Modal logic and the polynomial hierarchy: from QBFs to K and back

Anupam Das [1]

University of Birmingham
United Kingdom

Sonia Marin

University of Birmingham
United Kingdom

Abstract

In this work we classify formulas of the basic normal modal logic K into fragments complete for each level of the polynomial time hierarchy, with respect to validity. In particular, we identify a pair of encodings, from true Quantified Boolean Formulas (QBFs) to modal logic and vice-versa, whose composition preserves the number of quantifier alternations. This yields a formal analogue of 'quantifier complexity' within modal logic.

Our translation from QBFs to modal formulas is an optimised version of common translations employed in modal logic solving. In the other direction, we encode proof search itself, for a cut-free sequent calculus, as an alternating time predicate. The aforementioned tight bounds are obtained by carefully calibrating both the optimisation (QBFs to modal logic) and the measurement of proof search complexity (modal logic to QBFs). This approach is inspired by recent work achieving a similar result for the exponential-free fragment of Linear Logic (MALL).

Keywords: Modal Logic, Polynomial Hierarchy. Quantified Boolean Formulas, Proof Search, Sequent Calculus

1 Introduction

Ladner's seminal work [13] showed that a large number of modal logics between K and $S4$ are **PSPACE**-complete. Adding further axioms, such as 5, can simplify the underlying complexity of the validity problem, with $S5$ being *co***NP**-complete. Indeed, the 'gap' between *co***NP**-complete and **PSPACE**-complete normal modal logics has formed the subject of several works in recent years [19,12]. [2]

[1] Supported by a UKRI Future Leaders Fellowship, *Structure vs. Invariants in Proofs.*

[2] Ladner's result was rather **NP**-completeness of the *satisfiability* problem for $S5$. In this work we only consider *validity*, which duly exhibits dual complexity bounds to satisfiability.

That said, as far as we know, attempts to characterise fragments of modal logics corresponding to levels of the polynomial hierarchy (**PH**) have not appeared in the literature. **PH** essentially delineates **PSPACE** according to 'bounded quantifier alternation', e.g. by identifying **PSPACE** with the set of true quantified Boolean formulas (QBFs), another well-known **PSPACE**-complete problem. On the other hand, translations from QBFs to modal logic now comprise a fundamental benchmark in modal satisfiability solving [16,17].

There are many known translations from QBFs to the basic normal modal logic K; some of those (whose variants are) employed for benchmarking modal satisfiability solvers include Ladner's original one [13], a more optimal one due to Schmidt-Schauss and Scholka [21], [3] and that of Pan and Vardi designed to reduce modal solving to QBF solving [20]. These translations, their utility for benchmarking, and the approach to modal solving by QBF-encoding, are now well surveyed, e.g. [14,22,16,17].

However, despite the considerable literature relating QBFs and modal logic, their commonly employed complexity measures do not match up. In modal solving the key measure is that of *modal depth*, the maximal number of modalities in a path through the formula tree, cf. [22,14]. For QBFs the key measure is quantifier complexity, i.e. the number of alternations of $\exists$ and $\forall$ in a (prenex) QBF. While it is well-known that the alternation of quantifiers in QBFs corresponds precisely with the levels of the polynomial hierarchy [24,4], Halpern has showed in [10] that the validity problem for K (with any number of agents) for formulas with modal depth bounded by some constant $d \geq 2$ is in fact only *co***NP**-complete (see also [19]). [4]

It is this shortcoming of 'modal depth' that forms our principal motivation: can we identify a measure for modal formulas that coincides with quantifier complexity for QBFs, formally? In other words, can we find fragments of the modal logic K complete for each level of the polynomial hierarchy?

We answer this question positively in the present work by designing an *inverse* translation from true QBFs back into K. Our key idea is to encode modal provability itself as an alternating predicate, and analyse the alternation between 'invertible' and 'non-invertible' rules during proof search so as to delineate theorems according to the polynomial hierarchy. For this to work, we must first carefully devise a particular translation from QBFs to modal logics that is compatible with alternation in the aforementioned proof search predicate. In particular, composing our two translations yields an automorphism on (true) QBFs that *preserves* quantifier complexity.

---

[3] This translation was originally given for the Description Logic ALC, a notational variant of multi-agent modal logic K_m.

[4] Again, both Halpern and Nguyen study the satisfiability problem and state **NP**-completeness, whereas we study the dual problem of validity, inheriting *co***NP**-completeness from their results.

Related work and methodology

The idea of encoding proof search to obtain upper bounds for modal validity or satisfiability is not new, e.g. this is the approach taken in Halpern and Moses' 'guide' [11], only for a tableau system that is related to the sequent system we consider in this work. However none of the aforementioned works on complexity of modal logics give refinements of **PSPACE**-completeness to levels of the polynomial hierarchy, regardless of the method employed. [5]

This paper builds on recent work achieving similar delineations for multiplicative additive linear/affine logic [7,8] and fragments of intuitionistic logic [6], also well-known **PSPACE**-complete logics. Those works leveraged (alternative presentations of) *focussed* systems from structural proof theory (see, e.g., [1,15]), which elegantly control the alternation of invertible and non-invertible rules during proof search, the principal contributor to quantifier alternation in a proof search predicate.

(Normal) modal logics such as K (and, indeed, the entire 'modal $S5$ cube') have also recently received focussed treatments in the setting of *labelled* sequents [18] and *nested* sequents [5]. However nested and labelled systems, while admitting an elegant proof theory, do not enjoy terminating proof search per se, and thus are not adequate for obtaining alternating time bounds (see further discussion in Conclusions, Sec. 7). Instead we work with a standard cut-free sequent calculus for K and give a bespoke analysis of the proof search space according to invertible and non-invertible rules that suffices for our purposes.

2 Preliminaries on modal logic and (true) QBFs

Both of the logics we consider in this work are extensions of usual *classical propositional logic* (CPC). So we shall start by presenting CPC before duly extending it. Our exposition will be brief throughout this preliminary section, but we refer the reader to standard texts on modal logic [3] and proof theory [25] for further details.

We assume a countable set Var of *propositional variables*, written x, y, z etc. *(Propositional) formulas*, written A, B, C etc., are always in De Morgan normal form (i.e. with negation reduced to the variables) and are generated by the following grammar:

$$A, B \quad ::= \quad x \quad | \quad \bar{x} \quad | \quad (A \vee B) \quad | \quad (A \wedge B)$$

We assume usual bracketing conventions, i.e. omitting internal brackets of large disjunctions or conjunctions under associativity and external brackets too. Note that we may recover negation by extending the notation $\bar{\cdot}$ to all formulas by setting:

$$\bar{\bar{x}} := x \qquad \overline{A \vee B} := \bar{A} \wedge \bar{B} \qquad \overline{A \wedge B} := \bar{A} \vee \bar{B}$$

[5] Let us point out too that such delineations do not seem to be known for desription logics either, according to the online *Description Logic Complexity Navigator* [26].

We may employ standard logical abbreviations, e.g. writing $A \supset B$ for $\bar{A} \vee B$ and $A \equiv B$ for $(A \supset B) \wedge (B \supset A)$.

Semantics

A *(Boolean) assignment* is a map $\alpha : \mathsf{Var} \to \{0,1\}$. We write Ass for the set of all Boolean assignments. We may *evaluate* a Boolean formula with respect to an assignment by setting:

- $\alpha \vDash x$ if $\alpha(x) = 1$.
- $\alpha \vDash \bar{x}$ if $\alpha(x) = 0$.
- $\alpha \vDash A \vee B$ if $\alpha \vDash A$ or $\alpha \vDash B$.
- $\alpha \vDash A \wedge B$ if $\alpha \vDash A$ and $\alpha \vDash B$.

If $\alpha \vDash A$ then we say that α *satisfies* A. Note that, to evaluate a formula A with free variables among vector $\boldsymbol{x}$, we need only consider finite 'partial' assignments with domain containing $\boldsymbol{x}$.

If every $\alpha \in \mathsf{Ass}$ satisfies A, we say that A is *valid* and write simply $\vDash A$. The logic CPC is the set of valid propositional formulas.

Proof theory

CPC admits well-known Hilbert axiomatisations via axioms and rules, but we shall here rather present a standard (one-sided) sequent system for later use. The definitions and results of this subsection are based on analogous ones in [25].

A *sequent*, written Γ, Δ etc., is a multiset of formulas. We usually omit braces when writing such multisets and use the comma for multiset union, e.g. writing simply $A_1, \ldots, A_n$ for the multiset $\{A_1, \ldots, A_n\}$. We may interpret sequents as the disjunction of their formulas, in particular saying that a sequent Γ is valid or satisfied by an assignment just if $\bigvee \Gamma$ is.

Definition 2.1 (System for CPC**)** The system (one-sided, propositional) $\mathsf{G3c}$ is given by the following three rules:

$$\mathsf{id}\,\frac{}{\Gamma, x, \bar{x}} \qquad \vee\,\frac{\Gamma, A, B}{\Gamma, A \vee B} \qquad \wedge\,\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B}$$

Derivations and proofs are defined as usual for inference systems. We write $\vdash \Gamma$ if there is a $\mathsf{G3c}$-proof of the sequent Γ.

Remark 2.2 (Judgements) Note that we have indexed neither our satisfaction judgement $\vDash$ nor our provability judgement $\vdash$ by the logic or system in question. This is intentional since later logics and systems we consider are bona fide extensions of these. At the level of proofs, cut-freeness of our systems will guarantee that our notion of provability is unambiguous: any proof of Γ (in any of our systems) will contain only subformulas of Γ.

Proposition 2.3 (Soundness and completeness, CPC**)** *Let A be a propositional formula. $\vDash A$ if and only if $\vdash A$.*

2.1 Extension to modal logic

Modal formulas are generated by extending the grammar of propositional formulas by:

$$A, B \quad ::= \quad \dots \quad | \quad \Diamond A \quad | \quad \Box A$$

We extend the notation $\bar{A}$ to all modal formulas by setting,

$$\overline{\Diamond A} := \Box \bar{A} \qquad \overline{\Box A} := \Diamond \bar{A}$$

and admit logical abbreviations, e.g. $A \supset B$, as in propositional logic earlier.

We consider usual *relational semantics* for modal formulas, as found in, e.g., [3]. Note that we are only considering the basic normal modal logic K in this work.

Definition 2.4 (Relational semantics) A *(relational) structure* is a binary relation $R \subseteq W \times W$, where W is a set whose elements we call *worlds*. A *(relational) model* $\mathcal{M} = (W, R, \nu)$ is a relational structure $R \subseteq W \times W$ equipped with a *valuation* $\nu : W \to \mathsf{Ass}$. We sometimes write $|\mathcal{M}|$ for W (the *domain*).

Given a model $\mathcal{M} = (W, R, \nu)$ and some $w \in W$, we associate with the pair $(\mathcal{M}, w)$ the assignment $\nu(w)$. In this way, we define the judgement $\mathcal{M}, w \vDash A$ just like for CPC earlier, with the following additional clauses:

- $\mathcal{M}, w \vDash \Diamond A$ if there is $w' \in W$ s.t. wRw' and $\mathcal{M}, w' \vDash A$.
- $\mathcal{M}, w \vDash \Box A$ if, whenever wRw' for $w' \in W$, we have $\mathcal{M}, w' \vDash A$.

Similarly to propositional logic, we say that w *satisfies* A in $\mathcal{M}$ if $\mathcal{M}, w \vDash A$, and that A is *valid* (written $\vDash A$) if A is satisfied by every world in every model. The logic K is the set of valid modal formulas.

Let us write a, b, etc. for formulas of the form x or $\bar{x}$ (called *literals*).

Definition 2.5 (System for K) The calculus $\mathsf{G3k}$ is the extension of $\mathsf{G3c}$ by the rule:

$$\mathsf{k}\,\frac{\Gamma, A_i}{\boldsymbol{a}, \Diamond\Gamma, \Box A_0, \dots, \Box A_{n-1}}\, i < n$$

We again write $\vdash A$ if there is a $\mathsf{G3k}$-proof of A.

As expected, Prop. 2.3 extends to modal formulas:

Proposition 2.6 (Soundness and completeness, K) *Let A be a modal formula. $\vDash A$ if and only if $\vdash A$.*

2.2 Extension to second-order propositional logic

Quantified Boolean Formulas (QBFs) are generated by extending the grammar of propositional formulas by:

$$A, B \quad ::= \quad \dots \quad | \quad \exists x A \quad | \quad \forall x A$$

We refer to 'free' and 'bound' variables for QBFs in the usual way and write $\mathrm{FV}(A)$ for the set of free variables in the QBF A. A QBF A is *closed* if

$\mathrm{FV}(A) = \varnothing$, i.e. A has no free variables. We extend the notation $\bar{A}$ to all QBFs A by setting,

$$\overline{\exists x A} := \forall x \bar{A} \qquad \overline{\forall x A} := \exists x \bar{A}$$

and admit logical abbreviations, e.g. $A \supset B$, as in propositional logic earlier.

We shall often further assume that each quantifier of a QBF binds a distinct variable. Finally, we shall typically only work with QBFs in *prenex normal form*, i.e. of the form $Q_1 x_1 \dots Q_n x_n A$, for some $n \geq 0$, quantifiers $Q_1, \dots, Q_n$, and A quantifier-free (so a propositional formula). It is well known that each QBF is equivalent to one in prenex normal form, in terms of the following semantics:

Definition 2.7 (QBF semantics) Given an assignment α and a QBF A, we define the judgement $\alpha \vDash A$ just like for CPC, with the following additional clauses,

- $\alpha \vDash \exists x A$ if $\alpha[x \mapsto 0] \vDash A$ or $\alpha[x \mapsto 1] \vDash A$.
- $\alpha \vDash \forall x A$ if $\alpha[x \mapsto 0] \vDash A$ and $\alpha[x \mapsto 1] \vDash A$.

where we write $\alpha[x \mapsto i]$ for the assignment defined just like α but mapping x to $i \in \{0, 1\}$. We duly extend the terminology 'satisfies' and 'valid' from propositional formulas to arbitrary QBFs. A closed QBF is called simply *true* if it is valid (equivalently, satisfiable). The logic CPC2 is the set of valid QBFs.

We will not actually work with proofs for CPC2 in this work, but we include a system for it below for completeness.

Definition 2.8 (System for CPC2**)** The calculus $\mathsf{G3c2}$ is the extension of $\mathsf{G3c}$ by the following rules,

$$\exists \frac{\Gamma, \exists x A, A[B/x]}{\Gamma, \exists x A} \qquad \forall \frac{\Gamma, A[y/x]}{\Gamma, \forall x A}\ y \text{ not free in } \Gamma$$

where we write $A[B/x]$ for the QBF resulting from substituting each free occurrence of x in A by the QBF B.

Again, Prop. 2.3 extends to QBFs as expected:

Proposition 2.9 (Soundness and completeness, CPC2**)** *Let A be a QBF. $\vDash A$ if and only if $\vdash A$.*

2.3 Some examples and comments on proof search

Let us take the time to consider some examples of proofs in our systems, in particular to highlight some of the proof search dynamics that will later come into play. We intentionally choose rather simple validities/proofs.

First, consider the modal formula:

$$\Box x \vee \Box y \vee (\Diamond(x \supset y) \wedge \Diamond(y \supset x)) \tag{1}$$

It is not hard to see that this is valid in K, in particular thanks to the following proof in $\mathsf{G3k}$ under Prop. 2.6:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{x, \bar{x}, y}\,\mathsf{id}}{x, x \supset y}\,\vee}{\Box x, \Box y, \Diamond(x \supset y)}\,\mathsf{k} \qquad \dfrac{\dfrac{\dfrac{}{y, \bar{y}, x}\,\mathsf{id}}{y, y \supset x}\,\vee}{\Box x, \Box y, \Diamond(y \supset x)}\,\mathsf{k}}{\dfrac{\Box x, \Box y, \Diamond(x \supset y) \wedge \Diamond(y \supset x)}{\Box x \vee \Box y \vee (\Diamond(x \supset y) \wedge \Diamond(y \supset x))}\,2\vee}\,\wedge \tag{2}
$$

Viewing proof search here as an alternating predicate, we can see the $\wedge$ rule as *universal branching*: every premiss must be valid. The k rule, on the other hand, is an example of *existential branching*: some choice of premiss must be valid. Finally, the $\vee$ and id rules may be viewed as 'deterministic': they have no computational cost. We will start making these classifications formal in Sec. 5.

We may 'derive' from (1) a valid QBF by mimicking its formula structure,

$$\forall x A(x) \vee \forall y B(y) \vee (\exists z(A(z) \supset B(z)) \wedge \exists z'(B(z') \supset A(z')))$$

for some arbitrary formulas $A(x)$ and $B(y)$ (possibly with further free variables). This may be put into the following prenex normal form:

$$\forall x \forall y \exists z \exists z' (A(x) \vee B(y) \vee ((A(z) \supset B(z)) \wedge (B(z') \supset A(z'))) \tag{3}$$

In this case, notice that the quantifier prefix above matches the aforementioned universal-existential-branching exhibited during proof search for (1). This is coincidental for this particular case, since we have not yet properly fixed an appropriate translation from modal formulas to QBFs, but arriving at a formal such correspondence constitutes the principal aim of this work.

3 Alternating complexity and some decision problems

We shall assume basic familiarity with (non)deterministic Turing machines (with oracles) and their time and space complexity, for which there are several basic references available, e.g. [2]. Let us also point out that this preliminary section is similar to the analogous one in [8], where there are further details.

For the sake of formality, all languages we consider will be subsets of $\{0,1\}^*$. Throughout we may consider larger (but finite) alphabets than $\{0,1\}$, but these should always be assumed to be adequately coded in binary.

3.1 Some complexity classes

Let us fix **P** (and **PSPACE**) as the class of languages accepted by a deterministic Turing machine in polynomial time (resp., space).

$\mathbf{NP}(L)$ is the class of languages accepted by a nondeterministic Turing Machine with access to an oracle for the language L in polynomial time. Intuitively, such a machine acts just like a usual nondeterministic machine but

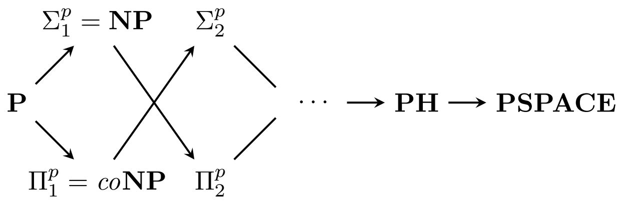


Fig. 1. Relationships between complexity classes. An arrow $\mathcal{C} \to \mathcal{D}$ means the complexity class $\mathcal{C}$ is contained in the complexity class $\mathcal{D}$.

may at any point query (in constant time) whether a string is in L or not. Given a class $\mathcal{C}$ of languages, $\mathbf{NP}(\mathcal{C}) := \bigcup_{L \in \mathcal{C}} \mathbf{NP}(L)$. Finally, given a class $\mathcal{C}$ of languages, $co\mathcal{C}$ is the set of complements of $\mathcal{C}$, i.e. $co\mathcal{C} := \{\{0,1\}^* \setminus L : L \in \mathcal{C}\}$.

Definition 3.1 (Polynomial hierarchy) We define the classes Σ_n^p and Π_n^p, for $n \geq 1$, as follows:

- $\Sigma_1^p := \mathbf{NP}$
- $\Sigma_{n+1}^p := \mathbf{NP}(\Sigma_n^p)$
- $\Pi_n^p := co\Sigma_n^p$

We write $\mathbf{PH} := \bigcup_{n \geq 1} \Sigma_n^p$ (equivalently, $\bigcup_{n \geq 1} \Pi_n^p$).

The following relationships are (almost) immediate from definitions:

(i) $\mathbf{P} \subseteq \mathbf{NP} \cap co\mathbf{NP}$.

(ii) $\Sigma_n^p \subseteq \Pi_{n+1}^p$ and $\Pi_n^p \subseteq \Sigma_{n+1}^p$.

(iii) $\mathbf{PH} \subseteq \mathbf{PSPACE}$.

Formally speaking, the final point is not quite immediate from definitions but follows since $\mathbf{NP}(\mathbf{PSPACE}) \subseteq \mathbf{NPSPACE} \subseteq \mathbf{PSPACE}$ by Savitch's theorem and since **PSPACE**, being a deterministic class, is closed under complementation. These inclusions are visualised in Fig. 1.

3.2 Complexity of QBFs and modal logic

As we have already mentioned, the complexity of checking validity for both modal formulas and QBFs are well-known. The results we state in this subsection are taken from (or implied by) [13] and [4].

Proposition 3.2 K *and* CPC2 *are* **PSPACE**-*complete.*

Note that the same complexity bound holds for the corresponding satisfiability problems, thanks to closure of **PSPACE** under complements, by a usual reduction: A is satisfiable (or valid) if and only if $\bar{A}$ is not valid (resp., satisfiable). On the other hand, the complexity of model checking for the two logics is (presumably) rather different:

Proposition 3.3 (Complexity of satisfaction) *We have the following:*

(i) *Checking* $\mathcal{M}, w \vDash A$ *is in* $\mathbf{P}$*, for a finite model* $\mathcal{M}$*, some* $w \in |\mathcal{M}|$ *and* A *a modal formula.*

(ii) *Checking* $\alpha \vDash A$ *is* **PSPACE***-complete, for a finite partial assignment* α *with domain* $\boldsymbol{x}$ *and* A *a QBF with free variables among* $\boldsymbol{x}$*.*

In the case of QBFs we can give a refinement that will also refine **PSPACE**-completeness of CPC2, and that we shall exploit later in Sec. 5 to reduce proof search to QBF satisfaction. First, we recall another hierarchy:

Definition 3.4 (QBF hierarchy) We define the following classes of QBFs (in prenex normal form):

- $\Sigma^q_0 = \Pi^q_0$ is the class of quantifier-free QBFs (i.e. propositional formulas).
- $\Sigma^q_{n+1} := \{\exists \boldsymbol{x} A : A \in \Pi^q_n\}$
- $\Pi^q_{n+1} := \{\forall \boldsymbol{x} A : A \in \Sigma^q_n\}$

It is well-known that the levels of the QBF hierarchy above match up precisely with the levels of the polynomial hierarchy from Dfn. 3.1:

Proposition 3.5 *We have the following, for* $n \geq 1$*:*

(i) $\{(\alpha, A) : A \in \Sigma^q_n, \alpha : \mathrm{FV}(A) \to \{0,1\}\}$ *is* Σ^p_n*-complete.*

(ii) $\{(\alpha, A) : A \in \Pi^q_n, \alpha : \mathrm{FV}(A) \to \{0,1\}\}$ *is* Π^p_n*-complete.*

An immediate consequence of this, by way of Boolean simplification under an assignment, is the following well-known delineation of CPC2 according to levels of **PH**:

Corollary 3.6 *We have the following, for* $n \geq 1$*:*

(i) *The set of true* Σ^q_n *sentences is* Σ^p_n*-complete.*

(ii) *The set of true* Π^q_n *sentences is* Π^p_n*-complete.*

One of the points of this work is to establish a similar such delineation for modal logic K.

4 From QBFs to K

In this section we present our first translation, from QBFs to modal formulas, inspired somewhat by Statman's translation from QBFs into intuitionistic propositional logic [23], only avoiding the need for 'extension' variables. The translation is polynomial-time computable, and induces an encoding from CPC2 to K, i.e. it is a polynomial-time reduction from CPC2 to K.

Let us write $e(A)$ for the number of existential quantifiers in a QBF A (typically in prenex form).

Definition 4.1 (The $\cdot^\bullet$-translation) For a prenex QBF A we define a modal formula $A^\bullet$ as follows:

- $A^\bullet := A$ if A is quantifier-free.
- $(\exists x A)^\bullet := (\Diamond \Box^{e(A)} x \wedge \Diamond \Box^{e(A)} \bar{x}) \supset \Diamond A^\bullet$

- $(\forall x A)^\bullet := (\Box^{e(A)} x \vee \Box^{e(A)} \bar{x}) \supset A^\bullet$

Let us point out that it is clear that $\cdot^\bullet$ is polynomial-time computable.

Example 4.2 Consider a (not necessarily valid) QBF $\exists x \forall y \exists z A$, for some arbitrary quantifier-free formula A, possibly with free variables. We have:

$$\begin{aligned}
A^\bullet &= A \\
(\exists z A)^\bullet &= (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A \\
(\forall y \exists z A)^\bullet &= (\Box y \vee \Box \bar{y}) \supset (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A \\
(\exists x \forall y \exists z A)^\bullet &= (\Diamond \Box x \vee \Diamond \Box \bar{x}) \supset \Diamond((\Box y \vee \Box \bar{y}) \supset (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A)
\end{aligned}$$

Theorem 4.3 *Let A be a closed QBF. $\vDash A$ if and only if $\vDash A^\bullet$.*

To prove this we need an intermediate result. First we set up some notation. Let us write $x^0 := \bar{x}$ and $x^1 := x$ and, for an assignment α, simply $x^\alpha := x^{\alpha(x)}$. Given variables $\boldsymbol{x} = x_1, \dots, x_k$ we shall write $\Box^n \boldsymbol{x}^\alpha := \Box^n x_1^\alpha, \dots, \Box^n x_k^\alpha$ and similarly $\Diamond^n \boldsymbol{x}^\alpha := \Diamond^n x_1^\alpha, \dots, \Diamond^n x_k^\alpha$. Finally, as a temporary abuse of notation to facilitate readability, we shall write $A_1, \dots, A_n \vdash B$ if $\vdash \bar{A}_1, \dots, \bar{A}_n, B$.[6] Instead of provability, here '$\vdash$' should be (temporarily) considered as the 'sequent arrow' in a two-sided calculus, which is a common notation.

Lemma 4.4 *Let A be a prenex QBF with free variables among $\boldsymbol{x}$ and α an assignment. We have that $\alpha \vDash A$ if and only if $\Box^{e(A)} \boldsymbol{x}^\alpha \vdash A^\bullet$.*

Let us point out that this lemma holds crucially due to the 'balanced' structure of the $\cdot^\bullet$-translation, in terms of modal depth. Before giving the proof, let us revisit the earlier example.

Example 4.5 Consider again a QBF of the form $\exists x \forall y \exists z A$ like in Ex. 4.2, now setting $A = x \wedge (y \equiv z)$ so that we have a validity. Notice that we have both $x, y, z \vdash A$ and $x, \bar{y}, \bar{z} \vdash A$ by some propositional rules. Now several examples of Lem. 4.4 are found within the following proof with grey background corresponding, respectively, to the $\cdot^\bullet$ instances from Ex. 4.2:

$$\begin{prooftree}
\AxiomC{}
\doubleLine
\UnaryInfC{$x, y, z \vdash A$}
\LeftLabel{k}
\UnaryInfC{$\Box x, \Box y, \Diamond z, \Diamond \bar{z} \vdash \Diamond A$}
\LeftLabel{$2\vee$}
\UnaryInfC{$\Box x, \Box y \vdash (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A$}
\AxiomC{}
\doubleLine
\UnaryInfC{$x, \bar{y}, \bar{z} \vdash A$}
\LeftLabel{k}
\UnaryInfC{$\Box x, \Box \bar{y}, \Diamond z, \Diamond \bar{z} \vdash \Diamond A$}
\LeftLabel{$2\vee$}
\UnaryInfC{$\Box x, \Box \bar{y} \vdash (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A$}
\LeftLabel{$\wedge$}
\BinaryInfC{$\Box x, \Box y \vee \Box \bar{y} \vdash (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A$}
\LeftLabel{$\vee$}
\UnaryInfC{$\Box x \vdash (\Box y \vee \Box \bar{y}) \supset (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A$}
\LeftLabel{k}
\UnaryInfC{$\Diamond \Box x \vdash \Diamond((\Box y \vee \Box \bar{y}) \supset (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A)$}
\LeftLabel{$\vee$}
\UnaryInfC{$\vdash \Diamond \Box x \supset \Diamond((\Box y \vee \Box \bar{y}) \supset (\Diamond z \wedge \Diamond \bar{z}) \supset \Diamond A)$}
\end{prooftree}$$

Proof of Lem. 4.4. We proceed by induction on the number of quantifiers in A. The base case, when A is quantifier-free, is trivial.

[6] Note that this really is an abuse of notation as the deduction theorem fails for K.

In the universal case we have:

$$
\begin{array}{lll}
& \alpha \vDash \forall x A & \\
\iff & \alpha[x \mapsto 0] \vDash A \quad\quad\quad \text{and } \alpha[x \mapsto 1] \vDash A & \text{by definition of } \vDash \\
\iff & \Box^{e(A)}\boldsymbol{x}^\alpha, \Box^{e(A)}\bar{x} \vdash A^\bullet \text{ and } \Box^{e(A)}\boldsymbol{x}^\alpha, \Box^{e(A)}x \vdash A^\bullet & \text{by inductive hypothesis} \\
\iff & \Box^{e(A)}\boldsymbol{x}^\alpha, \Box^{e(A)}\bar{x} \vee \Box^{e(A)}x \vdash A^\bullet & \text{by } \wedge \text{ rule} \\
\iff & \Box^{e(A)}\boldsymbol{x}^\alpha \vdash (\Box^{e(A)}\bar{x} \vee \Box^{e(A)}x) \supset A^\bullet & \text{by } \vee \text{ rule} \\
\iff & \Box^{e(\forall x A)}\boldsymbol{x}^\alpha \vdash (\forall x A)^\bullet & \text{by definition of } \cdot^\bullet
\end{array}
$$

The existential case is a little more subtle, so we treat the two directions separately.

$$
\begin{array}{rlll}
\alpha \vDash \exists x A & \implies & \alpha[x \mapsto i] \vDash A & \text{for some } i \in \{0,1\} \\
& \implies & \Box^{e(A)}\boldsymbol{x}^\alpha, \Box^{e(A)}x^i \vdash A^\bullet & \text{by inductive hypothesis} \\
& \implies & \Box^{e(A)+1}\boldsymbol{x}^\alpha, \Diamond\Box^{e(A)}x, \Diamond\Box^{e(A)}\bar{x} \vdash \Diamond A^\bullet & \text{by } \mathsf{k} \\
& \implies & \Box^{e(\exists x A)}\boldsymbol{x}^\alpha \vdash (\Diamond\Box^{e(A)}x \wedge \Diamond\Box^{e(A)}\bar{x}) \supset \Diamond A^\bullet & \text{by } \vee \text{ rule} \\
& \implies & \Box^{e(\exists x A)}\boldsymbol{x}^\alpha \vdash (\exists x A)^\bullet & \text{by definition of } \cdot^\bullet
\end{array}
$$

For the other direction the steps are quite similar, but the justifications are different:

$$
\begin{array}{lll}
& \Box^{e(\exists x A)}\boldsymbol{x}^\alpha \vdash (\exists x A)^\bullet & \\
\implies & \Box^{e(A)+1}\boldsymbol{x}^\alpha \vdash (\Diamond\Box^{e(A)}x \wedge \Diamond\Box^{e(A)}\bar{x}) \supset \Diamond A^\bullet & \text{by definition of } \cdot^\bullet \\
\implies & \Box\Box^{e(A)}\boldsymbol{x}^\alpha, \Diamond\Box^{e(A)}x, \Diamond\Box^{e(A)}\bar{x} \vdash \Diamond A^\bullet & \text{proof must end with } \vee \text{ steps} \\
\implies & \Box^{e(A)}\boldsymbol{x}^\alpha, \Box^{e(A)}x^i \vdash A^\bullet & \text{proof must end with } \mathsf{k} \\
\implies & \alpha[x \mapsto i] \vDash A & \text{by inductive hypothesis} \\
\implies & \alpha \vDash \exists x A &
\end{array}
$$

Note that we have alluded several times to proof search in system G3k above, crucially taking advantage of its cut-free nature. ☐

From here it is easy to deduce the main result of this section:

Proof of Thm. 4.3. Follows directly as a special case of Lem. 4.4, since A has no free variables, under soundness and completeness, Prop. 2.6. ☐

5 Proof search as an alternating time predicate

In this section we build up some of the theory of alternation complexity of proof search in G3k that we will need to ultimately define our 'inverse' translation to $\cdot^\bullet$. In particular, we encode proof search in G3k as a family of predicates, parametrised by their corresponding level in the polynomial hierarchy, and thus inducing polynomial-size families of QBFs computing proof search in G3k.

Some of the notions and results here are similar to ones appearing in [7,8], but are formulated necessarily bespoke to the calculus G3k.

5.1 The proof search hierarchy

First, in what follows, we shall classify the rules of G3k as follows:

- **Deterministic.** The id and $\vee$ rules.

- **Nondeterministic.** The k rule.
- **Co-nondeterministic.** The $\wedge$ rule.

The nomenclature above is suggestive, indicating the alternation cost of each rule application during proof search, as we previously hinted at in Subsec. 2.3. From here we can delineate the proof search space according to the number of alternations between nondeterministic and co-nondeterministic phases of rules.

Definition 5.1 (Proof search hierarchy) We define the classes Σ^s_n and Π^s_n of (provable) sequents as follows:

- $\Sigma^s_0 = \Pi^s_0$ is the class of sequents provable using only deterministic rules.
- Σ^s_{n+1} is the class of sequents derivable from sequents in Π^s_n using only deterministic and nondeterministic rules.
- Π^s_{n+1} is the class of sequents derivable from sequents in Σ^s_n using only deterministic and co-nondeterministic rules.

Example 5.2 Revisiting the examples from Subsec. 2.3, we have that the modal formula (1) is in Π^s_2. To see this, let us inspect the proof of (1) in (2). $\Box x, \Box y, \Diamond(x \supset y)$ and $\Box x, \Box y, \Diamond(y \supset x)$ are in Σ^s_1, since their subproofs consist of only $\mathsf{id}, \vee$ rules (deterministic) and k rules (nondeterministic). From these two sequents (1) is derived using only $\vee$ rules (deterministic) and $\wedge$ rules (co-nondeterministic), and so indeed (1) $\in \Pi^s_2$.

Using similar methods to those in [8,7], we may prove the following:

Theorem 5.3 *For $n \geq 0$ we have:*

(i) Σ^s_n *is decidable in* Σ^p_n*; and,*

(ii) Π^s_n *is decidable in* Π^p_n*,*

Proof. We proceed by induction on $n \geq 0$.

Let $\Gamma \in \Sigma^s_0$, then there exists a sequent Γ' and a variable x such that Γ can be derived from the axiom $\Gamma', x, \bar{x}$ using only disjuction rules.

$$\cfrac{\mathsf{id}\ \overline{\Gamma', x, \bar{x}}}{\Gamma}\ \vee \|$$

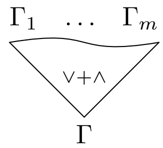
This means that to check whether a given sequent is in Σ^s_0, given that the disjuction rule is invertible, we can apply it maximally (linearly many steps in the size of Γ) and check whether we end up on a sequent that contains $x, \bar{x}$ for some variable x. Therefore, Σ^s_0 (and Π^s_0) provability is decidable in $\mathbf{P}$.

$\Gamma \in \Sigma^s_{n+1}$ just if there exists a sequent $\Gamma_m, A_m \in \Pi^s_n$ such that:

$$
\begin{array}{c}
k\,\dfrac{\Gamma_m, A_m}{\boldsymbol{a_m}, \Diamond\Gamma_m, \Box\Delta_m}\; A_m \in \Delta_m \\
\| \\
\vdots \\
\| \\
k\,\dfrac{\Gamma_1, A_1}{\boldsymbol{a_1}, \Diamond\Gamma_1, \Box\Delta_1}\; A_1 \in \Delta_1 \\
\vee \, \| \\
\Gamma
\end{array}
$$

This means that to check whether a given sequent Γ is in Σ^s_{n+1} we may guess the (polynomial-size) configuration above and then check that $\Gamma_m, A_m \in \Pi^s_n$. By the inductive hypothesis the latter is possible in Π^p_n and so the entire procedure is in $\mathbf{NP}(\Pi^p_n) = \Sigma^p_{n+1}$ as required.

$\Gamma \in \Pi^s_{n+1}$ just if there exist some sequents $\Gamma_i \in \Sigma^s_n$ such that



Note that we may assume that $\vee$ and $\wedge$ rules are applied maximally above as both rules are invertible and simplify sequents, bottom-up. Now, Γ is *not* Π^s_{n+1} provable just if there exists a (polynomial-size) branch in the finite derivation above from Γ to some Γ_i that is *not* Σ^s_n-provable. By induction hypothesis, non-Σ^s_n-provability is checkable in Π^p_n, and so we conclude that non-Π^s_{n+1}-provability is in $\mathbf{NP}(\Pi^p_n) = \Sigma^p_{n+1}$. Finally, by definition of complements, this means that Π^s_{n+1}-provability is in Π^p_{n+1}, as required. $\Box$

An immediate consequence of the Thm. 5.3 above, under Prop. 3.5, is:

Corollary 5.4 *For $k \geq 1$, there are polynomial-size Σ^q_k-QBFs Σ^s_k-Prov_n, and Π^q_k-QBFs Π^s_k-Prov_n, computing Σ^s_k-provability, and Π^s_k-provability respectively, on formulas A with $|A| \leq n$.*

5.2 (Co-)nondeterministic complexity of proof search

We now give a complexity measure for (provable) sequents that coincides with our proof search hierarchy earlier, but that later admits a feasible 'approximation' that will facilitate our inverse translation from modal formulas to QBFs.

Definition 5.5 Let $\mathcal{D}$ be a proof in $\mathsf{G3k}$. The *nondeterministic complexity* $\sigma(\mathcal{D})$ (resp. *co-nondeterministic complexity* $\pi(\mathcal{D})$) is the maximum number of alternations between k and $\wedge$ steps **bottom-up** along any branch in $\mathcal{D}$ **starting with the k steps** (resp. **starting with the $\wedge$ steps**); in particular,

- $\sigma(\mathcal{D}) = \pi(\mathcal{D}) = 0$, if $\mathcal{D}$ contains just $\vee$ steps,

- $\sigma(\mathcal{D}) = 1$, but $\pi(\mathcal{D}) = 2$, if $\mathcal{D}$ contains just k and $\vee$ steps,
- $\pi(\mathcal{D}) = 1$, but $\sigma(\mathcal{D}) = 2$, if $\mathcal{D}$ contains just $\wedge$ and $\vee$ steps.

For a modal sequent Γ, $\sigma(\Gamma)$ (resp. $\pi(\Gamma)$) is the least $n \in \mathbb{N}$ such that there is a proof of Γ in $\mathsf{G3k}$ with $\sigma(\Gamma) = n$ (resp. $\pi(\Gamma) = n$).

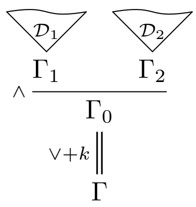
It is not hard to see the following:

Proposition 5.6 *Let Γ be a modal sequent. For $n \in \mathbb{N}$, we have that:*

(i) Γ *is in* Σ^s_n *iff* $\sigma(\Gamma) \leq n$

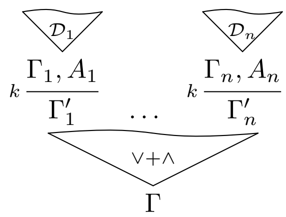
(ii) Γ *is in* Π^s_n *iff* $\pi(\Gamma) \leq n$

Proof sketch Both directions are proved by mutual induction on $n \geq 1$. We present only the right-to-left direction. If $\sigma(\Gamma) \leq n$, there exists a proof $\mathcal{D}$ of Γ such that $\sigma(\mathcal{D}) \leq n$, that is a proof of the form,



such that the maximum number of $\wedge/k$-alternations along any branch is at most n. Hence, the proof $\mathcal{D}_0$ of Γ_0 is such that $\pi(\mathcal{D}_0) \leq n-1$. By (IH), $\Gamma_0 \in \Pi^s_{n-1}$, so by Def. 5.1, $\Gamma \in \Sigma^s_n$.

If $\pi(\Gamma) \leq n$, there exists a proof $\mathcal{D}$ of Γ such that $\pi(\mathcal{D}) \leq n$, that is a proof of the form,



with $\Gamma'_i = \boldsymbol{a_i}, \Diamond\Gamma_i, \Box\Delta_i$ and $A_i \in \Delta_i$, such that the maximum number of $\wedge/k$-alternations along any branch is at most n. Hence, the proofs $\mathcal{D}'_i$ of Γ'_i are such that $\sigma(\mathcal{D}'_i) \leq n-1$. By (IH), $\Gamma'_i \in \Sigma^s_{n-1}$, so by Def. 5.1, $\Gamma \in \Pi^s_n$. □

5.3 Feasibly approximating (co-)nondeterministic complexity

The complexity measures σ and π defined in the previous section are evidently costly to compute, since they require us to consider all possible proofs of a sequent. Thus, despite the characterisation of Prop. 5.6, we have not yet arrived at a bona fide polynomial-time encoding from K to CPC2. Instead, we are able to define feasible 'over-approximations' for them directly on the structure of the sequent, without having to look at a particular proof.

Definition 5.7 (Approximation of (co-)nondeterministic complexity) We define the functions $\lceil\sigma\rceil$ and $\lceil\pi\rceil$ on modal sequents as follows:

$$\begin{aligned}
\lceil\sigma\rceil(A) &:= 1 \quad \text{if } A \text{ is propositional} \\
\lceil\sigma\rceil(\Gamma, \boldsymbol{a}) &:= \lceil\sigma\rceil(\Gamma) \\
\lceil\sigma\rceil(\Gamma, A \vee B) &:= \lceil\sigma\rceil(\Gamma, A, B) \\
\lceil\sigma\rceil(\Gamma, A \wedge B) &:= 1 + \lceil\pi\rceil(\Gamma, A \wedge B) \\
\lceil\sigma\rceil(\Diamond\Gamma, \Box\Delta) &:= \lceil\sigma\rceil(\Gamma, A) \quad \text{for } A \in \Delta \text{ with maximal } \lceil\sigma\rceil(A)
\end{aligned}$$

$$\begin{aligned}
\lceil\pi\rceil(A) &:= 1 \quad \text{if } A \text{ is propositional} \\
\lceil\pi\rceil(\Gamma, \boldsymbol{a}) &:= \lceil\pi\rceil(\Gamma) \\
\lceil\pi\rceil(\Gamma, A \vee B) &:= \lceil\pi\rceil(\Gamma, A, B) \\
\lceil\pi\rceil(\Gamma, A \wedge B) &:= \begin{cases} \lceil\pi\rceil(\Gamma, A) & \lceil\pi\rceil(A) \geq \lceil\pi\rceil(B) \\ \lceil\pi\rceil(\Gamma, B) & \text{otherwise} \end{cases} \\
\lceil\pi\rceil(\Diamond\Gamma, \Box\Delta) &:= 1 + \lceil\sigma\rceil(\Diamond\Gamma, \Box\Delta)
\end{aligned}$$

In the above definition, note that some choices are arbitrary, in particular when choosing a disjunction $A \vee B$ or conjunction $A \wedge B$ to evaluate $\lceil\sigma\rceil$ or $\lceil\pi\rceil$. For well-definedness, we could impose that the 'smallest' such formula is chosen, according to some global well-order on formulas. As long as this well-order is polynomial-time computable, so are the functions $\lceil\sigma\rceil$ and $\lceil\pi\rceil$.

However, it turns out that the definitions of $\lceil\sigma\rceil$ and $\lceil\pi\rceil$ are *independent* of such a choice, as was proved for similar measures in [8]. We can exploit this to show that $\lceil\sigma\rceil$ and $\lceil\pi\rceil$ indeed define over-approximations of σ and π:

Proposition 5.8 *For a modal sequent* Γ*,* $\lceil\sigma\rceil(\Gamma) \geq \sigma(\Gamma)$ *and* $\lceil\pi\rceil(\Gamma) \geq \pi(\Gamma)$*.*

Proof sketch Suppose $\sigma(\Gamma) = k$ and let $\mathcal{D}$ be a proof of Γ with $\sigma(\mathcal{D}) = k$. We may evaluate $\lceil\sigma\rceil(\Gamma)$ by following the choices of principal formula in $\mathcal{D}$. □

The alternating behaviour of $\lceil\sigma\rceil$ on conjunctions and of $\lceil\pi\rceil$ on modalities allows us to recover the following expected relations, again taking advantage of the fact that $\lceil\sigma\rceil$ and $\lceil\pi\rceil$ are independent of the underlying order on formulas:

Lemma 5.9 *Let* A *be a QBF and* Γ *be any modal sequent.*

(i) *If* $A \in \Pi^q_k$ *for* $k \geq 1$*, then* $\lceil\sigma\rceil(\Gamma, A^\bullet) = 1 + \lceil\pi\rceil(\Gamma, A^\bullet)$*.*

(ii) *If* $A \in \Sigma^q_k$ *for* $k \geq 1$*, then* $\lceil\pi\rceil(\Gamma, A^\bullet) = 1 + \lceil\sigma\rceil(\Gamma, A^\bullet)$*.*

Proof. If $A = \forall x B$, then $A^\bullet = (\Box^{e(B)}\bar{x} \vee \Box^{e(B)}x) \supset B^\bullet$ and hence:

$$\begin{aligned}
\lceil\sigma\rceil(\Gamma, A^\bullet) &= \lceil\sigma\rceil(\Gamma, \Diamond^{e(B)}x \wedge \Diamond^{e(B)}\bar{x}, B^\bullet) \\
&= 1 + \lceil\pi\rceil(\Gamma, \Diamond^{e(B)}x \wedge \Diamond^{e(B)}\bar{x}, B^\bullet) \\
&= 1 + \lceil\pi\rceil(\Gamma, A^\bullet)
\end{aligned}$$

If $A = \exists x B$, then $A^\bullet = (\Diamond\Box^{e(B^\bullet)}\bar{x} \wedge \Diamond\Box^{e(B)}x) \supset \Diamond B^\bullet$ and hence:

$$\begin{aligned}
\lceil\pi\rceil(\Gamma, A^\bullet) &= \lceil\pi\rceil(\Gamma, \Box\Diamond^{e(B)}x, \Box\Diamond^{e(B)}\bar{x}, \Diamond B^\bullet) \\
&= 1 + \lceil\sigma\rceil(\Gamma, \Box\Diamond^{e(B)}x, \Box\Diamond^{e(B)}\bar{x}, \Diamond B^\bullet) \\
&= 1 + \lceil\sigma\rceil(\Gamma, A^\bullet) \qquad \Box
\end{aligned}$$

Using this Lemma, we may conduct an inductive argument to show that our approximations accurately follow the QBF hierarchy, with respect to $\cdot^\bullet$.

Lemma 5.10 *Let A be a QBF, for any vector of literals $\boldsymbol{a}$:*

- *If $A \in \Pi^q_k$, then $\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) = k$*
- *If $A \in \Sigma^q_k$, then $\lceil\sigma\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) = k$*

Proof. By induction on $k \geq 1$.

Let $A = \forall x_1 \dots \forall x_n B$ with $B \in \Sigma^q_{k-1}$ and $A^\bullet = (\Box^{e(B)}\bar{x}_1 \vee \Box^{e(B)}x_1) \supset (\forall x_2 \dots \forall x_n B)^\bullet$, hence

$$\begin{aligned}
&\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet)\\
=\ &\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, \Diamond^{e(B)}x_1 \wedge \Diamond^{e(B)}\bar{x}_1, (\forall x_2 \dots \forall x_n B)^\bullet)\\
=\ &\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, \Diamond^{e(B)}x_1, (\forall x_2 \dots \forall x_n B)^\bullet)\\
=\ &\dots = \lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, \Diamond^{e(B)}x_1, \dots, \Diamond^{e(B)}x_n, B^\bullet)\\
=\ &\lceil\pi\rceil(\Diamond^{e(B)}\boldsymbol{a}, \Diamond^{e(B)}\boldsymbol{x}, B^\bullet) \text{ as } e(A) = e(B)
\end{aligned}$$

as we always choose x_i in the $\wedge$ case of $\lceil\pi\rceil$ in Def. 5.7 without loss of generality.

This gives us, in the inductive case when $k > 1$ that :

$$\begin{aligned}
\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) &= 1 + \lceil\sigma\rceil(\Diamond^{e(B)}\boldsymbol{a}, \Diamond^{e(B)}\boldsymbol{x}, B^\bullet) \text{ by Lemma 5.9}\\
&= 1 + (k-1) = k \text{ by (IH) applied to } B \in \Sigma^q_{k-1}
\end{aligned}$$

and in the base case when $k = 1$ that:

$$\lceil\pi\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) = \lceil\pi\rceil(\boldsymbol{a}, \boldsymbol{x}, B) = 1 \text{ as } B \text{ is quantifier-free}$$

Let $A = \exists x_1 \dots \exists x_n B$ with $B \in \Pi^q_{k-1}$ and $A^\bullet = (\Diamond\Box^{e(B)+n-1}\bar{x}_1 \wedge \Diamond\Box^{e(B)+n-1}x_1) \supset \Diamond(\exists x_2 \dots \exists x_n B)^\bullet$, hence

$$\begin{aligned}
&\lceil\sigma\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet)\\
=\ &\lceil\sigma\rceil(\Diamond^{e(A)}\boldsymbol{a}, \Box\Diamond^{e(B)+n-1}x_1, \Box\Diamond^{e(B)+n-1}\bar{x}_1, \Diamond(\exists x_2 \dots \exists x_n B)^\bullet)\\
=\ &\lceil\sigma\rceil(\Diamond^{e(A)-1}\boldsymbol{a}, \Diamond^{e(B)+n-1}x_1, (\exists x_2 \dots \exists x_n B)^\bullet)\\
=\ &\dots = \lceil\sigma\rceil(\Diamond^{e(A)-n}\boldsymbol{a}, \Diamond^{e(B)}x_1, \dots, \Diamond^{e(B)}x_n, B^\bullet)\\
=\ &\lceil\sigma\rceil(\Diamond^{e(B)}\boldsymbol{a}, \Diamond^{e(B)}\boldsymbol{x}, B^\bullet) \text{ as } e(A) = e(B) + n
\end{aligned}$$

where we always choose x_i for the $\Box$ case of $\lceil\sigma\rceil$ in Def. 5.7 without loss of generality.

This gives us, in the inductive case when $k > 1$ that:

$$\begin{aligned}
\lceil\sigma\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) &= 1 + \lceil\pi\rceil(\Diamond^{e(B)}\boldsymbol{a}, \Diamond^{e(B)}\boldsymbol{x}, B^\bullet) \text{ by Lemma 5.9}\\
&= 1 + (k-1) = k \text{ by (IH) applied to } B \in \Pi^q_{k-1}
\end{aligned}$$

and in the base case when $k = 1$ that:

$\lceil\sigma\rceil(\Diamond^{e(A)}\boldsymbol{a}, A^\bullet) = \lceil\sigma\rceil(\boldsymbol{a}, \boldsymbol{x}, B) = 1$ as B is quantifier-free ☐

Finally, we may conclude that the over-approximation we defined are in fact tight with respect to the $\cdot^\bullet$ encoding of QBFs.

Proposition 5.11 (Tightness) *Let A be a QBF.*

(i) *If $A \in \Pi^q_k$, for $k \geq 1$, then $\lceil \pi \rceil(A^\bullet) = \pi(A^\bullet) = k$.*

(ii) *If $A \in \Sigma^q_k$, for $k \geq 1$, then $\lceil \sigma \rceil(A^\bullet) = \sigma(A^\bullet) = k$.*

6 From K to QBFs via proof search

We are finally ready to present our 'inverse' encoding to $\cdot^\bullet$. By Cor. 5.4, let us henceforth fix polynomial-size Σ^q_k-formulas Σ^s_k-Prov_n, and Π^s_k-formulas Π^s_k-Prov_n, computing Σ^s_k-provability, and Π^s_k-provability respectively, on formulas of size $\leq n$.

Definition 6.1 Let A be a modal formula of size n. We define:

$$A^\circ := \begin{cases} \Sigma^s_k\text{-}\mathsf{Prov}_n(A) & k = \lceil \sigma \rceil(A) \leq \lceil \pi \rceil(A) \\ \Pi^s_k\text{-}\mathsf{Prov}_n(A) & k = \lceil \pi \rceil(A) \leq \lceil \sigma \rceil(A) \end{cases}$$

The main result of this work is:

Theorem 6.2 *We have the following:*

(i) $\cdot^\bullet$ *is a polynomial-time encoding from* CPC2 *to* K.

(ii) $\cdot^\circ$ *is a polynomial-time encoding from* K *to* CPC2.

(iii) *The composition $\cdot^{\bullet\circ}$ preserves quantifier complexity, i.e. for $k \geq 1$, if $A \in \Sigma^q_k$ (or $A \in \Pi^q_k$) then also $A^{\bullet\circ} \in \Sigma^q_k$ (or, respectively, $A^{\bullet\circ} \in \Pi^q_k$).*

Proof sketch. (i) already follows from Thm. 4.3. (ii) follows from the fact that $\lceil \sigma \rceil$ and $\lceil \pi \rceil$ are polynomial-time computable, and by Prop. 5.11 and Prop. 5.6. Finally (iii) follows by Prop. 5.6 and the definition of the QBFs Σ^s_k-Prov_n and Π^s_k-Prov_n, under Thm. 5.3 Cor. 5.4. □

In particular, we now inherit the following delineation of K:

Corollary 6.3 *We have the following, for $k \geq 1$:*

(i) $\{A$ *modal s.t.* $\lceil \sigma \rceil(A) \leq k : \vDash A\}$ *is Σ^p_k-complete.*

(ii) $\{A$ *modal s.t.* $\lceil \pi \rceil(A) \leq k : \vDash A\}$ *is Π^p_k-complete.*

Let us emphasise here that checking $\lceil \sigma \rceil(A) \leq k$ or $\lceil \pi \rceil(A) \leq k$ is polynomial-time, and so the above corollary indeed induces a *feasible* delineation of modal formulas into classes where validity is complete for the corresponding level of the polynomial-hierarchy.

7 Conclusions

In this work we classified fragments of modal logic K complete for each level of the polynomial hierarchy. In particular, we defined polynomial-time encodings from CPC2 to K and from K to CPC2 whose composition preserves the quantifier complexity of QBFs. Our translation from K to CPC2 employs a decomposition of the proof search space for K's sequent calculus $\mathsf{G3k}$ according to invertible and non-invertible rules in order to control alternation in its proof search predicate.

The fact that G3k admits terminating proof search, bottom-up, is crucial to our argument. We suspect that similar results can be found for extensions of K by axioms t or d, and in a multi-agent setting, all of which also give rise to terminating sequent calculi.

On the other hand, axioms such as 4 do not easily admit such terminating calculi (though there has been some recent progress in this regard [9]). Arguments for their **PSPACE**-membership via proof search rely on a form of loop checking, due to the necessary presence of *contraction* in the system. It would be interesting to try employ alternative formalisms, such as labelled or nested sequents equipped with a form of focussing (cf. [18,5]), to find similar treatments of such logics, building on previous works for fragments of intuitionistic logic [6] and linear logic [7,8].

References

[1] Andreoli, J., *Logic programming with focusing proofs in linear logic*, J. Log. Comput. **2** (1992), pp. 297–347.
URL `http://dx.doi.org/10.1093/logcom/2.3.297`

[2] Arora, S. and B. Barak, "Computational Complexity - A Modern Approach," Cambridge University Press, 2009.
URL `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`

[3] Blackburn, P., J. F. A. K. van Benthem and F. Wolter, editors, "Handbook of Modal Logic," Studies in logic and practical reasoning **3**, North-Holland, 2007.
URL `https://www.sciencedirect.com/bookseries/studies-in-logic-and-practical-reasoning/vol/3/suppl/C`

[4] Chandra, A. K., D. C. Kozen and L. J. Stockmeyer, *Alternation*, J. ACM **28** (1981), pp. 114–133.
URL `http://doi.acm.org/10.1145/322234.322243`

[5] Chaudhuri, K., S. Marin and L. Straßburger, *Focused and synthetic nested sequents*, in: B. Jacobs and C. Löding, editors, *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, Lecture Notes in Computer Science **9634** (2016), pp. 390–407.
URL `https://doi.org/10.1007/978-3-662-49630-5_23`

[6] Das, A., *Alternating time bounds from variants of focussed proof systems* (2017), preprint. `http://anupamdas.com/alt-time-bnds-var-foc-sys.pdf`.

[7] Das, A., *Focussing, MALL and the polynomial hierarchy*, in: D. Galmiche, S. Schulz and R. Sebastiani, editors, *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, Lecture Notes in Computer Science **10900** (2018), pp. 689–705.
URL `https://doi.org/10.1007/978-3-319-94205-6_45`

[8] Das, A., *From qbfs to MALL and back via focussing*, J. Autom. Reason. **64** (2020), pp. 1221–1245.
URL `https://doi.org/10.1007/s10817-020-09564-x`

[9] Fiorentini, C., *Terminating sequent calculi for proving and refuting formulas in S4*, J. Log. Comput. **25** (2015), pp. 179–205.
URL `https://doi.org/10.1093/logcom/exs053`

[10] Halpern, J. Y., *The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic*, Artif. Intell. **75** (1995), pp. 361–372.
URL `https://doi.org/10.1016/0004-3702(95)00018-A`

[11] Halpern, J. Y. and Y. Moses, *A guide to completeness and complexity for modal logics of knowledge and belief*, Artif. Intell. **54** (1992), pp. 319–379.
URL `https://doi.org/10.1016/0004-3702(92)90049-4`

[12] Halpern, J. Y. and L. C. Rêgo, *Characterizing the NP-PSPACE gap in the satisfiability problem for modal logic*, J. Log. Comput. **17** (2007), pp. 795–806.
URL `https://doi.org/10.1093/logcom/exm029`

[13] Ladner, R. E., *The computational complexity of provability in systems of modal propositional logic.*, SIAM J. Comput. **6** (1977), pp. 467–480.
URL `http://dblp.uni-trier.de/db/journals/siamcomp/siamcomp6.html#Ladner77`

[14] Lagniez, J., D. L. Berre, T. de Lima and V. Montmirail, *On checking kripke models for modal logic K*, in: P. Fontaine, S. Schulz and J. Urban, editors, *Proceedings of the 5th Workshop on Practical Aspects of Automated Reasoning co-located with International Joint Conference on Automated Reasoning (IJCAR 2016), Coimbra, Portugal, July 2nd, 2016*, CEUR Workshop Proceedings **1635** (2016), pp. 69–81.
URL `http://ceur-ws.org/Vol-1635/paper-07.pdf`

[15] Liang, C. and D. Miller, *Focusing and polarization in linear, intuitionistic, and classical logics*, Theoretical Computer Science **410** (2009), pp. 4747–4768.

[16] Massacci, F., *Design and results of the tableaux-99 non-classical (modal) systems comparison*, in: N. V. Murray, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '99, Saratoga Springs, NY, USA, June 7-11, 1999, Proceedings*, Lecture Notes in Computer Science **1617** (1999), pp. 14–18.
URL `https://doi.org/10.1007/3-540-48754-9_2`

[17] Massacci, F. and F. M. Donini, *Design and results of TANCS-2000 non-classical (modal) systems comparison*, in: R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2000, St Andrews, Scotland, UK, July 3-7, 2000, Proceedings*, Lecture Notes in Computer Science **1847** (2000), pp. 52–56.
URL `https://doi.org/10.1007/10722086_4`

[18] Miller, D. and M. Volpe, *Focused labeled proof systems for modal logic*, in: M. Davis, A. Fehnker, A. McIver and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, Lecture Notes in Computer Science **9450** (2015), pp. 266–280.
URL `https://doi.org/10.1007/978-3-662-48899-7_19`

[19] Nguyen, L. A., *On the complexity of fragments of modal logics*, in: R. A. Schmidt, I. Pratt-Hartmann, M. Reynolds and H. Wansing, editors, *Advances in Modal Logic 5, papers from the fifth conference on "Advances in Modal logic," held in Manchester, UK, 9-11 September 2004* (2004), pp. 249–268.
URL `http://www.aiml.net/volumes/volume5/Nguyen.ps`

[20] Pan, G. and M. Y. Vardi, *Optimizing a bdd-based modal solver*, in: F. Baader, editor, *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction Miami Beach, FL, USA, July 28 - August 2, 2003, Proceedings*, Lecture Notes in Computer Science **2741** (2003), pp. 75–89.
URL `https://doi.org/10.1007/978-3-540-45085-6_7`

[21] Schmidt-Schauß, M. and G. Smolka, *Attributive concept descriptions with complements*, Artif. Intell. **48** (1991), pp. 1–26.
URL `https://doi.org/10.1016/0004-3702(91)90078-X`

[22] Sebastiani, R. and M. Vescovi, *Automated reasoning in modal and description logics via SAT encoding: the case study of k(m)/alc-satisfiability*, CoRR **abs/1401.3463** (2014).
URL `http://arxiv.org/abs/1401.3463`

[23] Statman, R., *Intuitionistic propositional logic is polynomial-space complete*, Theor. Comput. Sci. **9** (1979), pp. 67–72.
URL `http://dx.doi.org/10.1016/0304-3975(79)90006-9`

[24] Stockmeyer, L. J., *The polynomial-time hierarchy*, Theor. Comput. Sci. **3** (1976), pp. 1–22.
URL `https://doi.org/10.1016/0304-3975(76)90061-X`

[25] Troelstra, A. S. and H. Schwichtenberg, "Basic proof theory, Second Edition," Cambridge tracts in theoretical computer science **43**, Cambridge University Press, 2000.

[26] Zolin, E., *Description logic complexity navigator*, webpage, accessed June 14th 2022. URL `https://www.cs.man.ac.uk/~ezolin/dl/`