R1}\\ \\ \mathbf{R1}\end{array}\ \dfrac{\begin{array}{l}\mathbf{R2}\ \dfrac{\Box\left(\bigvee_{i=1}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right) \qquad \Diamond\Box\neg x_{\geq 1}}{\Diamond\left(\Box\neg x_{\geq 1} \wedge \left(\bigvee_{i=2}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)\right)}\\ \dfrac{\Diamond\left(\Box\neg x_{\geq 1} \wedge \bigwedge_{j=2}^3 \left(\bigvee_{i=j}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)\right)}{\vdots}\end{array}}{\dfrac{\Diamond\left(\Box\neg x_{\geq 1} \wedge \bigwedge_{j=2}^m \left(\bigvee_{i=j}^m \Diamond\left((x_{\geq 1} \vee \neg x_i) \wedge x_{\geq 1}\right)\right)\right)}{0}}$$

Each of the derivations π_1, π_2 and π_3 are of polynomial size in m. Hence putting the three derivations together we obtain an $\mathbf{RK}_n$ refutation of θ_m with size polynomial in m. □

We conclude with the proof of the main theorem of the section.

Proof of Theorem 7.4.1. Respectively Theorems 7.4.2 and 7.4.3 state that every $\mathbf{RK}_n$ refutation of θ_m has width at least m and that there exist polynomial size $\mathbf{RK}_n$ refutations of θ_m respectively. Hence our theorem follows immediately. □

Chapter 8

Game theoretic lower bound technique

In this chapter we introduce an asymmetric two player game based on those of [13, 14, 76]. This game is played by a Prover and a Delayer, on an unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$. Prover's goal is to construct a countermodel for a certain set of clauses $\mathcal{D} \subseteq \{C \mid \mathcal{C} \vdash_{\mathbf{K}_{mc}\text{-Res}} C\}$. The set $\mathcal{D}$ is defined in such a way as to ensure that it is unsatisfiable if and only if $\mathcal{C}$ is, and so it will always be possible for Prover to construct a countermodel. Hence Delayer's goal is not to prevent Prover from doing so, but to score as many points as possible before the game ends. We show that lower bounds on the proof size required to refute some unsatisfiable set of SNF$_{mc}$ clauses using *tree-like* $\mathbf{K}_{mc}$-Res can be obtained indirectly by showing a lower bound on Delayer's score. In particular such lower bounds are lower bounds on the number of modal proof steps required to refute $\mathcal{C}$.

Before formally defining our two player game we must extend the set of words we use to specify the modal contexts of a given set of SNF$_{mc}$ clauses $\mathcal{C}$. This is because we need to be able to specify the modal context of every literal l that appears in a clause of the form $(e : x \to \Diamond_a l)$. If $l \in \mathcal{X}_\mathcal{C}$ then we can do this using the set of words $\mathcal{E}^*_\mathcal{C}$ (as l has modal context $e(a, l)$), however if $l \notin \mathcal{X}_\mathcal{C}$ then its modal context cannot be described by any word in $\mathcal{E}^*_\mathcal{C}$. Hence we have the following definition.

Definition 8.0.1. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses. We define:

$$\mathcal{L}_{\mathcal{C}-} = \{(x', x) \in \mathcal{L} \times \mathcal{L} \mid (e : x' \to \Diamond_a x) \in \mathcal{C}\},$$
$$\text{and} \quad \bar{\mathcal{E}}_\mathcal{C} = \mathcal{E}_\mathcal{C} \cup (\mathcal{A} \times \mathcal{L}_{\mathcal{C}-}).$$

We say each element of $\bar{\mathcal{E}}_\mathcal{C}$ is a *context marker* for $\mathcal{C}$.

Now if $(e : x \to \Diamond_a l) \in \mathcal{C}$ then $(x, l) \in \mathcal{L}_{\mathcal{C}-}$ and so the modal context of l is given by the word $e(a, (x, l))$. Therefore in this section we use the set of finite words over $\bar{\mathcal{E}}_\mathcal{C}$ to specify the modal contexts of clauses and variables.

We further extend the definition of the unification function σ so that $\sigma : \bar{\mathcal{E}}^*_{\mathcal{C}} \times \cdots \times \bar{\mathcal{E}}^*_{\mathcal{C}} \to \bar{\mathcal{E}}^*_{\mathcal{C}}$ and for $y_1, \ldots, y_n \in \bar{\mathcal{E}}_{\mathcal{C}}$ we have $\sigma(y_1, \ldots, y_n) = (a, (x', x))$ if for some $j \in [n]$ we have $y_j = (a, (x', x))$ and for all $k \neq j$ we have $y_k = a$ or $y_k = (a, (x', x))\}$. We also extend the definition of the reachability of a world (Definition 4.4.4) to $\bar{\mathcal{E}}^*_{\mathcal{C}}$ in the obvious way[1].

The following three definitions give us some convenient notation.

Definition 8.0.2. Let Σ be a set of symbols and let $w \in \Sigma^*$. We say w is a *prefix* of some word $u \in \Sigma^*$ (denoted $w \sqsubseteq u$) if and only if $u = wv$ where $v \in \Sigma^*$. We say w is a *proper prefix* of some word $u \in \Sigma^*$ (denoted $w \sqsubset u$) if and only if w is a prefix of u and $w \neq u$.
We say w is a *suffix* of some word $u \in \Sigma^*$ (denoted $w \sqsupseteq u$) if $u = vw$ where $v \in \Sigma^*$. We say w is a *proper suffix* of some word $u \in \Sigma^*$ (denoted $w \sqsupset u$) if w is a suffix of u and $w \neq u$.

We say u is a *subword* of w (denoted $u \lhd w$) if $w = w_1 u w_2$ for some $w_1, w_2 \in \Sigma^*$.

Definition 8.0.3. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses and let $e \in \bar{\mathcal{E}}^*_{\mathcal{C}}$. We define:

$$\bar{\mathcal{E}}_{e\sqsubset} = \{e' \in \bar{\mathcal{E}}^*_{\mathcal{C}} \mid \sigma(e, e'') \in \bar{\mathcal{E}}^*_{\mathcal{C}} \text{ and } e'' \sqsubset e'\}.$$

The sets $\bar{\mathcal{E}}_{e\sqsupset}$, $\bar{\mathcal{E}}_{e\sqsubseteq}$, $\bar{\mathcal{E}}_{e\sqsupseteq}$, $\bar{\mathcal{E}}_{e\not\sqsubseteq}$, $\bar{\mathcal{E}}_{e\not\sqsupseteq}$ and $\bar{\mathcal{E}}_{e=}$ are defined similarly.

Definition 8.0.4. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses. For each $e \in \bar{\mathcal{E}}^*_{\mathcal{C}}$ we define:

$$L_e = \{(e' : C) \in \mathcal{C} \mid C \in \mathcal{CL} \text{ and } \sigma(e, e') \in \bar{\mathcal{E}}^*_{\mathcal{C}}\},$$
$$N_e = \{(e' : x' \to \Diamond_a x) \in \mathcal{C} \mid \sigma(e, e') \in \bar{\mathcal{E}}^*_{\mathcal{C}}\},$$
$$\mathcal{C}_e = L_e \cup \{(e' : x' \to \circ_a x) \in \mathcal{C} \mid \sigma(e, e'') \in \bar{\mathcal{E}}^*_{\mathcal{C}} \text{ where } e'' \text{ is the modal context of } x\}.$$

Then the set L_e consists of all literal clauses in $\mathcal{C}$ whose modal context is unifiable with e and the set N_e is the set of all negative modal clauses in $\mathcal{C}$ whose modal context is unifiable with e. The set $\mathcal{C}_e$ is the set of all clauses to which a rule of $\mathbf{K}_{mc}$-Res can be applied to resolve on some variable whose modal context is unifiable with e (not to be confused with the set of all clauses whose modal context is unifiable with e).

8.1 Query sets

Several different Prover-Delayer games have been used to prove lower bounds for tree-like propositional resolution (cf. [13,14,76]). Such games are played over an unsatisfiable propositional formula ϕ in CNF. Over the course of a game on ϕ Prover and Delayer build a propositional countermodel for ϕ (that is, a partial assignment α to the variables in ϕ such that for some propositional clause $C \in \phi$ we have $\alpha(C) = 0$). At each round Prover queries some as yet unassigned variable in ϕ and α is extended to include an assignment for this variable. The game ends when $\alpha(C) = 0$ for some propositional clause C in ϕ.

[1] So for example given a model M we say a world w is $(a, (x', x)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}^-}$-reachable from a world u if $(x' \to \Diamond_a x) \in \mathcal{C}$, the valuation $V(u)(x') = V(w)(x) = 1$ and $(u, w) \in R_a$.

Similarly over the course of a modal game (as defined in Section 8.2) played on an unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$, Prover and Delayer build a pointed countermodel[2] $\langle M, w_\varepsilon \rangle$ for some set of clauses $\mathcal{D} \subseteq \mathcal{C} \cup \{D \mid \mathcal{C} \vdash_{\mathbf{K}_{mc}\text{-Res}} D\}$. The exact definition of $\mathcal{D}$ is given in Section 8.2, however for now it suffices to note that $\mathcal{D}$ is unsatisfiable if and only if $\mathcal{C}$ is unsatisfiable. At the start of any such game Prover and Delayer have a model consisting of a single world w_ε. New worlds are added to this model at each round of the game. Hence the key difference between the previously proposed propositional games and our modal game is that at each round Prover queries a world in the current model, instead of a variable in $\mathcal{C}$. Querying a world w essentially means asking whether or not to add a new world w' which is reachable from w to the model and if so for which context marker $b \in \bar{\mathcal{E}}_\mathcal{C}$ is w' b-reachable from w. If at a given round no world is added to the model then the game ends.

Now, suppose some pointed model $\langle M, w_\varepsilon \rangle$ is a countermodel for a set of SNF$_{mc}$ clauses $\mathcal{C}$. Then there must exist some $C \in \mathcal{C}$ for which $(M, w_\varepsilon) \not\models C$. If C is a negative model clause then $C = (e : l \rightarrow \Diamond_a l')$ for some $e \in \bar{\mathcal{E}}^*_\mathcal{C}$, $l, l' \in \mathcal{L}$ (respectively $l \in \mathcal{L}$, $l' \in \mathcal{X}_\mathcal{C}$) and $a \in \mathcal{A}$. Hence as $(M, w_\varepsilon) \not\models C$, the model M must contain a world w which is e-reachable from w_ε and for which $V(w)(l) = 1$, but no world w' that is $(a, (l, l'))$-reachable (respectively (a, l')-reachable) from w and so we say that $\langle M, w_\varepsilon \rangle$ *modally falsifies* C.

Otherwise C is either a positive modal clause or a literal clause. In either case M must fail to satisfy C because of its valuation functions[3] and so we say that $\langle M, w_\varepsilon \rangle$ *propositionally falsifies* C.

If a model $\langle M, w_\varepsilon \rangle$ modally falsifies some clause $C = (e : l \rightarrow \Diamond_a l')$, where $e \in \bar{\mathcal{E}}^*_\mathcal{C}$, $l \in \mathcal{X}_\mathcal{C}$ and $l' \in \mathcal{L} \setminus \mathcal{X}_\mathcal{C}$ (respectively $l' \in \mathcal{X}_\mathcal{C}$) then we can obtain a new model which satisfies C by adding a new world w' which is $e(a, (l, l'))$-reachable (respectively $e(a, l')$-reachable) from w_ε. Whereas if $\langle M, w_\varepsilon \rangle$ propositionally falsifies some clause C then no extension of M can possibly satisfy C. Given this it is natural to require that the countermodel for $\mathcal{D} \subseteq \mathcal{C} \cup \{D \mid \mathcal{C} \vdash_{\mathbf{K}_{mc}\text{-Res}} D\}$ built over the course of a modal game on $\mathcal{C}$ propositionally falsifies some clause $C \in \mathcal{D}$.

Recall that our modal game ends at a round where some world w is queried only if Prover chooses not to add a new world to the model. Hence we add the condition that after querying a world w Prover may only choose not to add a world to the model $\langle M, w_\varepsilon \rangle$ if this model already propositionally falsifies some $C \in \mathcal{D}$. We shall see in Section 8.2 that the exact definition of $\mathcal{D}$ depends on the worlds queried in the previous rounds of the game. Furthermore $\mathcal{D}$ is defined so that every negative modal clause in $\mathcal{D}$ is satisfied by $\langle M, w_\varepsilon \rangle$ and so $\mathcal{D}$ is propositionally falsified by $\langle M, w_\varepsilon \rangle$ whenever $(M, w_\varepsilon) \not\models \mathcal{D}$.

Finally, to ensure that the game always terminates we require that every new world added to the model is b-reachable, for some $b \in \bar{\mathcal{E}}_\mathcal{C} \setminus \mathcal{A}$. Note that this ensures that each new world corresponds to some negative modal clause in $\mathcal{C}$, preventing Prover and Delayer from adding new worlds to the

[2]Recall from Definition 2.2.8 that a pointed model is a model with some distinguished world at which formulas are evaluated.

[3]If C is a positive modal clause then $C = (e : l \rightarrow \Box_a l')$ and so M must contain some world w which is e-reachable from w_ε and for which $V(w)(l) = 1$, and some world w' that is a-reachable from w and for which $V(w')(l') = 0$. Similarly if C is a literal clause then $C = (e : l_1 \vee \cdots \vee l_z)$ and so M must contain a world w which is e-reachable from w_ε and for which $V(w)(l_1) = \cdots = V(w)(l_z) = 1$.

model which tell us nothing about the satisfiability of $\mathcal{C}$.

We formalise these restrictions by requiring that whenever Prover queries a world w that is e-reachable from the root world w_ε, she must also query some *query set* for e with respect to $\mathcal{D}$ (Definition 8.1.1). A query set is a set of context markers. We say a clause is *modally inferable* if it can be inferred using some modal rule (i.e. any rule other than LRES) of $\mathbf{K}_{mc}$-Res. The exact definition of a query set for e with respect to $\mathcal{D}$ depends on the set of modally inferable clauses with modal context e which can be derived from $\mathcal{D}$. We define the *set of modally inferable clauses* for some modal context eb with respect to the set $\mathcal{D}$, where $e \in \bar{\mathcal{E}}^*_{\mathcal{D}}$ and $b \in \bar{\mathcal{E}}_{\mathcal{D}}$ to be the set $\mathcal{M}_{eb}$ of all clauses that can be inferred by applying some modal rule to some set of clauses $\mathcal{D}'$ such that for every $D \in \mathcal{D}'$ either $D \in \mathcal{D}_{eb}$ or $\bigcup_{e_1 \in \bar{\mathcal{E}}_{eb\sqsubseteq}} \mathcal{D}_{e_1} \vdash_{\mathbf{K}_{mc}\text{-Res}} D$ (recall from Definition 8.0.4 that for any set of SNF$_{mc}$ clauses and any modal context e' the set $\mathcal{D}'_{e'}$ is the subset of $\mathcal{D}'$ consisting of all clauses which contain a variable with modal context e' that can be resolved on.). Note that the set of all modally inferable clauses with modal context e must be a subset of $\bigcup_{b \in \bar{\mathcal{E}}_{\mathcal{D}}} \mathcal{M}_{eb}$.

In our game we allow Prover to choose not to add any world to the model at a given round only if she has queried the empty set. Otherwise, Prover must add a world w' that is b-reachable from w to the model, where b is some element of the query set.

Definition 8.1.1. Let $\mathcal{C}$ be an unsatisfiable set of SNF$_{mc}$ clauses and let $e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$. We say that a set Q_e is a *query set* for e with respect to $\mathcal{C}$ if and only if it satisfies the following constraints:

(a) $Q_e \subseteq \{(a,(x_1,x_2)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-} \mid (e' : x_1 \to \Diamond_a x_2) \in N_e\} \cup$
$\{(a,x_3) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-} \mid (e' : x_4 \to \Diamond_a x_3) \in N_e\}.$

(b) For every model $M = (W, R_{a_1}, \ldots, R_{a_n}, V)$ and every world $w \in W$ either M contains no world that is e-reachable from w or:

$$(M,w) \not\models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\sqsupseteq}} \mathcal{C}_{e_1} \cup \bigcup_{b \in Q_e} \mathcal{M}_{eb}.$$

Consider the unsatisfiable formula $\Diamond_a(x \wedge \neg x) \wedge (\Diamond_a y \vee \Diamond_a z)$. The corresponding set of SNF$_{mc}$ clauses is:

$$\mathcal{C} = \{(\varepsilon : x_\varepsilon),\ (\varepsilon : x_\varepsilon \to \Diamond_a x_1),\ ((a,x_1) : \neg x_1 \vee x),\ ((a,x_1) : \neg x_1 \vee \neg x),$$
$$(\varepsilon : \neg x_\varepsilon \vee x_2 \vee x_3),\ (\varepsilon : x_2 \to \Diamond_a y),\ (\varepsilon : x_3 \to \Diamond_a z)\}.$$

It is not hard to see that every unsatisfiable subset of $\mathcal{C}$ must be a superset of $\mathcal{C}_{(a,x_1)}$. Hence $\{(a,x_1)\}$, $\{(a,x_1),(a,(x_2,y))\}$, $\{(a,x_1),(a,(x_2,y)),(a,(x_3,z))\}$, and $\{(a,x_1),(a,(x_3,z))\}$ are all query sets for ε with respect to $\mathcal{C}$.

Further, any model that satisfies:

$$\mathcal{C}_\varepsilon \cup N_\varepsilon = \{(\varepsilon : x_\varepsilon), (\varepsilon : x_\varepsilon \to \Diamond_a x_1), (\varepsilon : \neg x_\varepsilon \vee x_2 \vee x_3), (\varepsilon : x_2 \to \Diamond_a y), (\varepsilon : x_3 \to \Diamond_a z)\},$$

at some world w_ε must also contain a world w which is (a, x_1)-reachable from w_ε. Hence as every query set, Q_ε for ε with respect to $\mathcal{C}$ contains (a, x_1) the following statement holds:

"every model that satisfies $\mathcal{C}_\varepsilon \cup N_\varepsilon$ must contain a world that is b-reachable from for some $b \in Q_\varepsilon$".

We will see in the following proposition that an analogous statement holds for any modal context e and any set of clauses $\mathcal{C}$. Hence we can think of a query set for e with respect to $\mathcal{C}$ as representing a set of worlds W' such that any model M that could possibly satisfy $\mathcal{C}$ contains some $w \in W'$.

Proposition 8.1.1. Let $\mathcal{C}$ be an unsatisfiable set of SNF$_{mc}$ clauses and let Q_e be a query set for some modal context $e \in \left(\bar{\mathcal{E}}_\mathcal{C} \setminus \mathcal{A}\right)^*$ with respect to $\mathcal{C}$. Further let $M = (W, R_{a_1}, \ldots, R_{a_n}, V)$ and $w_\varepsilon \in W$. If W contains some world w_1 that is e-reachable from w_ε and:

$$(M, w_\varepsilon) \models N_e \cup \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\not\sqsupseteq}} \mathcal{C}_{e_1},$$

then there exists some $w_2 \in W$ that is eb-reachable from w_ε, for some $b \in Q_e$.

Proof. As Q_e is a query set for e and $w_1 \in W$ is e-reachable from w_ε by part (b) of Definition 8.1.1 we have $(M, w_\varepsilon) \not\models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\sqsupseteq}} \mathcal{C}_{e_1} \cup \bigcup_{b \in Q_e} \mathcal{M}_{eb}$. But by assumption $(M, w_\varepsilon) \models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\not\sqsupseteq}} \mathcal{C}_{e_1}$, also $\bigcup_{e_1 \in \bar{\mathcal{E}}_{e\sqsupseteq}} \mathcal{C}_{e_1} \subseteq \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\not\sqsupseteq}} \mathcal{C}_{e_1}$ hence there must exist some $C \in \mathcal{M}_{eb}$ such that $(M, w_\varepsilon) \not\models C$, where $b \in Q_e$. As any such clause is inferred by applying some modal rule of $\mathbf{K}_{mc}$-Res to a set of clauses whose modal contexts are unifiable with eb, the clause C must be of the form $(e' : x_1 \vee \cdots \vee \neg x_z \vee \neg y')$ where $e' \in \bar{\mathcal{E}}_{e=}$ and $y' \in \mathcal{X}_\mathcal{C}$ such that $(e'' : y' \to \Diamond_a y'') \in N_e$ and either $b = (a, (y', y''))$ or $b = (a, y'')$. And so there must exist some $w \in W$ such that $V(w)(y') = 1$ and w is e'-reachable from w_ε. Further as $(M, w) \models N_e$ we have $(M, w_\varepsilon) \models (e'' : y' \to \Diamond_a y'')$ and so $V(w)(\Diamond_a y'') = 1$. That is, there exists some $w_2 \in W$ such that $V(w_2)(y'') = 1$ and $(w, w_2) \in R_a$ and so w_2 is b-reachable from w.

To prove that w_2 is eb-reachable from w_ε we show by contradiction that w is e-reachable from w_ε. Suppose that w is not e-reachable from w_ε, then $e' \neq e$. By Remark 4.4.1 we have $\bigcup_{e_1 \in \bar{\mathcal{E}}_{e'y\sqsubseteq}} \mathcal{C}_{e_1} \vdash_{\mathbf{K}_{mc}\text{-Res}} C$ and so by the strong soundness of $\mathbf{K}_{mc}$-Res we have $(M, w_\varepsilon) \not\models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e'b\sqsubseteq}} \mathcal{C}_{e_1}$. Clearly $\bigcup_{e_1 \in \bar{\mathcal{E}}_{e'b\sqsubseteq}} \mathcal{C}_{e_1} \subseteq \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\not\sqsupseteq}} \mathcal{C}_{e_1}$ and so $(M, w_\varepsilon) \not\models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\not\sqsupseteq}} \mathcal{C}_{e_1}$, contradicting our original assumption. □

8.2 Prover-delayer game

In this section we define our two player game which is played by a Prover (who, for clarity is female) and a Delayer (who is male) on some unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$. Recall that Prover's goal is to construct a countermodel for a given set of clauses $\mathcal{D} \subseteq \{C \mid \mathcal{C} \vdash_{\mathbf{K}_{mc}\text{-Res}} C\}$. Further this model must propositionally falsify some $C \in \mathcal{D}$. The set of clauses $\mathcal{D}$ that Prover is trying to build a countermodel for depends on the modal context of the game and so changes throughout the game.

At the beginning of the game we have a pointed model consisting of a single world with modal context ε and the set of clauses $\mathcal{C}$. Further, Delayer's score is 0 and the modal context of the game is ε. At each round, if the game has modal context e and we have the set of clauses $\mathcal{D}$ then Prover chooses some query set Q_e for e with respect to $\mathcal{D}$. If $Q_e = \emptyset$ then the game ends and Prover wins. Otherwise the round continues with Prover adding a new world with modal context ec to the model, where $c \in Q_e$. Before Prover adds a new world to the model Delayer gives a weight to each $c \in Q_e$. The lower the weight Delayer gives to a particular $c \in Q_e$ the more points he will score if Prover chooses to add a world with modal context ec. At the end of the round Delayer's score and the set of clauses are updated, and the modal context of the game is changed to ec.

Formally a game on some unsatisfiable set of SNF_{mc} clauses $\mathcal{C}$ is played as follows. At the start of the game there exists a pointed model, $\langle M^1, w_\varepsilon \rangle$ where $M^1 = (W^1, R^1_{a_1}, \ldots, R^1_{a_n}, V^1)$, $W^1 = \{w_\varepsilon\}$, $R^1_{a_i} = \emptyset$ for all $i \in [n]$ and $V^1(w_\varepsilon)(x_\varepsilon) = 1$. Further, the games modal context is $e^1 = \varepsilon$, the set $\mathcal{D}^1 = \mathcal{C}$ and Delayer's score is $s^1 = 0$. The ith round of the game is played as follows:

- Prover fixes some query set Q_{e^i} for e^i with respect to $\mathcal{D}^i$.
- If $Q_{e^i} = \emptyset$ then the game ends.
- Otherwise Delayer assigns a weight p_c to each $c \in Q_{e^i}$ so that $\sum_{c \in Q_{e^i}} p_c = 1$.
- Prover picks some $c = (a', z) \in Q_{e^i}$ and the status of the game is updated as follows:

$$e^{i+1} = e^i c, \quad s^{i+1} = s^i + \log\left(\frac{1}{p_c}\right),$$

$$\mathcal{D}^{i+1} = \bigcup_{e \in \bar{\mathcal{E}}_{e^i \sqsupseteq}} \mathcal{D}^i_e \cup \bigcup_{e \in \bar{\mathcal{E}}_{e^{i+1} \sqsubseteq}} \mathcal{C}_e \cup \bigcup_{b \in Q_{e^i} \setminus \{c\}} \mathcal{M}_{e^i b},$$

$$W^{i+1} = W^i \cup \{w_{e^i c}\}, \quad R^{i+1}_a = \begin{cases} R^i_a \cup \{(w_{e^i}, w_{e^i c})\} & \text{if } a = a', \\ R^i_a & \text{otherwise,} \end{cases}$$

$$V^{i+1}(w_{e^i c})(x) = 1 \text{ if either } z = x \text{ or } z = (x', x).$$

Where $\mathcal{D}^i_e$ denotes the subset of $\mathcal{D}^i$ defined as in Definition 8.0.4.

The set $\mathcal{D}^{i+1}$ in the above definition is defined so that it contains only clauses with modal context e' where e' is either unifiable with some prefix of e^i or such that e^i is unifiable with some prefix of e', and is satisfiability equivalent to $\mathcal{D}^i$.

Note that our game can only be played if at each round the modal context e^i and the set of clauses $\mathcal{D}^i$ are such that there exists a query set for e^i with respect to $\mathcal{D}^i$. We will see in Proposition 8.2.1 that this is always the case.

At each round of the game Delayer claims that the subset of $\mathcal{D}^i$ consisting of every clause whose modal context is a prefix of e^i is satisfied by $\langle M^i, w_\varepsilon \rangle$, and that some extension of M^i satisfies $\mathcal{D}^i$.

Prover then picks some query set Q_{e^i} for e^i with respect to $\mathcal{D}^i$ and proceeds in one of two ways. If $Q_{e^i} = \emptyset$ then by definition no model containing a world w which is e^i-reachable from w_ε satisfies the set:

$$\bigcup_{e \in \bar{\mathcal{E}}_{e^i \not\sqsupseteq}} \mathcal{D}^i_e = \bigcup_{e \in \bar{\mathcal{E}}_{e^i \sqsupseteq}} \mathcal{C}_e.$$

As M^i contains such a world no extension of M^i can possibly satisfy $\mathcal{D}^i$ and so Prover sees that Delayer must be lying and ends the game. Note that every negative modal clause in $\bigcup_{e \in \bar{\mathcal{E}}_{e^i \sqsupseteq}} \mathcal{C}_e$ is satisfied by $\langle M^i, w_\varepsilon \rangle$ so M^i must propositionally falsify $\bigcup_{e \in \bar{\mathcal{E}}_{e^i \not\sqsupseteq}} \mathcal{D}^i_e \subseteq \mathcal{D}^i$.

If $Q_{e^i} \neq \emptyset$ then Prover first notes that M^i contains a world w_{e^i} which is e^i-reachable from w_ε. Hence by Proposition 8.1.1 any extension of M^i can only satisfy the set of negative clauses with modal context e^i (that is, the set $N_{e^i} \subseteq \mathcal{D}^i$) at w_ε if it contains a world that is $e^i b$-reachable from w_ε for some $b \in Q_{e^i}$. Hence $\langle M^i, w_\varepsilon \rangle$ is not a model for $\mathcal{D}^i$ and so Prover adds some such world to M^i to create a new model M^{i+1}, which could potentially satisfy N_{e^i}, and so $\mathcal{D}^i$.

In Proposition 8.2.1 we prove that any countermodel for a set $\mathcal{D}^i$ is also a countermodel for $\mathcal{C}$. Hence the model M^k built over the course of some game with exactly k rounds, and every model that extends M^k are countermodels for $\mathcal{C}$. Note that it is not necessarily the case that no previously considered model M^i where $i \in [k-1]$ was a countermodel for $\mathcal{C}$, as the rules of the game do not force Prover to set $Q_{e^i} = \emptyset$ whenever it is a valid query set for e^i. However if Prover wishes to minimise Delayers score she would always choose to set $Q_{e^i} = \emptyset$ at the first opportunity as this ends the game without allowing Delayer to score any more points.

The following proposition ensures that the game can always be played.

Proposition 8.2.1. Let $\mathcal{C}$ be a set of SNF$_{mc}$ clauses. If a game is played on $\mathcal{C}$ then:

(a) For each i, if $(M, w) \not\models \mathcal{D}^i$ then $(M, w) \not\models \mathcal{C}$.

(b) If $\mathcal{C}$ is unsatisfiable then there exists a query set for each $\mathcal{D}^i$.

(c) For each i, the set $\mathcal{D}^i$ is satisfiable if and only if $\mathcal{C}$ is satisfiable.

Proof. (a) As $\mathcal{D}^1 = \mathcal{C}$ it follows by definition that for each i every clause in $\mathcal{D}^i$ is either in $\mathcal{C}$ or is $\mathbf{K}_{mc}$-Res provable from $\mathcal{C}$. It follows immediately from the strong soundness of $\mathbf{K}_{mc}$-Res that for each i, if $(M, w) \not\models \mathcal{D}^i$ then $(M, w) \not\models \mathcal{C}$.

(b) This can be seen by induction on i. If $i = 1$ then $\mathcal{D}^i = \mathcal{C}$. As $\mathcal{C}$ is unsatisfiable and $\mathbf{K}_{mc}$-Res is complete it follow that if we let:

$$Q_\varepsilon = \{(a, (x', x)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-} \mid (\varepsilon : x' \to \Diamond_a x) \in N_\varepsilon\} \cup \{(a, x) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-} \mid (\varepsilon : x'' \to \Diamond_a x) \in N_\varepsilon\}.$$

then $\mathcal{C}_\varepsilon \cup \bigcup_{b \in Q_\varepsilon} \mathcal{M}_b$ is unsatisfiable as it is the set of all clauses with modal context ε that are $\mathbf{K}_{mc}$-Res derivable. Hence Q_ε is a query set for ε with respect to $\mathcal{D}^1$.

If $i > 1$ then:

$$\mathcal{D}^i = \bigcup_{e \in \bar{\mathcal{E}}_{e^{i-1}\sqsupseteq}} \mathcal{D}_e^{i-1} \cup \bigcup_{e \in \bar{\mathcal{E}}_{e^i\sqsubseteq}} \mathcal{C}_e \cup \bigcup_{b \in Q_{e^{i-1}} \setminus \{c\}} \mathcal{M}_{e^{i-1}b},$$

where $Q_{e^{i-1}}$ is a query set for e^{i-1} with respect to $\mathcal{D}^{i-1}$ and $c \in Q_{e^{i-1}}$ such that $e^i = e^{i-1}c$. That $\mathcal{D}^i$ is well defined follows by induction. As $Q_{e^{i-1}}$ is a query set for e^{i-1} the set $\bigcup_{e \in \bar{\mathcal{E}}_{e^{i-1}\sqsupseteq}} \mathcal{D}_e^{i-1} \cup \bigcup_{b \in Q_{e^{i-1}}} \mathcal{M}_{e^{i-1}b}$ must be unsatisfiable. Note that every clause in $\mathcal{M}_{e^{i-1}c}$ must be inferred from some set $\mathcal{C}' \subseteq \mathcal{C}_{e^i} \cup \bigcup_{b \in Q_{e^i}} \mathcal{M}_{e^i b}$, where:

$$Q_{e^i} = \{(a, (x', x)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-} \mid (e' : x' \to \Diamond_a x) \in N_{e^i}\} \cup \{(a, x) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-} \mid (e' : x'' \to \Diamond_a x) \in N_{e^i}\}.$$

It follows by the completeness of $\mathbf{K}_{mc}$-Res that the set:

$$\bigcup_{e \in \bar{\mathcal{E}}_{e^{i-1}\sqsupseteq}} \mathcal{D}_e^{i-1} \cup \bigcup_{b \in Q_{e^{i-1}} \setminus \{c\}} \mathcal{M}_{e^{i-1}b} \cup \mathcal{C}_{e^i} \cup \bigcup_{b \in Q_{e^i}} \mathcal{M}_{e^i b},$$

is unsatisfiable. Hence Q_{e^i} is a query set for e^i with respect to $\mathcal{D}^i$.

(c) This follows from parts (a) and (b).

8.3 Modal decision trees

To use our two player game to obtain modal proof size lower bounds we need to establish a connection between it and the number of modal resolution steps required to refute a formula using tree-like $\mathbf{K}_{mc}$-Res. Hence in this section we introduce *modal decision trees*. The number of vertices in a modal decision tree for some unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$ is connected to both the number of modal resolution steps required to refute $\mathcal{C}$ using tree-like $\mathbf{K}_{mc}$-Res (Proposition 8.3.1) and the Delayer's score in any game over $\mathcal{C}$ (Theorem 8.3.1).

A modal decision tree T for an unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$ is a tree where each vertex is labelled by a modal context $e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ and a set of clauses $\mathcal{D}$, and each edge is labelled by an agent $a \in \mathcal{A}$. Intuitively, we can think of T as a partial Kripke model $(W, R_{a_1}, \ldots, R_{a_n}, V)$ where the set W is the set of vertices of T, the relation R_{a_i} is the set of a_i-edges of T for each $a_i \in \mathcal{A}$, and the partial valuation function V is such that if a vertex in T is labelled by modal context e then that world is e-reachable from the world corresponding to the root of T. If a vertex η of some modal decision tree T is labelled by the modal context e then the children of η must correspond to some query set for e with respect to $\mathcal{D}$.

Definition 8.3.1. A *modal decision tree* for some unsatisfiable set of SNF$_{mc}$ clauses, $\mathcal{C}$ is a tree T where:

1. Each vertex of T is labelled by a unique modal context $e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ and an unsatisfiable set of

SNF$_{mc}$ clauses $\mathcal{D}$. In particular the root is labelled by the modal context ε and the set of clauses $\mathcal{C}$.

2. If two vertices in T are labelled by the modal contexts e_1 and e_2 respectively then there is an a-edge from η_1 to η_2 if and only if $e_2 = e_1(a, z)$ for some $z \in \mathcal{X}_{\mathcal{C}-} \cup \mathcal{L}_{\mathcal{C}-}$.

3. The modal context e and the set of clauses $\mathcal{D}$ labelling each vertex η must be such that the set $Q_e = \{c \in \bar{\mathcal{E}}_{\mathcal{C}} \mid ec \text{ labels some child of } \eta\}$ is a query set for e with respect to $\mathcal{D}$. Further for each $c \in Q_e$, the set of clauses labelling the corresponding child of η is:

$$\mathcal{D}' = \bigcup_{e_1 \in \bar{\mathcal{E}}_{e\sqsupseteq}} \mathcal{D}_{e_1} \cup \bigcup_{e_1 \in \bar{\mathcal{E}}_{ec\sqsubseteq}} \mathcal{C}_{e_1} \cup \bigcup_{b \in Q_e \setminus \{c\}} \mathcal{M}_{eb}.$$

Each path P from the root of the tree to a given vertex specifies a partial Kripke model $M^P = (W^P, R^P_{a_1}, \ldots, R^P_{a_n}, V^P)$, where:

$$W^P = \{w_e \mid e \in \bar{\mathcal{E}}^*_{\mathcal{C}} \text{ labels some } \eta \in P\},$$

for each $i \in [n]$ the set:

$$R^P_{a_i} = \{(\eta_{e_1}, \eta_{e_2}) \in P \mid (\eta_{e_1}, \eta_{e_2}) \text{ is an } a\text{-edge of } T\},$$

and:

$$V^P = \{V^P(w_{ec}) \mid w_{ec} \in W^P\},$$

where:

$$V^P(w_{ec})(x) = 1 \text{ if } c = (a, x) \text{ or } c = (a, (x', x)).$$

It is not hard to see that for each root to leaf path P through T, the partial model M^P corresponds to the model constructed over the course of some two-player game over $\mathcal{C}$. We will further see in Proposition 8.3.1 that every tree-like $\mathbf{K}_{mc}$-Res refutation of some unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$ corresponds to some unique modal decision tree for $\mathcal{C}$. It is not the case however that every modal decision tree for $\mathcal{C}$ corresponds to a unique tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{C}$.

Let $\mathcal{C}$ be an unsatisfiable set of SNF$_{mc}$ clauses and π be a tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{C}$. For every $e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ let π_e denote the set of all clauses C in π such that:

- C has modal context $e' \in \bar{\mathcal{E}}_{e=}$.
- C is inferred using some modal rule of $\mathbf{K}_{mc}$-Res (that is, either MRES, GEN1, GEN2 or GEN3).
- For each $(a, x_2) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-}$ such that $(a, x_2) \lhd e$, the refutation π contains an inference where some descendant of C is resolved with $(e_1 : x_1 \to \Diamond_a x_2)$ where $e_1 \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ is such that $e_1(a, x_2) \sqsubseteq e$.

- For each $(a,(x_1,x_2)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-}$ such that $(a,(x_1,x_2)) \lhd e$, the refutation π contains an inference where some descendant of C is resolved with $(e_1 : x_1 \to \Diamond_a x_2)$ where $e_1 \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ is such that $e_1(a,(x_1,x_2)) \sqsubseteq e$.

Proposition 8.3.1. Let π be a tree-like $\mathbf{K}_{mc}$-Res refutation of some unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$. Then we can construct a unique modal decision tree T that corresponds to π.

Further if we let N be the number of modal resolution steps in π and n be the number of vertices of T then $N \geq n-1$.

Proof. For every $e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ such that π_e is non-empty let:

$$Q_e = \{(a,x_1) \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-} \mid (e' : x_2 \to \Diamond_a x_1) \text{ is used to infer some } C \in \pi_e\} \cup \{(a,(x_3,x_4)) \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-} \mid (e' : x_3 \to \Diamond_a x_4) \text{ is used to infer some } C \in \pi_e\}.$$

Let the vertex set for T be:

$$V(T) = \{\eta_\varepsilon\} \cup \{\eta_{ec} \mid c \in Q_e \text{ for some } e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*\}.$$

Further let each $\eta_e \in V(T)$ be labelled by the modal context e and the set of SNF$_{mc}$ clauses $\mathcal{D}^{\eta_e}$, where:

$$\mathcal{D}^{\eta_e} = \begin{cases} \mathcal{C} & \text{if } e = \varepsilon, \\ \bigcup_{e' \in \bar{\mathcal{E}}_{e_1\sqsupseteq}} \mathcal{D}^{\eta_{e_1}}_{e'} \cup \bigcup_{e' \in \bar{\mathcal{E}}_{e\sqsubseteq}} \mathcal{C}_{e'} \cup \bigcup_{c \in Q_e \setminus \{b\}} \mathcal{M}_{ec} & \text{if } e = e_1 b \text{ for some } e_1 \in \bar{\mathcal{E}}^*_{\mathcal{C}} \text{ and } b \in \bar{\mathcal{E}}_{\mathcal{C}}. \end{cases}$$

Finally for each $a \in \mathcal{A}$ let:

$$E_a(T) = \{(\eta_{e_1}, \eta_{e_1(a,z)}) \mid (a,z) \in Q_{e_1}\},$$

be the set of a-edges in T.

Clearly T is a tree. Further, $|V(T)| - 1 = n - 1 = \sum_{e \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*} |Q_e|$. We will now show that every element of every Q_e corresponds to some unique modal resolution inference. This is true whenever $Q_{e_1} \cap Q_{e_2} = \emptyset$ for all $e_1 \neq e_2 \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$. Hence suppose $Q_{e_1} \cap Q_{e_2} \neq \emptyset$ for some $e_1 \neq e_2 \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$. Then there must exist a negative modal clause which is used to infer some clause $C_1 \in \pi_{e_1}$ and some clause in $C_2 \in \pi_{e_2}$. If $C_1 \neq C_2$ then these inferences must be distinct. Hence suppose $C_1 = C_2$. As $e_1 \neq e_2$ we must have $b_1^j \neq b_2^j$ for some j where b_1^j and b_2^j denote the jth symbols in e_1 and e_2 respectively. By definition this can be the case only if some descendant C_1' of C_1 is resolved with some negative modal clause $(e_1' : x_1 \to \Diamond_{a_1} x_1')$, where $e_1' \in \bar{\mathcal{E}}^*_{\mathcal{C}}$ which is unifiable with the prefix of e_1 with length $j-1$ and x_1, x_1' and a_1 are such that either $(a_1, x_1') = b_1^j$ or $(a_1,(x_1,x_1')) = b_1^j$, and some descendant C_2' of C_2 is resolved with some negative modal clause $(e_2' : x_2 \to \Diamond_{a_2} x_2')$, where $e_2' \in \bar{\mathcal{E}}^*_{\mathcal{C}}$ which is unifiable with the prefix of e_2 with length $j-1$ and x_2, x_2' and a_2 are such that either $(a_2, x_2') = b_2^j$ or $(a_2,(x_2,x_2')) = b_2^j$. As π is tree-like C_1' and C_2' can both be descendants of the same instance of $C_1 = C_2$ only if either C_1' is a descendant of C_2' or

vice versa. As both clauses are inferred using different negative modal clauses with the same modal level this cannot be the case. Hence π must contain separate inferences of C_1 and C_2. It follows that every element of every distinct Q_e must correspond to a unique modal inference, therefore giving us $|V(T)| - 1 \leq N$, and so $n - 1 \leq N$.

To prove that T is a modal decision tree for $\mathcal{C}$ it remains to show that each Q_e is a valid query set for e with respect to $\mathcal{D}^{\eta_e}$. That is, we must show that each Q_e satisfies conditions (a) and (b) of Definition 8.1.1.

Every element of Q_e corresponds to some negative modal clause in $\mathcal{C}$ whose modal context is unifiable with e. Further every negative modal clause in $\mathcal{C}$ with modal context $e_1 \in \bar{\mathcal{E}}_{e=}$ is in $\bigcup_{e' \in \bar{\mathcal{E}}_{e\sqsubseteq}} \mathcal{C}_{e'} \subseteq \mathcal{D}^{\eta_e}$. Hence that (a) is satisfied follows immediately from the definition of Q_e.

To prove that each Q_e also satisfies condition (b) of the definition of a query set we will first prove that the refutation π contains some sub-refutation of the set:

$$\bigcup_{e_1 \in \bar{\mathcal{E}}_{e\sqsupseteq}} \mathcal{D}^{\eta_e}_{e_1} \cup \pi_e.$$

This is done by induction on $|e|$. If $|e| = 0$ then $e = \varepsilon$ and $\bigcup_{e_1 \in \bar{\mathcal{E}}_{\varepsilon\sqsupseteq}} \mathcal{D}^{\eta_\varepsilon}_{e_1} = \mathcal{C}_\varepsilon$. Further π_ε is the set of all clauses in π that are inferred using some modal rule and have modal context ε. Hence π must contain a sub-refutation of:

$$\pi_\varepsilon \cup \mathcal{C}_\varepsilon = \pi_\varepsilon \cup \bigcup_{e_1 \in \bar{\mathcal{E}}_{\varepsilon\sqsupseteq}} \mathcal{D}^{\eta_\varepsilon}_{e_1}.$$

Suppose $|e| > 0$. Then there exists some $e_1 \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})^*$ and some $b \in (\bar{\mathcal{E}}_{\mathcal{C}} \setminus \mathcal{A})$ such that $e = e_1 b$. Further by definition:

$$\mathcal{D}^{\eta_e} = \bigcup_{e' \in \bar{\mathcal{E}}_{e_1\sqsupseteq}} \mathcal{D}^{\eta_{e_1}}_{e'} \cup \bigcup_{e' \in \bar{\mathcal{E}}_{e\sqsubseteq}} \mathcal{C}_{e'} \cup \bigcup_{c \in Q_{e_1} \setminus \{b\}} \mathcal{M}_{e_1 c},$$

where for each $c \in Q_{e_1}$ the set $\mathcal{M}_{e_1 c}$ is the set of modally inferable clauses for $e_1 c$ with respect to $\mathcal{D}^{\eta_{e_1}}$. By the inductive hypothesis π contains a sub-refutation of $\bigcup_{e' \in \bar{\mathcal{E}}_{e_1\sqsupseteq}} \mathcal{D}^{\eta_{e_1}}_{e'} \cup \pi_{e_1}$. Let $\pi^b_{e_1}$ denote the subset of π_{e_1} consisting of every clause inferred using the negative modal clause $(e' : x_1 \to \Diamond_a x_2)$, where x_1, x_2 and a are such that either $b = (a, x_2)$ if $b \in \mathcal{A} \times \mathcal{X}_{\mathcal{C}-}$, or $b = (a, (x_1, x_2))$ if $b \in \mathcal{A} \times \mathcal{L}_{\mathcal{C}-}$. The rules of $\mathbf{K}_{mc}$-Res are such that every clause $C \in \pi^b_{e_1}$ must be inferred from some set of modal clauses whose modal contexts are unifiable with e_1 and possibly also some literal clause whose modal context is unifiable with e. As all modal clauses are initial clauses they must all be in $\mathcal{C}_e$. Suppose some literal clause is also used to infer C. If this clause is initial then it must be in $\mathcal{C}_e$. If this clause is non-initial then π must contain a derivation of this clause from the set of all literal clauses in $\mathcal{C}$ whose modal contexts are unifiable with e and the set of all modally inferred literal clauses in π whose modal contexts are unifiable with e. Every such modally inferred literal clause must be in π_e as it has a descendant in $\pi^b_{e_1}$. Hence π must contain a

refutation of:

$$\bigcup_{e' \in \bar{\mathcal{E}}_{e_1 \sqsupseteq}} \mathcal{D}^{\eta_{e_1}}_{e'} \cup (\pi_{e_1} \setminus \pi^b_{e_1}) \cup \mathcal{C}_e \cup \pi_e.$$

By definition $(\pi_{e_1} \setminus \pi^b_{e_1}) \subseteq \bigcup_{c \in Q_{e_1} \setminus \{b\}} \mathcal{M}_{e_1 c}$ and so this refutation is also a refutation of $\bigcup_{e' \in \bar{\mathcal{E}}_{e \sqsupseteq}} \mathcal{D}^{\eta_e}_{e'} \cup \pi_e$.

To see that the existence of a $\mathbf{K}_{mc}$-Res refutation of $\bigcup_{e' \in \bar{\mathcal{E}}_{e \sqsupseteq}} \mathcal{D}^{\eta_e}_{e'} \cup \pi_e$ implies that condition (b) is satisfied for each Q_e we first note that as $\mathbf{K}_{mc}$-Res is strongly sound and the set $\bigcup_{e' \in \bar{\mathcal{E}}_{e \sqsupseteq}} \mathcal{D}^{\eta_e}_{e'} \cup \pi_e$ must be unsatisfiable. It then follows from the definition of Q_e that $\pi_e \subseteq \bigcup_{c \in Q_e} \mathcal{M}_{ec}$, and so the set:

$$\bigcup_{e' \in \bar{\mathcal{E}}_{e \sqsupseteq}} \mathcal{D}^{\eta_e}_{e'} \cup \bigcup_{c \in Q_e} \mathcal{M}_{ec},$$

must also be unsatisfiable. □

In the next theorem we state the connection between the number of modal resolution steps required to refute a formula using tree-like $\mathbf{K}_{mc}$-Res and our two-player game. This connection will allow us to prove modal proof size lower bounds for tree-like $\mathbf{K}_{mc}$-Res indirectly.

Theorem 8.3.1. Let $\mathcal{C}$ be an unsatisfiable set of clauses in SNF_{mc} and let π be a tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{C}$ with N modal resolution steps. Then there is a Prover strategy such that Delayer scores $s_m \leq \log(N+1)$ modal points. Hence $2^{s_m} \leq N+1$.

Proof. Let T be the unique modal decision tree which corresponds to π. By Proposition 8.3.1 we have $n - 1 \leq N$, where n is the number of vertices in T and N is the number of modal resolution steps in π. Hence if we let $L(T)$ be the set of all leaf vertices of T then $|L(T)| \leq n \leq N+1$.

The modal decision tree T completely specifies Prover's strategy. Recall that each vertex in T is labelled by some modal context and that if a vertex η is labelled by $e \in \bar{\mathcal{E}}^*_{\mathcal{C}}$ and the set $\mathcal{D}$ then the set $Q_e = \{c \in \bar{\mathcal{E}}_{\mathcal{C}} \mid ec \text{ labels some child of } \eta\}$ is a query set for e with respect to $\mathcal{D}$. In particular the root vertex η_ε is labelled by the modal context ε and the set $\mathcal{C}$, and its children correspond to some query set Q_ε for ε with respect to $\mathcal{C}$. At the first round of the game Prover queries the set Q_ε. If $Q_\varepsilon = \emptyset$ then the game ends. Otherwise, Delayer gives each $c \in Q_\varepsilon$ a weight p_c and Prover chooses some $c = (a_c, z) \in Q_\varepsilon$ with probability p_c, sets:

$$e^2 = c, \quad s^2 = s^1 + \log \frac{1}{p_c}, \quad R^2_a = \begin{cases} R^1_a \cup \{(w_\varepsilon, w_c)\} & \text{if } a = a_c, \\ R^1_a & \text{otherwise,} \end{cases}$$

$$W^2 = W^1 \cup \{w_c\} \text{ and } V^2(w_c)(x) = 1 \text{ if } x = z \text{ or } (x', x) = z \text{ for some } x' \in \mathcal{X}_{\mathcal{C}}.$$

and moves along the corresponding edge of T to the vertex labelled by c.

At the next round Prover queries the set Q_c corresponding to the children of this new vertex and proceeds as above. Continuing in this manner will result in a root to leaf path on T. Note that the set of all possible such paths is in bijection with the set of leaves of T.

Let $q_{D,\lambda}$ denote the probability of the game ending at leaf $\lambda \in L(T)$ when played with a fixed Delayer D. Let θ_D be the probability distribution over the leaves of T. If the game ends at leaf λ then D scores exactly $\log \frac{1}{q_{D,\lambda}}$ points.

To see this consider a fixed leaf λ and the unique path P from the root of T to λ. The modal context of the ith vertex in P is e^i. Hence the probability of reaching λ is:

$$q_{D,\lambda} = q_1 q_2 \dots q_m,$$

where q_j is the probability of choosing c_j from Q_{e^j}. The score at the end of the game is:

$$\sum_{j=1}^{m} \log \frac{1}{q_j} = \log \frac{1}{\prod_{i=j}^{m} q_j} = \log \frac{1}{q_{D,\lambda}},$$

and the expected score of the Delayer is:

$$\sum_{\lambda \in L(T)} q_{D,\lambda} \log \frac{1}{q_{D,\lambda}} = H(\theta_D),$$

which is exactly the Shannon entropy of θ_D. The entropy is maximal when the probability distribution considered is the uniform distribution [28], that is when $q_{D,\lambda} = \frac{1}{|L(T)|}$ and so $\sum_{\lambda \in L(T)} \frac{1}{|L(T)|} \log |L(T)| = \log |L(T)|$. Hence as the support of θ_D has size at most $|L(T)|$ it follows that $H(\theta_D) \leq \log |L(T)| \leq \log(N+1)$. □

The above theorem allows us to prove lower bounds on the number of modal resolution steps needed to refute some unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$ using tree-like $\mathbf{K}_{mc}$-Res. Such lower bounds are proved indirectly by first proving a lower bound $f(n)$, where n is the size of $\mathcal{C}$, on the Delayers score for any game played on a given unsatisfiable set of SNF$_{mc}$ clauses $\mathcal{C}$. It follows from the above theorem that $2^{f(n)} - 1$ is a lower bound for the number of modal resolution steps N required to refute $\mathcal{C}$, hence if $2^{f(n)} - 1$ is superpolynomial then we have proved a superpolynomial lower bound for N. In the next chapter we define a new family of hard modal formulas to which our game theoretic lower bound technique can be applied.

Chapter 9

Lower bounds

In this chapter we apply our game theoretic lower bound technique to a new family of hard modal formulas, which we call the modal pigeonhole principle (Definition 9.1.1). By doing this we obtain a new exponential proof size lower bound for tree-like $\mathbf{K}_{mc}$-Res.

9.1 The modal pigeonhole principle

The pigeonhole principle with m pigeons and n pigeonholes states that whenever $m > n$ there is no $1-1$ map from the pigeons to the pigeonholes. This can be formulated as a propositional formula as follows:

$$PHP_n^m = \bigwedge_{i\in[m]} \bigvee_{j\in[n]} p_{i,j} \wedge \bigwedge_{1\leq i<i'\leq m} \bigwedge_{j\in[n]} (\neg p_{i,j} \vee \neg p_{i',j}).$$

Intuitively, the propositional variable $p_{i,j}$ denotes that the ith pigeon is in the jth pigeonhole, hence the above formula says that each pigeon is in a pigeonhole and that no pigeonhole contains more than one pigeon. Clearly whenever $m > n$ the formula PHP_n^m must be unsatisfiable. The propositional pigeonhole principle is known to be hard for propositional resolution [41].

We can formulate the pigeonhole principle as a modal formula with modal depth m over the set of agents $\mathcal{A} = \{a\}$ and the set of variables $\{l_1, \ldots, l_n\}$.

Definition 9.1.1 ($\mathbf{MPHP}_n^m$). Let $P_i = \Box^{i-1}(\bigvee_{j=1}^n \Diamond l_j)$ for every $1 \leq i \leq m$ and $H_{i,i'}^j = \Box^i \neg l_j \vee \Box^{i'} \neg l_j$ for every $1 \leq j \leq n$ and $1 \leq i < i' \leq m$. We define:

$$\mathbf{MPHP}_n^m = \bigwedge_i P_i \wedge \bigwedge_{1\leq i<i'\leq m} \bigwedge_j H_{i,i'}^j.$$

Note that as $\mathbf{MPHP}_n^m$ is a modal formula over just a single agent we have omitted the subscripts

from our modal operators. Further $\Box^i$ denotes i successive $\Box_a$ operators.

Intuitively pigeon i is in pigeonhole j only if $\Diamond^i l_j = 1$, where $\Diamond^i$ denotes i successive $\Diamond_a$ operators. Hence P_i says that the pigeon i occupies at least one pigeonhole and $H^j_{i,i'}$ says that pigeons i and i' cannot both occupy the same hole. Whenever $m > n$ (that is, there are more pigeons than pigeonholes) $\mathbf{MPHP}^m_n$ is unsatisfiable.

We can easily convert $\mathbf{MPHP}^m_n$ into the following set of SNF$_{mc}$ formulas:

$$\mathcal{MPHP}^m_n = x_\varepsilon \wedge \bigwedge_i \hat{P}_i \wedge \bigwedge_{1 \le i < i' \le m} \bigwedge_j \hat{H}^j_{i,i'}$$

where $i, i' \in [m]$, $j \in [n]$ and for all $i_1 \in \{2, \ldots, m\}$, i, i' and j:

$$\hat{P}_1 = (\varepsilon : \neg x_\varepsilon \vee x^1_1 \vee y^1_1) \wedge \bigwedge_{k_1=2}^{n-2} (\varepsilon : \neg x^1_{k_1-1} \vee x^1_{k_1} \vee y^1_{k_1}) \wedge (\varepsilon : \neg x^1_{n-2} \vee y^1_{n-1} \vee y^1_n) \wedge \bigwedge_{k_2=1}^{n} (\varepsilon : y^1_{k_2} \to \Diamond l_{k_2}),$$

$$\hat{P}_{i_1} = (\varepsilon : x_\varepsilon \to \Box z^{i_1}_1) \wedge \bigwedge_{k_1=1}^{i_1-2} \left(a^{k_1} : z^{i_1}_{k_1} \to \Box z^{i_1}_{k_1+1}\right) \wedge$$

$$(a^{i_1-1} : \neg z^{i_1}_{i_1} \vee x^{i_1}_1 \vee y^{i_1}_1) \wedge \bigwedge_{k_2=2}^{n-2} \left(a^{i_1-1} : \neg x^{i_1}_{k_2-1} \vee x^{i_1}_{k_2} \vee y^{i_1}_{k_2}\right) \wedge$$

$$(a^{i_1-1} : \neg x^{i_1}_{n-2} \vee y^{i_1}_{n-1} \vee y^{i_1}_n) \wedge \bigwedge_{k_3=1}^{n} (a^{i_1-1} : y^{i_1}_{k_3} \to \Diamond l_{k_3}),$$

$$\hat{H}^j_{i,i'} = (\varepsilon : \neg x_\varepsilon \vee x^j_{i,i',1} \vee y^j_{i,i',1}) \wedge \bigwedge_{k_1=1}^{i-1} \left(a^{k_1-1} : x^j_{i,i',k_1} \to \Box x^j_{i,i',k_1+1}\right) \wedge$$

$$(a^{i-1} : x^j_{i,i',i} \to \Box \neg l_j) \wedge \bigwedge_{k_2=1}^{i'-1} \left(a^{k_2-1} : y^j_{i,i',k_2} \to \Box y^j_{i,i',k_2+1}\right) \wedge (a^{i'-1} : y^j_{i,i',i'} \to \Box \neg l_j).$$

Note that we have designed $\mathbf{MPHP}^m_n$ so that when translated into SNF$_{mc}$ each negative modal clause corresponds to an assignment of a pigeon to a pigeonhole.

Whilst we have written $\mathcal{MPHP}^m_n$ as a conjunction of clauses, however we can equivalently think of it as a set of clauses. Hence in the remainder of this section we will use set notation.

9.2 An exponential lower bound for the modal pigeonhole principle

In the following theorem we prove that the number of modal resolution steps used in any tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{MPHP}^m_n$ is superpolynomial in n. To obtain this lower bound we show that if the two-player game defined in Section 8.2 is played on $\mathcal{MPHP}^m_n$ then Delayer can play according to a certain strategy which ensures that he always scores at least $\log(n!)$ points, no matter what strategy Prover adopts. Recall that Theorem 8.3.1 states that, if there exists a refutation of

$\mathcal{MPHP}^m_n$ with N modal resolution steps then Prover can ensure that Delayer's score never exceeds $\log(N+1)$ points. Hence our lower bound follows.

Theorem 9.2.1. Every tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{MPHP}^m_n$ has at least $n! - 1$ modal resolution steps.

Proof. Let $\mathcal{C} = \mathcal{MPHP}^m_n$. Suppose a Prover and a Delayer play the game defined in Section 8.2 on $\mathcal{C}$. By definition, at the beginning of the game we have the modal context $e^1 = \varepsilon$, the set of clauses $\mathcal{D}^1 = \mathcal{C}$ and the pointed model $\langle M^1, w_\varepsilon\rangle$ where $M^1 = (W^1, R^1, V^1)$, $W^1 = \{w_\varepsilon\}$, $R^1 = \emptyset$ and $V^1(w_\varepsilon)(x_\varepsilon) = 1$. At the kth round of the game Prover fixes some query set Q_{e^k} for e^k with respect to $\mathcal{D}^k$ and then adds a world that is $e^k b$-accessible from w_ε to the model, where $b \in Q_{e^k}$.

Let $Q^{max}_{e^k} = \{(a, (y^k_j, l_j)) \mid j \in [n]\}$. As the set $N_{e^k} = \{(e^k : y^k_j \to \Diamond l_j) \mid j \in [n]\}$ it follows by condition (a) of Definition 8.1.1 that every query set for e^k is a subset of $Q^{max}_{e^k}$. If Prover chooses to add a world that is $e^k(a, (y^k_j, l_j))$-accessible from w_ε to the model then $(M, w_\varepsilon) \models \Diamond^k l_j$. Hence intuitively at the kth round of the game Prover chooses some pigeonhole for the kth pigeon to occupy. If we let:

$$A_k = \{(a, (y^k_j, l_j) \mid (a, (y^{k_1}_j, l_j)) \lhd e^k,\ k_1 \in [k-1],\ j \in [n]\},$$

then A_k is the set of pigeonholes occupied by the first $k-1$ pigeons.

We will now give Delayer's strategy for the first n rounds of the game. If Prover queries some set Q_{e^k} at the kth round of the game then for each $b = (a, (y^k_j, l_j)) \in Q_{e^k}$ Delayer sets the weights as follows:

$$p_b = \frac{1}{|Q_{e^k}| - |A_k|}, \qquad \text{if } b \notin A_k,$$
$$p_b = 0, \qquad \text{otherwise.}$$

At each round Delayer forces Prover to put the kth pigeon into some unoccupied pigeonhole. This strategy can only be followed if $|Q_{e^k}| - |A_k| > 0$. Hence in order to prove our lower bound we need the following claim.

Claim. Delayer's strategy is such that for every $k \in [n]$ the set $Q_{e^k} = Q^{max}_{e^k}$.

We have already seen that $Q_{e^k} \subseteq Q^{max}_{e^k}$. To see why $Q_{e^k} \supseteq Q^{max}_{e^k}$ we can think of each query set Q_{e^k} as a set of candidate pigeonholes for the kth pigeon. By Proposition 8.1.1 the set Q_{e^k} must contain every pigeonhole that can possibly be occupied by pigeon k, whilst satisfying $\bigcup_{e \in \bar{\mathcal{E}}_{e^k \sqsupseteq}} \mathcal{D}^k_e$. This set is satisfied by any model corresponding to an assignment of pigeons to pigeonholes where for each $i \in [k-1]$ the ith pigeon is put into the pigeonhole specified to by the ith symbol of e^k, and the kth pigeon is just in some pigeonhole. Hence as there is no restriction on which pigeonhole is occupied by pigeon k we have $Q_{e^k} \supseteq Q^{max}_{e^k}$.

Before giving a formal proof of our claim we will explain how it allows us to prove our lower bound. By the above claim $|Q_{e^k}| = n$ and so for each k we have $|Q_{e^k}| - |A_k| = n - (k-1)$. Hence Delayer can follow the above strategy for the first n rounds. It follows that Delayers score at the end of the nth round will be:

$$s^n = \sum_{k=1}^{n} (\log(n+1-k)) = \log\left(\prod_{k=1}^{n}(n+1-k)\right) = \log(n!).$$

As Delayer scores at least 0 at each round his final score must be $\geq s^n$ and so by Theorem 8.3.1 any tree-like $\mathbf{K}_{mc}$-Res refutation of $\mathcal{MPHP}^m_n$ contains at least $2^{s^n} - 1 = n! - 1$ modal steps.

Proof of claim: We will now give a formal proof of our claim. To prove that $Q_{e^k} \supseteq Q^{max}_{e^k}$ it suffices to show that for every $b \in Q^{max}_{e^k}$ there exists some model $M = (W, R, V)$ and some world $w_\varepsilon \in W$ such that $(M, w_\varepsilon) \models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e^k\sqsupseteq}} \mathcal{D}^k_{e_1} \cup \bigcup_{c \in Q^{max}_{e^k}\setminus\{b\}} \mathcal{M}_{e^k c}$, where each $\mathcal{M}_{e^k c}$ is the set of modally inferable clauses for $e^k c$ with respect to $\mathcal{D}_{e^k}$.

In fact we prove will a slightly stronger result. Namely that for every e^k and every $b \in Q^{max}_{e^k}$ there exists a model $M = (W, R_a, V)$ where:

$$W = \{w_e \mid e \sqsubseteq e^k\}, \quad R_a = \{(w_e, w_{ec}) \mid w_e, w_{ec} \in W \text{ and } c \in \bar{\mathcal{E}}_{\mathcal{C}}\}, \tag{9.1}$$

$$V(w_e)(l_j) = \begin{cases} 1 & \text{if } j = k_2 \text{ and } e = e'(a, (y^{k_1}_{k_2}, l_{k_2})) \text{ for some } e' \in \bar{\mathcal{E}}^*_{\mathcal{C}}, \\ 0 & \text{if } j \neq k_2 \text{ and } e = e'(a, (y^{k_1}_{k_2}, l_{k_2})) \text{ for some } e' \in \bar{\mathcal{E}}^*_{\mathcal{C}}, \end{cases} \tag{9.2}$$

for all $w_e \in W$ and all $j, k_1, k_2 \in [n]$. Further $(M, w_\varepsilon) \models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e^k\sqsupseteq}} \mathcal{D}^k_{e_1} \cup \bigcup_{c \in Q^{max}_{e^k}\setminus\{b\}} \mathcal{M}_{ec}$. To prove this we use induction on k.

If $k = 1$ then $e^1 = \varepsilon$. Let $b = (a, (y^1_p, l_p)) \in Q^{max}_\varepsilon$ where $p \in [n]$ and let $M = (W, R_a, V)$ where:

$$W = \{w_\varepsilon\}, \quad R_a = \emptyset, \quad V(w_\varepsilon)(x_\varepsilon) = 1, \quad V(w_\varepsilon)(x^j_{i,i',1}) = 1, \quad V(w_\varepsilon)(y^j_{i,i',1}) = 1,$$

$$V(w_\varepsilon)(y^1_j) = \begin{cases} 1 & \text{if } j = p, \\ 0 & \text{if } j \neq p, \end{cases} \text{ and } V(w_\varepsilon)(x^1_j) = \begin{cases} 1 & \text{if } j < p, \\ 0 & \text{if } j \geq p, \end{cases}$$

for all $i, i' \in [m]$ and $j \in [n]$. Then $(M, w_\varepsilon) \models \bigcup_{e \in \bar{\mathcal{E}}_{\varepsilon\sqsupseteq}} \mathcal{D}^1_e = \mathcal{C}_\varepsilon$. Further, for each $c = (a, (y^1_j, l_j)) \in Q^{max}_{e^1}$ the set $\mathcal{M}_c$ contains only clauses that can be inferred by applying some modal rule of $\mathbf{K}_{mc}$-Res to some set of clauses containing $(\varepsilon : y^1_j \to \Diamond l_j)$. Hence every clause in $\mathcal{M}_c$ is of the form $(\varepsilon : A \vee \neg y_j)$, where A is a propositional clause and so $(M, w_\varepsilon) \models \bigcup_{c \in Q^{max}_\varepsilon\setminus\{b\}} \mathcal{M}_c$.

Suppose $k > 1$. Delayer's strategy ensures that $e^k = e^{k-1}b_1$ for some $b_1 \in Q^{max}_{e^{k-1}} \setminus A_{k-1}$.

Hence by the inductive hypothesis there exists a partial model $M' = (W', R', V')$ where:

$$W' = \{w_e \mid e \sqsubseteq e^{k-1}\}, \quad R' = \{(w_e, w_{ec}) \mid w_e, w_{ec} \in W \text{ and } c \in \bar{\mathcal{E}}_{\mathcal{C}}\}$$

$$V'(w_e)(l_j) = \begin{cases} 1 & \text{if } j = j_1 \text{ and } e = e'(a, (y^{k_1}_{j_1}, l_{j_1})) \text{ for some } e' \in \bar{\mathcal{E}}^*_{\mathcal{C}}, \\ 0 & \text{if } j \neq j_1 \text{ and } e = e'(a, (y^{k_1}_{j_1}, l_{j_1})) \text{ for some } e' \in \bar{\mathcal{E}}^*_{\mathcal{C}}, \end{cases}$$

for all $w_e \in W$ and all $j \in [n]$. Further $(M', w_\varepsilon) \models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e^{k-1}\sqsupseteq}} \mathcal{D}^{k-1}_{e_1} \bigcup_{c \in Q^{max}_{e^{k-1}} \setminus \{b_1\}} \mathcal{M}_{e^{k-1}c}$.

For each $b_2 = (a, (y^k_p, l_p)) \in Q^{max}_{e^k}$ we can construct a model M, whose worlds, relations and valuations are as in Equations (9.1) and (9.2), and which satisfies $\bigcup_{e_1 \in \bar{\mathcal{E}}_{e^k\sqsupseteq}} \mathcal{D}^k_{e_1} \cup \bigcup_{c \in Q^{max}_{e^k} \setminus \{b_2\}} \mathcal{M}_{e^{k-1}c}$ at w_ε as follows. Let $M = (W, R, V)$ where:

$$W = W' \cup \{w_{e^k}\}, \quad R = R' \cup \{(w_{e^{k-1}}, w_{e^k})\},$$

$$V(w)(x) = V'(w)(x) \text{ for every } w \in W' \text{ and } V(w_{e^k})(l_j) = \begin{cases} 1 & \text{if } b_2 = (a, (y^k_j, l_j)), \\ 0 & \text{otherwise.} \end{cases}$$

Then M is a model of the required form. Further as M is an extension of M' we have $(M, w_\varepsilon) \models \bigcup_{e_1 \in \bar{\mathcal{E}}_{e^{k-1}\sqsupseteq}} \mathcal{D}^{k-1}_{e_1} \cup \bigcup_{c \in Q^{max}_{e^{k-1}} \setminus \{b_1\}} \mathcal{M}_{e^{k-1}c}$. Hence to show that $Q^{max}_{e^k} \setminus \{b_2\}$ is not a query set for e^k with respect to $\mathcal{D}^k$ all we have to do is show that $(M, w_\varepsilon) \models \mathcal{C}_{e^k} \cup \bigcup_{c \in Q^{max}_{e^k} \setminus \{b_2\}} \mathcal{M}_{e^k c}$.

Recall that for each $c = (a, (y^k_j, l_j)) \in Q^{max}_{e^k}$ the set $\mathcal{M}_{ec}$ contains only clauses of the form $(A \vee \neg y^k_j)$, where $A \in \mathcal{CL}$. Hence to ensure that $(M, w_\varepsilon) \models \bigcup_{c \in Q^{max}_{e^k} \setminus \{b_2\}} \mathcal{M}_{e^k c}$ we let:

$$V(w_{e^k})(y^k_j) = \begin{cases} 1 & \text{if } j = p, \\ 0 & \text{if } j \neq p. \end{cases}$$

Note that the set $\mathcal{C}_{e^k}$ consists of every literal clause in $\mathcal{C}$ with modal context a^{k-1}, every positive modal clause in $\mathcal{C}$ with modal context a^{k-2} and the negative modal clause $(a^{k-2} : y^{k-1}_q \to \Diamond l_q)$, where is q such that $b_1 = (a, (y^{k-1}_q, l_q))$. As $V(w_{e^k})(l_j) = 1$ if and only if $j = q$ it follows that $(M, w_\varepsilon) \models (a^{k-2} : y^{k-1}_q \to \Diamond l_q)$. If we further let:

$$V(w_{e^k})(z^{i_1}_{k-1}) = 1 \text{ and } V(w_{e^k})(x^k_j) = \begin{cases} 1 & \text{if } j < p, \\ 0 & \text{if } j \geq p, \end{cases}$$

for all $k < i_1 \leq n$ and all j then it is not hard to see that every positive modal clause of the form $(a^{k-2} : z^{i'}_{k-2} \to \Box z^{i'}_{k-1})$ and every literal clause in $\hat{P}_{k-1}$ is satisfied at w_ε in M.

We have now shown that $(M, w_\varepsilon) \models \mathcal{C}_{e^k} \cap \bigcup_i \hat{P}_i$. It remains is to show that $(M, w_\varepsilon) \models \mathcal{C}_{e^k} \cap \bigcup_{i \neq i', j} \hat{H}^j_{i,i'}$. The set $\mathcal{C}_{e^k} \cap \bigcup_{i \neq i, j} \hat{H}^j_{i,i'}$ consists of all clauses of the following forms:

1. $(a^{k-2} : x^j_{i,i',k-1} \to \Box x^j_{i,i',k})$,
2. $(a^{k-2} : y^j_{i,i',k-1} \to \Box y^j_{i,i',k})$,
3. $(a^{k-2} : x^j_{k-1,i',k-1} \to \Box \neg l_j)$,
4. $(a^{k-2} : y^j_{i,k-1,k-1} \to \Box \neg l_j)$,

where $i < i' \in [m]$ and $j \in [n]$. Every clause of the form 1 and 2 can be satisfied at w_ε in M by letting:

$$V(w_{e^k})(x^j_{i,i',k}) = 1 \text{ and } V(w_{e^k})(y^j_{i,i',k}) = 1.$$

However clauses of the form 3 and 4 do not contain any unassigned variables, so we cannot simply extend V to ensure that they are satisfied at w_ε in M. A clause $(a^{k-2} : x^j_{k-1,i',k-1} \to \Box \neg l_j)$ is not satisfied at w_ε in M if and only if $V(w_{e^{k-1}})(x^j_{k-1,i',k-1}) = 1$ and $V(w_{e^k})(l_j) = 1$. As $V(w_{e^k})(l_j) = 0$ for all $j \neq q$ this can only be the case if $j = q$. Hence to ensure that every such clause is satisfied we set:

$$V(w_{e^{i_1}})(x^q_{k-1,i',i_1}) = 0 \text{ for all } i_1 < k,$$

However doing this may cause the clause to become false and so we also set:

$$V(w_{e^{i_1}})(y^q_{k-1,i',i_1}) = 1 \text{ for all } i_1 < k,$$

Note that every clause containing these variables is in $H^q_{i,i'}$. By inspection we can easily see that changing the assignments of these variables as above does not change the truth valuation of any clause in $\mathcal{C}_{e^k} \cap H^q_{i,i'}$. Similarly to ensure that every clause of the form $(a^{k-2} : y^j_{i,k-1,k-1} \to \Box \neg l_j)$ is satisfied at w_ε in M we set:

$$V(w_{e^{i_1}})(x^q_{i,k-1,i_1}) = 1 \text{ and } V(w_{e^{i_1}})(y^q_{i,k-1,i_1}) = 0 \text{ for all } i_1 < k$$

Note that this will not cause the clause $(a^{i-2} : x^q_{i,k-1,i} \to \Box \neg l_q)$ to be falsified as Delayer's strategy ensures that e^k contains no repeated pigeonholes and so $V(w_{e^i})(l_q) = 0$. Hence as above changing the assignments of these variables will not change the truth valuation of any clause in $\mathcal{C}_{e^k} \cap H^q_{i,i'}$.

This concludes the proof of our claim, and so the proof of the theorem. □

Recall from Corollary 4.6.2 that the tree-like **K**-Res systems are all p-equivalent to one another. Hence the above lower bound is a lower bound for every tree-like **K**-Res systems.

Chapter 10

Modal proof systems beyond resolution

In this chapter we compare the proof complexity of the family of both tree-like and dag-like **K**-Res systems with that of Frege systems for $\mathbf{K}_n$.

10.1 Comparing modal Frege systems with modal resolution systems

In propositional proof complexity Frege systems (defined in Subsection 3.1.1) are among the most studied proof systems. Indeed it was shown in [27] that every Frege system over a fixed set of logical connectives is p-equivalent to every other Frege system over that set of connectives. It was further shown in [79] that this p-equivalence holds even for Frege systems over different sets of connectives. Hence we can take the Frege system shown in Figure 3.1 (or any other Frege system) to be the canonical Frege system.

Definition 10.1.1. Let P be a Frege system. We say that a rule $\phi_1, \ldots, \phi_z \vdash \phi$ of P is *sound* if every model which satisfies $\phi_1, \ldots, \phi_z$ also satisfies ϕ.

The resolution rule is sound and so can be taken as a rule of any Frege system. Hence Frege p-simulates propositional resolution. Furthermore, there exists an exponential separation between Frege and propositional resolution [22]. That is, there exist propositional formulas that are known to be require exponential size proofs for propositional resolution and polynomial size proofs for Frege.

Definition 10.1.2. An *extended Frege* system is a Frege system with the additional axiom:

$$p \leftrightarrow \phi,$$

where p is a new propositional variable (called an extension variable), that does not appear in ϕ, any previously derived formulas or the final formula of the proof. We call this axiom the *extension axiom*.

Whilst extended Frege p-simulates Frege, there is as of yet no known exponential separation between these systems. In fact there are currently no propositional formulas that have been shown to require exponential size proofs in Frege.

Definition 10.1.3. A *Frege system for the modal logic* $\boldsymbol{K}_n$ ($\mathbf{K}_n$-Frege) is a line based proof system P consisting of a finite set of inference rules and axioms of the form $\phi_1, \ldots, \phi_k \vdash_P \phi$ and $\vdash_P \phi$ respectively, where $\phi_1, \ldots, \phi_k, \phi$ are modal formulas. Further P must be sound and strongly complete.

Given any propositional Frege system we can obtain a $\mathbf{K}_n$-Frege system by adding the following rules, for every $a \in \mathcal{A}$:

$$K_a\colon \quad \Box_a(A \to B) \to (\Box_a A \to \Box_a B) \quad \text{and} \quad \text{NEC}_a\colon \frac{A}{\Box_a A}\,.$$

Definition 10.1.4. An *extended Frege system for* $\boldsymbol{K}_n$ is a $\mathbf{K}_n$-Frege system together with the extension axiom.

Definition 10.1.5. [49] We say that a $\mathbf{K}_n$-Frege system P is *standard* if every formula ϕ for which $\phi_1, \ldots, \phi_k \vdash_P \phi$ is in the closure of $\mathbf{K}_n \cup \{\phi_1, \ldots, \phi_k\}$ under MP and NEC_a for every $a \in \mathcal{A}$.

We can extend Definition 10.1.1 to $\mathbf{K}_n$-Frege systems by saying that a rule, $\phi_1, \ldots, \phi_z$ is sound if and only if whenever $\phi_1, \ldots, \phi_z$ are satisfied at some world w in some Kripke model M, so is ϕ. A $\mathbf{K}_n$-Frege system can be non-standard only if it contains some rule, other than NEC_a, which is not sound.

Every standard $\mathbf{K}_n$-Frege system p-simulates every other standard $\mathbf{K}_n$-Frege system [49]. Hence we can take for example the Frege system in Figure 3.1 together with K_a and NEC_a for every $a \in \mathcal{A}$ to be the canonical standard $\mathbf{K}_n$-Frege systems. The analogous statement is not known to hold for non-standard $\mathbf{K}_n$-Frege systems, hence here we only consider standard $\mathbf{K}_n$-Frege systems.

Proposition 10.1.1. $\mathbf{K}_n$-Frege p-simulates $\mathbf{K}_{mp}$-Res.

Proof. Each of the rules of $\mathbf{K}_{mp}$-Res is sound. Hence the proposition follows immediately. □

Corollary 10.1.1. $\mathbf{K}_n$-Frege p-simulates each of the **K**-Res proof systems.

Proof. This follows from above proposition and the p-equivalence of the family of **K**-Res systems (Corollary 4.6.1). □

Corollary 10.1.2. $\mathbf{K}_n$-Frege p-simulates each of the tree-like **K**-Res proof systems.

Proof. Every tree-like **K**-Res proof is also a dag-like **K**-Res proof and so this follows trivially from Corollary 10.1.1. □

10.2 Separation of $\mathbf{K}_n$-Frege and tree-like $\mathbf{K}_{mc}$-Res

It was proved by Buss in [22] that there exist polynomial size Frege refutations of the propositional pigeonhole principle with $m > n$ pigeons. Buss' proof relies on the fact that Frege systems can count efficiently. The idea behind the proof is as follows. If we assume that the pigeonhole principle holds for some $m > n$ and count the number of holes that are occupied by the first $n + 1$ pigeons then we can construct a polynomial size Frege derivation of some formula encoding that this number is greater than n. However as there are only n pigeonholes, we can also construct polynomial size Frege derivation of a formula encoding that that the number of occupied holes is less than or equal to n, leading to a contradiction.

We show that a very similar proof can be used to prove that there exists a polynomial size $\mathbf{K}_n$-Frege proof of the modal pigeonhole principle. We will give a sketch of Buss' proof of the pigeonhole principle, highlighting the steps that make explicit use of PHP_n^m.

Recall that:

$$PHP_n^m = \bigwedge_{i\in[m]} \bigvee_{j\in[n]} p_{i,j} \wedge \bigwedge_{1\leq i<i'\leq m} \bigwedge_{j\in[n]} (\neg p_{i,j} \vee \neg p_{i',j}).$$

Theorem 10.2.1 ([22])**.** There exist polynomial size Frege refutations of PHP_n^m, where $m > n$.

Sketch. The proof has two parts. First we show that there exists a polynomial size extended Frege derivation of 0 from PHP_n^m. We will then show that this extended Frege derivation can be used to obtain a polynomial size Frege derivation of 0 from PHP_n^m.

The extended Frege refutation of PHP_n^m is obtained as follows. First, for each $i \in [m]$ and $j \in [n]$ we introduce an extension variable r_j^i which abbreviates the formula $\bigvee_{k\in[i]} p_{k,j}$. Each r_j^i is true if and only if one of the first i pigeons occupies pigeonhole j. Hence the number of r_j^i's that are true in a given assignment is equal to the number of pigeonholes occupied by the first i pigeons. If we assume without loss of generality that n is a power of 2 then for each i we can define $\log n$ formulas $a^{i,1}, \ldots, a^{i,\log n}$ (henceforth denoted by the vector $\vec{a}^i$) which encode the number of r_j^i's which are true in a given model. We denote the number encoded by $\vec{a}^i$ (that is the number of pigeonholes occupied by the first i pigeons) by A^i.

Let:

$$\phi_i = \left(\bigwedge_{j\in[n]} \left(r_j^i \rightarrow r_j^{i+1} \right) \wedge \bigvee_{j\in[n]} \left(r_j^{i+1} \wedge \neg r_j^i \right) \right).$$

This formula states that for each $j \in [n]$ if one of the first i pigeons is in pigeonhole j then so is one of the first $i + 1$ pigeons, and that some pigeonhole $j \in [n]$ is occupied by the $i + 1$th pigeon. It is not hard to see from the definition of r_j^1 that there exists a polynomial size Frege proof of $PHP_n^m \rightarrow \bigvee_{j\in[n]} r_j^1$. Further, given $\bigvee_{j\in[n]} r_j^1$ there exists a polynomial size Frege derivation of a formula encoding that $0 < A^1$. Similarly, for each $i \in [m]$ there exists a straightforward polynomial size Frege proof of $PHP_n^m \rightarrow \phi_i$ and, for each i given ϕ_i there exists a polynomial

size Frege derivation of a propositional formula encoding that $A^i < A^{i+1}$. Finally these proofs can be combined to obtain polynomial size proofs of a formula encoding that $n < A^{n+1}$ from $\bigwedge_{i\in[m]} \phi_i \wedge \bigvee_{j\in[n]} r^1_j$ (and so from PHP^m_n).

There further exists a polynomial size Frege proof of a formula which encodes that $n \geq A^n$. Hence there exists a polynomial size Frege derivation of a formula encoding that $n < n$ from PHP^m_n. As $n = n$ this formula must be false and so we have a polynomial size Frege refutation of PHP^m_n.

For every i, each formula in $\vec{a}^i$ can be defined so that it has size polynomial in that of the largest propositional formula abbreviated by any r^i_j. Hence replacing all the extension variables in the extended Frege proof of $PHP^m_n \to 0$ with the formulas they abbreviate yields a polynomial size Frege refutation of PHP^m_n. □

Recall that:

$$\mathbf{MPHP}^m_n = \bigwedge_i P_i \wedge \bigwedge_{1\leq i<i'\leq m} \bigwedge_j H^j_{i,i'},$$

where $P_i = \Box^{i-1}(\bigvee^n_{j=1} \Diamond l_j)$ for every $1 \leq i \leq m$ and $H^j_{i,i'} = \Box^i \neg l_j \vee \Box^{i'} \neg l_j$ for every $1 \leq j \leq n$ and $1 \leq i < i' \leq m$.

Theorem 10.2.2. There exists a polynomial size $\mathbf{K}_n$-Frege refutation of $\mathbf{MPHP}^m_n$, where $m > n$.

Proof. Note that the only parts of the extended Frege refutation of PHP^m_n that depend on the formula itself are the proofs of:

$$PHP^m_n \to \bigvee_{j\in[n]} r^1_j \quad \text{and} \quad PHP^m_n \to \bigwedge_{i\in[m]} \phi_i.$$

Hence to show that there exists a polynomial size extended $\mathbf{K}_n$-Frege refutation of $\mathbf{MPHP}^m_n$ it suffices to show that we there exist polynomial size $\mathbf{K}_n$-Frege proofs of $\mathbf{MPHP}^m_n \to \bigvee_{j\in[n]} r^1_j$ and $\mathbf{MPHP}^m_n \to \bigwedge_{i\in[m]} \phi_i$ for some suitable choice of extension variables r^i_j.

Let $r^i_j \leftrightarrow \bigvee_{k\in[i]} \Diamond^k l_j$. Then as in the propositional case r^i_j is true if and only if one of the first i pigeons occupies pigeonhole j. There exist short $\mathbf{K}_n$-Frege derivations of $\mathbf{MPHP}^m_n \to \bigvee_{j\in[n]} \Diamond l_j$ and so of $\mathbf{MPHP}^m_n \to \bigvee_{j\in[n]} r^1_j$. Further for each $j \in [n]$ and $i \in [m-1]$ there exists a simple $\mathbf{K}_n$ Frege proof of $r^i_j \to r^{i+1}_j$. It remains to show that for each $i \in [m-1]$ there exists a polynomial size Frege proof of $\mathbf{MPHP}^m_n \to \bigvee_{j\in[n]} (r^{i+1}_j \wedge \neg r^i_j)$. To see this first note that there exists a simple Frege proof of:

$$\left(\Diamond^{i+1} l_j \wedge \bigwedge_{k\in[i]} \neg \Diamond^k l_j \right) \to \left(r^{i+1}_j \wedge \neg r^i_j \right).$$

As the formulas $(\Diamond\psi_1 \wedge \Box\psi_2) \to \Diamond\psi_2$ and $(\Diamond\psi_1 \vee \Diamond\psi_2) \to \Diamond(\psi_1 \vee \psi_2)$ are theorems of $\mathbf{K}_n$ we can assume without loss of generality that they are axioms of our $\mathbf{K}_n$-Frege system. Hence we can

easily obtain a polynomial size $\mathbf{K}_n$-Frege proof of:

$$\bigwedge_{k\in[i+1]} P_k \rightarrow \bigvee_{j\in[n]} \Diamond^{i+1} l_j.$$

Further there exist small $\mathbf{K}_n$-Frege proofs of:

$$\left(\Diamond^{i+1} l_j \wedge \bigwedge_{k\in[i]} \bigwedge_{j\in[n]} H^j_{k,i+1}\right) \rightarrow \left(\Diamond^{i+1} l_j \wedge \bigwedge_{k\in[i]} \neg\Diamond^k l_j\right).$$

Putting these proofs together we obtain a polynomial size proof of:

$$\mathbf{MPHP}^m_n \rightarrow \bigvee_{j\in[n]} (r^{i+1}_j \wedge \neg r^i_j).$$

Hence we have shown that $\mathbf{MPHP}^m_n \rightarrow \bigwedge_{i\in[m]} \phi_i$ and so by the reasoning in the proof of Theorem 10.2.1 there exists a polynomial size extended $\mathbf{K}_n$-Frege refutation of $\mathbf{MPHP}^m_n$.

As in the propositional case for every i, each formula in $\vec{a}^i$ can be defined so that it has size polynomial in that of the largest formula abbreviated by any r^i_j. Hence replacing all the extension variables in the extended Frege proof of $\mathbf{MPHP}^m_n \rightarrow 0$ with the formulas they abbreviate yields a polynomial size $\mathbf{K}_n$-Frege refutation of $\mathbf{MPHP}^m_n$. □

Corollary 10.2.1. There exists an exponential separation between the proof size required to refute $\mathbf{MPHP}^m_n$ in $\mathbf{K}_n$-Frege and tree-like $\mathbf{K}_{mc}$-Res.

Proof. This follows immediately from Theorem 9.2.1 and Theorem 10.2.2. □

Propositional separations between $\mathbf{K}_n$-Frege and both tree-like and dag-like $\mathbf{K}_{mp}$-Res follow trivially from the fact that a number of propositional formulas (including the propositional pigeonhole principle) have previously been shown to be hard for (both tree-like and dag-like) propositional resolution but easy for propositional Frege. Hence the significance of the above result is that tree-like $\mathbf{K}_{mp}$-Res requires an exponential number of *modal* resolution steps to refute $\mathbf{MPHP}^m_n$, whereas there exists a polynomially size $\mathbf{K}_n$-Frege refutation of $\mathbf{MPHP}^m_n$. Clearly any polynomial size $\mathbf{K}_n$-Frege refutation may contain at most a polynomial number of modal proof steps (i.e. applications of K_a or NEC$_a$) and so the separation in Corollary 10.2.1 is a truly modal one.

10.3 Game theoretic lower bound technique vs existing lower bound techniques

Recall from Section 6.4 that the modal clique colour formulas:

$$Clique^{k+1}_n(\Box\bar{p}, \bar{q}) \rightarrow \Box(\neg Colour^k_n(\bar{p}, \bar{r})),$$

are an exponential lower bound for $\mathbf{K}_n$-Frege. Notably, this lower bound is in fact an exponential lower bound on the number of K axioms needed to prove $Clique_n^k(\Box\bar{p}, \bar{r}) \rightarrow \Box(\neg Colour_n^k(\bar{p}, \bar{r}))$ in K_n-Frege, and so is a modal lower bound for $\mathbf{K}_n$-Frege.

Further recall from Theorem 6.4.2 that the modal clique-colour formulas give an exponential lower bound for the **K**-Res systems. As the negation of the modal clique-colour formula is of the form:

$$Clique_n^{k+1}(\Box\bar{p}, \bar{q}) \wedge \Diamond Colour_n^k(\bar{p}, \bar{r}),$$

where $Colour_n^k(\bar{p}, \bar{r})$ is a propositional formula, the set of SNF_{mc} clauses obtained by applying the translation function T_{mc} to this formula must contain only one negative modal clause. Hence any modal decision tree for any refutation of the modal clique-colour formulas contains at most two vertices. No instance of our two-player game played on such a tree can result in a high Delayer score and so our game theoretic lower bound technique cannot be used to show that the modal clique-colour formulas are an exponential modal lower bound for tree-like $\mathbf{K}_{mc}$-Res.

In [13] it was shown that a propositional Prover-Delayer game characterises the proof size of tree-like propositional resolution. That our game cannot be used to prove the hardness of the modal clique-colour formula illustrates that it does not provide a characterisation of the modal proof size of tree-like $\mathbf{K}_{mc}$-Res. It is unsurprising that our modal game fails to provided such a characterisation as it counts only the number of distinct modal contexts which need to be considered when refuting a formula, not the total number of modal resolution steps required.

Chapter 11

Conclusion

11.1 Summary of our contributions

In this thesis we have initiated the study of the proof complexity of modal resolution systems. Our main contributions are:

- Defining several refinements of the resolution systems $\mathbf{K}_n$-Res and $\mathbf{K}_{ml}$-Res, and further establishing that all of these **K**-Res resolution systems are p-equivalent to one another.
- Comparing the strength of the **K**-Res systems with $\mathbf{RK}_n$ and $\mathbf{K}_n$-Frege. In particular showing that $\mathbf{RK}_n$ and $\mathbf{K}_n$-Frege both p-simulate the **K**-Res systems, and that there is a separation between the tree-like **K**-Res systems and $\mathbf{K}_n$-Frege.
- Introducing two new lower bound proving techniques for the **K**-Res systems. The first of these (feasible interpolation) is a reasonably straightforward adaptation of the analogous propositional lower bound proving technique. Whereas our game theoretic lower bound technique differs rather significantly from known game theoretic lower bound proving techniques for propositional resolution. We have further shown that our modal game proves lower bounds on the number of modal resolution steps needed to refute families of formulas, hence these lower bounds are truly modal lower bounds.
- Showing that size width, which is arguably the main lower bound proving technique for propositional resolution, cannot be used to obtain lower bounds for either the **K**-Res systems or $\mathbf{RK}_n$.
- Proving a new exponential lower bound for the **K**-Res systems. This is obtained by applying our modal feasible interpolation technique to Hrubeš' modal clique-colour formulas.

- Proving a new exponential lower bound for the tree-like versions of the **K**-Res systems. To obtain this lower bound we defined a new family of modal formulas which we called the modal pigeonhole principle. We then applied our game theoretic lower bound technique to these formulas to obtain an exponential lower bound for tree-like **K**-Res.

11.2 Open questions and further directions

Our comparison of **K**-Res with $\mathbf{RK}_n$ is incomplete as whilst we have shown that $\mathbf{RK}_n$ p-simulates the **K**-Res systems, we have neither proved that $\mathbf{RK}_n$ is p-simulated by the **K**-Res systems nor shown a separation between these systems. We conjecture that $\mathbf{RK}_n$ is p-simulated by the **K**-Res systems and believe this can be shown by reordering the inference steps of $\mathbf{RK}_n$ refutations. More precisely we believe that given any $\mathbf{RK}_n$ refutation we can reorder the inferences so that all pivots with some fixed maximal modal context (i.e. some modal context of maximal modal depth) are resolved on first, followed by all pivots with some modal context which is maximal with respect to the remaining modal contexts and so on. Given a refutation in this form the inferences could be further reordered to correspond to **K**-Res inferences.

Further whilst we have shown a modal separation between tree-like **K**-Res and $\mathbf{K}_n$-Frege, it remains open as to whether there exists a modal separation between the full dag-like versions of the **K**-Res systems and their tree-like restrictions. Hence the question of whether there exists a modal separation between the dag-like **K**-Res systems and $\mathbf{K}_n$-Frege also remains open.

The only known hard formulas for modal proof systems are our modal pigeonhole principle, and Hrubeš' modal clique-colour formula. Both of these formulas are obtained by taking a hard propositional formula and then augmenting it with some modal operators. It would be interesting to establish whether there exists some general technique which can be used to non-trivially convert hard propositional formulas into hard modal formulas. Indeed, we believe that it is possible to convert the propositional clique formulas of [4, 13, 14] into hard modal formulas using the same ideas as for obtaining our modal pigeonhole principle from the propositional version.

We hope that our contributions are only the beginning of a much more in depth study of the proof complexity of modal resolution systems, which not only offers a wider perspective on proof complexity in general but also contributes towards the development of better modal theorem provers. Whilst in this thesis we have chosen to focus primarily on the proof complexity of the family of **K**-Res proof systems, both because of the comparative simplicity of their clausal forms and the fact that one of these systems has an associated prover, there exist many other modal resolution systems for $\mathbf{K}_n$ (e.g. [3, 5, 34, 61, 62]) whose proof complexity should be investigated. Further the **K**-Res systems, whose proof complexity has been the main focus of this thesis, have been extended both to the global satisfiability problem for $\mathbf{K}_n$ [64] and to a number of modal logics beyond $\mathbf{K}_n$ (e.g. preferential logics [68], coalition logics [47] and modal logics of confluence [67]). Studying the proof complexity of such systems would be of particular interest as they are able to deal with more expressive problems and some even have associated theorem provers [47].

Bibliography

[1] Martín Abadi and Zohar Manna. Modal theorem proving. In *8th International Conference on Automated Deduction*, pages 172–189. Springer Berlin Heidelberg, 1986.

[2] Noga Alon and Ravi Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[3] Carlos Areces, Hans de Nivelle, and Maarten de Rijke. Resolution in Modal, Description and Hybrid Logic. *Journal of Logic and Computation*, 11(5):717–736, 2001.

[4] Albert Atserias, Ilario Bonacina, Susanna de Rezende, Massimo Lauria, Jakob Nordström, and Alexander A. Razborov. Clique is hard on average for regular resolution. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing STOC*, pages 866–877, 2018.

[5] Yves Auffray, Patrice Enjalbert, and Jean-Jacques Hébrard. Strategies for modal resolution: Results and problems. *Journal of Automated Reasoning*, 6(1):1–38, 1990.

[6] Peter Balsiger, Alain Heuerding, and Stefan Schwendimann. A benchmark method for the propositional modal logics K, KT, S4. *Journal of Automated Reasoning*, 24(3):297–317, 2000.

[7] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *Journal of Artificial Intelligence Research*, 22:319–351, 2004.

[8] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific Publishing, 2001.

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

[10] Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 249–260, 2016.

[11] Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *32nd International Symposium on Theoretical Aspects of Computer Science*.

[12] Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like q-resolution size. *Journal of Computer and System Sciences*, 104:82–101, 2019.

[13] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A characterization of tree-like resolution size. *Information Processing Letters*, 113(18):666–671, 2013.

[14] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Parameterized complexity of DPLL search procedures. *ACM Transactions on Computational Logic*, 14(3), 2013.

[15] Olaf Beyersdorff and Oliver Kutz. Proof complexity of non-classical logics. In *Lectures on Logic and Computation - ESSLLI 2010 /ESSLLI 2011, Selected Lecture Notes*, pages 1–54. Springer-Verlag, Berlin Heidelberg, 2012.

[16] Olaf Beyersdorff, Arne Meier, Sebastian Müller, Michael Thomas, and Heribert Vollmer. Proof complexity of propositional default logic. *Archive for Mathematical Logic*, 50(7):727–742, 2011.

[17] Meghyn Bienvenu, Hélène Fargier, and Pierre Marquis. Knowledge compilation in the modal logic S5. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*, 2010.

[18] Marta Bílková. *Interpolation in modal logics*. PhD thesis, Charles University Prague, 2006.

[19] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

[20] Archie Blake. *Canonical expressions in boolean algebra*. PhD thesis, University of Chicago, 1937.

[21] Maria Luisa Bonet, Samuel Buss, and Toniann Pitassi. Are there hard examples for Frege systems? In *Feasible Mathematics II*, pages 30–56. Birkhäuser, 1995.

[22] Samuel Buss. Polynomial size proofs of the propositional pigeonhole principle.

[23] Samuel Buss. Towards NP-P via proof complexity and search. *Annals of Pure and Applied Logic*, 163(7):906–917, 2012.

[24] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[25] Stephen Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[26] Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[27] Stephen Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[28] Thomas Cover and Joy Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006.

[29] William Craig. Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory. *The Journal of Symbolic Logic*, 22(3):269–285, 1957.

[30] David Tena Cucala, Bernardo Cuenca Grau, and Ian Horrocks. 15 years of consequence-based reasoning. In *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, pages 573–587, 2019.

[31] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:210–215, 1960.

[32] Hans de Nivelle, Renate Schmidt, and Ullrich Hustadt. Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8:265–292, 2000.

[33] Luis Fariñas del Cerro. A simple deduction method for modal logic. *Information Processing Letters*, 14(2):49–51, 1982.

[34] Patrice Enjalbert and Luis Fariñas del Cerro. Modal resolution in clausal form. *Theoretical Computer Science*, 65(1):1–33, 1989.

[35] Liangda Fang, Kewen Wang, Zhe Wang, and Ximing Wen. Knowledge compilation in the multi-agent epistemic logic K_n. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018*, pages 637–638, 2018.

[36] Melvin Fitting. Destructive modal resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.

[37] Dov Gabbay. Craig's interpolation theorem for modal logics. In *Conference in Mathematical Logic*, pages 111–127. Springer Berlin Heidelberg, 1972.

[38] Rajeev Goré. Tableau methods for modal and temporal logics. In *Handbook of Tableau Methods*. Kluwer, 1998.

[39] Rajeev Goré, Kerry Olesen, and Jimmy Thomson. Implementing tableau calculi using bdds: Bddtab system description. In *Automated Reasoning - 7th International Joint Conference, IJCAR. Proceedings*, pages 337–343, 2014.

[40] Daniel Götzmann, Mark Kaminski, and Gert Smolka. Spartacus: A tableau prover for hybrid logic. *Electronic Notes in Theoretical Computer Science*, 262:127–139, 2010.

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

[42] Joseph Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.

[43] Ian Horrocks. Ontologies and the semantic web. *Communications of the ACM*, 51(12):58–67, 2008.

[44] Pavel Hrubeš. A lower bound for intuitionistic logic. *Annals of Pure and Applied Logic*, 146(1):72–90, 2007.

[45] Pavel Hrubeš. Lower bounds for modal logics. *The Journal of Symbolic Logic*, 72(3):941–958, 2007.

[46] Pavel Hrubeš. On lengths of proofs in non-classical logics. *Annals of Pure and Applied Logic*, 157(2–3):194–205, 2009.

[47] Ullrich Hustadt, Paul Gainer, Clare Dixon, Cláudia Nalon, and Lan Zhang. Ordered resolution for coalition logic. In *Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference, TABLEAUX 2015. Proceedings*, pages 169–184, 2015.

[48] Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. *Artificial Intelligence*, 234:1–25, 2016.

[49] Emil Jeřábek. Substitution Frege and extended Frege proof systems in non-classical logics. *Annals of Pure and Applied Logic*, 159(1–2):1–48, 2009.

[50] Emil Jerábek. Proof complexity of intuitionistic implicational formulas. *Annals of Pure and Applied Logic*, 168(1):150–190, 2017.

[51] Mark Kaminski and Tobias Tebbi. Inkresat: Modal reasoning via incremental reduction to sat. In Maria Paola Bonacina, editor, *Automated Deduction – CADE-24*, pages 436–442. Springer Berlin Heidelberg, 2013.

[52] Marcus Kracht and Frank Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[53] Jan Krajíček. Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

[54] Agi Kurucz, Frank Wolter, Michael Zakharyaschev, and Dov Gabbay. *Many-Dimensional Modal Logics: Theory and Applications*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2003.

[55] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[56] Char-Tung Lee. *A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms*. PhD thesis, University of California, 1967.

[57] Leonid Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1975. Translation into English of Russian article originally published in 1973.

[58] Florian Lonsing and Uwe Egly. Depqbf 6.0: A search-based QBF solver beyond traditional QCDCL. In *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction*, pages 371–384, 2017.

[59] Emiliano Lorini and François Schwarzentruber. A modal logic of epistemic games. *Games*, 1(4):478–526, 2010.

[60] João Marques-Silva and Sharad Malik. Propositional SAT solving. In *Handbook of Model Checking*, pages 247–275. Springer, 2018.

[61] Grigori Mints. Gentzen-type systems and resolution rules. Part I. Propositional logic. In *COLOG-88, International Conference on Computer Logic, Proceedings*, pages 198–231, 1988.

[62] Grigori Mints. Resolution calculi for modal logics. *American Mathematical Society Translations*, 143:1–14, 1989.

[63] Cláudia Nalon and Clare Dixon. Clausal resolution for normal modal logics. *Journal of Algorithms*, 62(3-4):117–134, 2007.

[64] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. A modal-layered resolution calculus for K. In *Automated Reasoning with Analytic Tableaux and Related Methods - 24th International Conference*, pages 185–200, 2015.

[65] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. : A resolution-based prover for multimodal K. In *Automated Reasoning*, pages 406–415. Springer International Publishing, 2016.

[66] Cláudia Nalon, Ullrich Hustadt, and Clare Dixon. KSP: A resolution-based prover for multimodal K, abridged report. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4919–4923, 2017.

[67] Cláudia Nalon, João Marcos, and Clare Dixon. Clausal resolution for modal logics of confluence. In *Automated Reasoning - 7th International Joint Conference, IJCAR 2014. Proceedings*, pages 322–336, 2014.

[68] Cláudia Nalon and Dirk Pattinson. A resolution-based calculus for preferential logics. In *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Proceedings*, pages 498–515, 2018.

[69] Jakob Nordström. On the interplay between proof complexity and SAT solving. *SIGLOG News*, 2(3):19–44, 2015.

[70] Hans Jürgen Ohlbach. A resolution calculus for modal logics. In *9th International Conference on Automated Deduction, Argonne, Illinois, USA, May 23-26, 1988, Proceedings*, pages 500–516, 1988.

[71] Hans Jürgen Ohlbach. Translation methods for non-classical logics: An overview. *Logic Journal of the IGPL*, 1(1):69–89, 1993.

[72] Juan Pablo Aguilera Ozuna and David Fernández-Duque. Verification logic. *Journal of Logic and Computation*, 27(8):2451–2469, 2017.

[73] Guoqiang Pan, Ulrike Sattler, and Moshe Vardi. BDD-based decision procedures for the modal logic K. *Journal of Applied Non-Classical Logics*, 16(1-2):169–208, 2006.

[74] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175(2):512–525, 2011.

[75] Pavel Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.

[76] Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for SAT. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 128–136, 2000.

[77] Wolfgang Rautenberg. Modal tableau calculi and interpolation. *Journal of Philosophical Logic*, 12(4):403–423, 1983.

[78] Alexander Razborov. Lower bounds on the monotone complexity of boolean functions. *Doklady Akademii Nauk SSSR*, 282:1033–1037, 1985. English translation in: Soviet Mathematics Doklady, 31, pp. 354–357.

[79] Robert Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, University of Toronto, 1976.

[80] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

[81] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 466–471, 1991.

[82] Renate Schmidt and Ullrich Hustadt. *First-Order Resolution Methods for Modal Logics*, pages 345–391. Springer Berlin Heidelberg, 2013.

[83] Renate A. Schmidt. *E*-unification for subsystems of *S4*. In *Rewriting Techniques and Applications, 9th International Conference, RTA-98, Tsukuba, Japan, March 30 - April 1, 1998, Proceedings*, pages 106–120, 1998.

[84] Roberto Sebastiani and Michele Vescovi. Automated reasoning in modal and description logics via SAT encoding: the case study of K(m)/ALC-satisfiability. *Journal of Artificial Intelligence Research*, 35:343–389, 2009.

[85] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

[86] Edith Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.

[87] Heinrich Wansing. Sequent calculi for normal modal propositional logics. *Journal of Logic and Computation*, 4(2):125–142, 1994.