# Resolution-Based Uniform Interpolation for Multi-Agent Modal Logic $K_n$

Ruba Alassaf, Renate A. Schmidt, and Uli Sattler

University of Manchester, UK
{ruba.alassaf,renate.schmidt,uli.sattler}@manchester.ac.uk

**Abstract.** Research on uniform interpolation in modal logic has been largely focused on the theoretical investigation of the problem. This paper presents a system to compute uniform interpolants for a locally satisfiable formula in the multi-agent modal logic $K_n$. The system is based on a direct resolution approach. The idea of the system is that given a formula $\phi$ and a signature as input, it computes the strongest local consequence of $\phi$ over the input signature. We have shown that the system is guaranteed to terminate, soundness and completeness can be shown using model-theoretic proofs, and the worst-case space complexity bound is double exponential. We illustrate how the system is used via examples.

**Keywords:** Uniform Interpolation · Resolution · Modal logic · $K_n$.



## 1 Introduction

Uniform interpolation is the task of computing a formula that captures all logical consequences up to a given signature for a given formula. The problem of computing a uniform interpolant is generally not decidable. We are interested in logics that have the *uniform interpolation* property: that is, the property that for any formula $\phi$ and any signature $\Sigma$, a uniform interpolant for $\phi$ over $\Sigma$ exists in the logic.

Uniform interpolation amounts to the second-order quantifier elimination problem: given a formula with an existential second-order quantifier prefix, the process of eliminating the second-order quantifiers is essentially the process of producing a formula that does not contain the symbols that are being quantified.

A related notion is Craig interpolation. A logic has the Craig interpolation property if given two formulas $\phi$ and $\psi$ with a shared signature $\Sigma$ such that $\phi \to \psi$ holds, there exists a middle formula $\phi'$ such that $\phi \to \phi'$ and $\phi' \to \psi$ hold. A uniform interpolation method can be used to compute Craig interpolants by performing uniform interpolation to keep the shared signature $\Sigma$.

The importance of having uniform interpolation methods stems from the significant potential it has for applications. For example, in agent-based applications, it is often assumed that agents communicate using the same language.

Uniform interpolation becomes very useful when this assumption is relaxed; it can be used to allow an agent to express knowledge about a certain topic by computing a view that only uses some signature symbols. This gives agents the ability to share their knowledge with other agents who specialise in different domains.

The modal logic community has focused on uncovering theoretical results. It has been shown via constructive proofs that the modal logic $K$ has the uniform interpolation property [7, 14]. An approach to constructing uniform interpolants was given in [2] for the modal logics $K$ and $T$. Wolter [15] proved that the modal logic $S5$ has the uniform interpolation property, and that uniform interpolation for any normal single-agent modal logic can be generalised to its multi-agent case. Recently, it was shown that $K45_n$ and $KD45_n$ have the uniform interpolation property in [4]. It is known that $S4$ and $K4$ do not have the uniform interpolation property [8].

This paper presents the first complete resolution-based system for computing uniform interpolants in the multi-agent modal logic $K_n$. As far as the authors know, the only other paper which considers this logic is [4]. Different from our method, they construct a uniform interpolant by considering canonical formulas, which are conceptually simple but, as the authors explicitly state, inefficient to compute [4]. We show that our system has double exponential worst-case space complexity. We prove that the termination of our method is guaranteed, and that it is sound and complete. We are the first to use bisimulations to prove completeness for a resolution-based uniform interpolation system. We illustrate how the method is used via examples. Due to the lack of space, proof are provided in the full version of the paper which can be found here: `https://personalpages.manchester.ac.uk/staff/ruba.alassaf/publications.html`

## 2 Preliminaries

We assume the reader is familiar with the multi-modal logic $K_n$ [5, 10]. We use $\mathcal{F} = (\mathcal{W}, R)$ to denote a Kripke frame and $\mathcal{M} = (\mathcal{W}, R, V)$ to denote a Kripke model. A formula $\phi$ is (locally) *satisfiable in a model* $\mathcal{M}$, denoted $\mathcal{M}, w \models \phi$, if there is a point $w$ in $\mathcal{W}$ at which $\phi$ is true. A formula $\phi$ is (unconditionally) *satisfiable* if it is true at some point in some model. A formula $\phi$ is globally satisfied (or true) in a model $\mathcal{M}$, denoted $\mathcal{M} \models \phi$, if it is true at every $w$ in $\mathcal{M}$. A formula $\phi$ is valid if it is satisfied in all models over any frame $\mathcal{F}$. A set of formulae $N$ is globally satisfied by a model $\mathcal{M}$, denoted $\mathcal{M} \models N$, if for each formula $\phi$ in $N$, $\mathcal{M}$ globally satisfies $\phi$.

We are interested in the problem of computing a uniform interpolant of a locally satisfiable formula and a signature.

**Definition 1 (Uniform Interpolation).** *Given a formula $\phi$, a* uniform interpolant *of $\phi$ with respect to a signature $\Sigma$ of propositional symbols is a formula $\phi'$ such that:*

1. *$\phi'$ does not contain symbols outside of $\Sigma$, and*

*2. for any modal formula $\psi$ over $\Sigma$, we have that for all models $\mathcal{M}$, $\mathcal{M} \models \phi \to \psi$ iff for all models $\mathcal{M}$, $\mathcal{M} \models \phi' \to \psi$.*

## 3 Related Work

In this section, we outline the methods we found related to our method, and explain how our method is different to these systems. A summary of the related methods is given in Table 1. In the table, we give the logic over which each method is defined, the expressivity of the input and output, and we state if the method is complete.

The first method is a uniform interpolation algorithm of Bilkova [2]. In her work, she describes an approach for constructing a uniform interpolant from a table. She uses a sequent calculus to prove that her algorithm is sound and complete.

The second is a resolution-based calculus introduced in Herzig and Mengin [9]. There are two differences to our method, the first is that the method proposed by the present paper is for $K_n$ which is an extension of $K$, and the second is that we use a kind of labelling technique that allows us to flatten the input and apply resolution almost classically.

There are three more resolution-based systems for computing uniform interpolation: the SCAN approach [6] for first-order logic, and the LETHE system [11] and the system of Ludwig and Konev [12], both for description logics. These systems are designed for logics where a solution does not always exist. In the case of SCAN, the computation may not terminate [6]. In the case of LETHE, nominals/definer symbols may remain in the solution [11], or solutions may be approximated by a depth bound as in the method in [12]. We prove that a solution is always achievable via our method in a finite number of steps and without extending the logic or the signature. The completeness proofs provided for these methods are based on consequence finding, whereas our proof uses bisimulations. Moreover, compared to [11], the method we describe does not use unification-based reasoning.

Finally, second-order quantifier elimination methods which can be used to compute uniform interpolants often use Ackermann's lemma [1]. Such methods include the DLS algorithm [3] for second-order quantifier elimination of first-order logic formulae, the MA system [13] for computing frame correspondence properties for modal axioms and the FAME tool [16] computing semantic forgetting in description logic.

## 4 Uniform Interpolation Method $UI_{K_n}$ for $K_n$

We start with a high-level description of our uniform interpolation system for multi-modal logic $K_n$.

| | $UI_{K_n}$ | Bilkova [2] | Herzig & Mengin [9] | SCAN [6] | LETHE [11] | Ludwig & Konev [12] | DLS algorithm [3] |
|---|---|---|---|---|---|---|---|
| Logic | Modal logic | Modal logic | Modal logic | First-order logic | Description logic | Description logic | First-order logic |
| Method | Resolution | Sequent | Resolution | Resolution | Resolution | Resolution | Ackermann |
| Input Language | $K_n$ (locally satisfiable formula) | $K$ (locally satisfiable formula) | $K$ (locally satisfiable formula) | Full first-order logic | $\mathcal{ALC}$ ($\mathcal{T}box$ $+\mathcal{A}box$) | $\mathcal{ALC}$ ($\mathcal{T}box$) | Full first-order logic |
| Output Language | $K_n$ (locally satisfiable formula) | $K$ (locally satisfiable formula) | $K$ (locally satisfiable formula) | Full first-order logic | $\mathcal{ALCO}\mu$ ($\mathcal{T}box$ $+\mathcal{A}box$) | $\mathcal{ALC}$ ($\mathcal{T}box$) | Full first-order logic |
| Complete | Yes | Yes | Yes | No | Yes | No | No |

Table 1: A comparison between our method $UI_{K_n}$ for $K_n$ and related methods.

### 4.1 Overview

The calculus is based on resolution, with adaptations for modal logic. The idea behind our approach is the following: for each symbol $x$ outside the given signature $\Sigma$, we generate a sufficient set of conclusions for the given formula and subsequently eliminate any formulae that contain $x$. We repeat the process for all propositional symbols outside $\Sigma$.

The calculus uses special *world symbols*, or *W-symbols* for short, which are propositional symbols that help in two related ways:

1. They are used to flatten the input formula to surface some parts of it. E.g., $\Box(\psi \vee \Diamond\phi)$ becomes $\Box W_1$, $W_1 \Rightarrow \psi \vee \Diamond W_2$ and $W_2 \Rightarrow \phi$.
2. They allow our rules to detect legal inferences between the subformulae by labelling them with a $W$-symbol. E.g., $\Box(x \wedge (\neg x \vee p))$ becomes $\Box W$, $W \Rightarrow x$ and $W \Rightarrow \neg x \vee p$. Later on, we see that one of our rules allows us to apply a resolution step on $x$.

The idea behind using $W$-symbols is similar to using constants in a labelled tableau algorithm.

For a formula $\phi$, a signature $\Sigma$, and an ordering $\succ$ over the symbols outside the input signature $\Sigma$, the calculus is provided a clause set $N_0 = \{W_0 \Rightarrow \phi\}$ as input, and applies its rules exhaustively to the formulae in the set until no rule can be applied, resulting in a clause set of the form $N_n = \{W_0 \Rightarrow \phi_1, ..., W_0 \Rightarrow \phi_m\}$. The formula $\phi' = \phi_1 \wedge \ldots \wedge \phi_m$ is then a uniform $\Sigma$-interpolant of $\phi$, which is proved later.

The role of $W_0$ is to capture a specific world that satisfies $\phi$. Any model $\mathcal{M}$ that satisfies $\phi$ at point $w$ can be extended to one that satisfies $W_0$ and $W_0 \Rightarrow \phi$ in a non-vacuous way by setting $w \in V(W_0)$. In this extended model, $W_0 \Rightarrow \phi$ is globally and witnessed as non-vacuously true.

The process of constructing a uniform interpolant is iterative with respect to the symbols outside $\Sigma$, and the ordering $\succ$ fixes the order in which these symbols are eliminated. For some uniform interpolation problems, a good ordering may allow the calculus to solve a problem in far fewer steps. For simplicity, and since the ordering does not improve any worst-case complexity results, we can assume that this ordering is arbitrary. We use $x$ to denote the maximal propositional symbol occurring in the current clause set $N_i$.

### 4.2 The Calculus

The rules of our uniform interpolant calculus are given in Figures 1, 2 and 3. Each rule has a premise, some conditions and a conclusion. The rules are structured with the premise above a horizontal line and the conclusion below it. The premise (respectively conclusion) can be one or more clauses depending on which rule is being applied. There are three types of rules in the calculus: preprocessing rules, resolution rules, and elimination rules.

The preprocessing rules and the elimination rules are replacement rules; they replace the premise in the current working clause set with the conclusion. The resolution rules are saturation rules; they keep the premise and extend the clause set with the conclusion. The rules can be applied in any order as long as the conditions for each rule are met.

Generally, we can expect that for a formula in the clause set, it is preprocessed into another formula, or formulae, that is then involved in a few resolution rule applications and subsequently purified, if an elimination rule is applicable.

The clauses obtained and handled by our calculus are in a normal form. They are all labelled with a $W$-symbol in the condition of the implication. We can have a formula or another $W$-symbol in the consequence of the implication. Concretely, for some $W$-symbols $W_i$ and $W_j$, and some modal formula $\psi$, a clause can be in the form

$$W_i \Rightarrow \psi \quad \text{or} \quad W_i \Rightarrow W_j.$$

If $\psi$ is a disjunction of modal formulas, we assume that it is a set, i.e., there is no repetition. This is essential for the correctness of the method. We use $\Rightarrow$, in contrast to $\rightarrow$, to distinguish an implication that is generated by our system, to maintain our normal form, from an implication provided as part of the input. Semantically, they are identical.

To describe the different types of $W$-symbols, we introduce some terminology and the function $Corr$ which will be used in the conditions of our system, and later on in the proofs.

**Definition 2.** *Given a set $N$ of clauses, the set $S_w$ is the set of $W$-symbols introduced for subformulas appearing under a modal operator via the world introduction rule. We call these symbols* base $W$-symbols*.*

*The set $C_w$ is the set of $W$-symbols introduced by the $\Box\bigcirc$ rule. We call these symbols* combinatory $W$-symbols*.*

*We define a function $Corr$ that maps $W$-symbols to subsets of $S_w$ as follows:*

$$Corr(W_i) = \begin{cases} \{W_i\}, & \text{if } W_i \in S_w \\ Corr(W_n) \cup Corr(W_m), & \text{if } W_i \in C_w \text{ where } W_n \text{ and } W_m \text{ come} \\ & \text{from the premise of the } \textsc{Res}\ \Box\bigcirc \\ & \text{rule that has introduced } W_i. \end{cases}$$

Intuitively, a base $W$-symbol is introduced to represent a subformula, and a combinatory $W$-symbol can be seen as a unique representative of a subset of the base $W$-symbols.

We now describe the three groups of rules which together make up our calculus. We use $N$ to refer to the current working clause set. We assume that $x$ is the current symbol we would like to eliminate, i.e., it is the maximal symbol with respect to a given ordering $\succ$ for symbols outside $\Sigma$. The $W$-symbol $W_i$ is the $i$th $W$-symbol introduced during the inference process.

**Preprocessing.** The purpose of the preprocessing rules is to apply transformations to the members of the working clause set so that they can be handled by the other rules. Generally, the idea is to surface symbols appearing in $\phi$ that are not in $\Sigma$, i.e., to surface $x$ in $\phi$.

The normal form is based on pushing negation inwards, clausifying and applying structural transformation. The rules are applied in a lazy manner which means their application can be deferred to whenever they are necessary. The preprocessing rules are provided in Figure 1.

The first five rules are standard rules to transform modal formulae into negation normal form. The clausification rule distributes disjunction over conjunction. The world introduction rule performs structural transformation that flattens the modal formulae. Consider a clause $W_i \Rightarrow \neg\neg\psi$, the first negation normal form rule replaces this clause with $W_i \Rightarrow \psi$, so the original clause is no longer in the working set.

**Resolution.** The second type of rules are the resolution rules. The purpose of these rules is to deduce a sufficient number of clauses/formulas to generate a uniform interpolant. The rules are given in Figure 2.

The literal resolution rule is the heart of our calculus; it computes a formula by resolving on a maximal symbol $x$ if the premise is labelled with the same $W$-symbol. The world resolution rule is used to propagate formulas labelled by another $W$-symbol, which is essentially a resolution step between world symbols. The $\Box\bigcirc$ resolution rule is used to capture combinations of successor relations. The second and third conditions are the blocking conditions; they aim to ensure that the rule application is not redundant which is important for complexity, and that the calculus does not infinitely introduce $W$-symbols which is essential for termination.

**Elimination.** The last type of rules are the elimination rules. These rules are responsible for eliminating symbols outside of $\Sigma \cup \{W_0\}$. They are applied once we have exhaustively applied the resolution rules to compute conclusions over $\Sigma$. The rules are given in Figure 3.

The positive and negative purification rules replace a maximal symbol $x$, occurring either positively or negatively, with $\top$. The world elimination rule collects modal formulas labelled with the same $W$-symbol, and replaces right

**Negation Normal Form (1)**:

$$\frac{N, W_i \Rightarrow \neg\neg\phi_1 \vee \phi_2}{N, W_i \Rightarrow \phi_1 \vee \phi_2}$$

provided that $\phi_1$ contains $x$. $\phi_2$ may be empty.

**Negation Normal Form (2)**:

$$\frac{N, W_i \Rightarrow \neg(\phi_1 \wedge \phi_2) \vee \phi_3}{N, W_i \Rightarrow \neg\phi_1 \vee \neg\phi_2 \vee \phi_3}$$

provided that either $\phi_1$ or $\phi_2$ contain $x$. $\phi_3$ may be empty.

**Negation Normal Form (3)**:

$$\frac{N, W_i \Rightarrow \neg(\phi_1 \vee \phi_2) \vee \phi_3}{N, W_i \Rightarrow \neg\phi_1 \vee \phi_3, W_i \Rightarrow \neg\phi_2 \vee \phi_3}$$

provided that either $\phi_1$ or $\phi_2$ contain $x$. $\phi_3$ may be empty.

**Negation Normal Form (4)**:

$$\frac{N, W_i \Rightarrow \neg\Diamond_a\phi_1 \vee \phi_2}{N, W_i \Rightarrow \Box_a\neg\phi_1 \vee \phi_2}$$

provided that $\phi_1$ contains $x$. $\phi_2$ may be empty.

**Negation Normal Form (5)**:

$$\frac{N, W_i \Rightarrow \neg\Box_a\phi_1 \vee \phi_2}{N, W_i \Rightarrow \Diamond_a\neg\phi_1 \vee \phi_2}$$

provided that $\phi_1$ contains $x$. $\phi_2$ may be empty.

**Implication Elimination**:

$$\frac{N, W_i \Rightarrow (\phi_1 \rightarrow \phi_2) \vee \phi_3}{N, W_i \Rightarrow \neg\phi_1 \vee \phi_2 \vee \phi_3}$$

provided that either $\phi_1$ or $\phi_2$ contain $x$. $\phi_3$ may be empty.

**Clausification**:

$$\frac{N, W_i \Rightarrow (\phi_1 \wedge \phi_2) \vee \phi_3}{N, W_i \Rightarrow \phi_1 \vee \phi_3, W_i \Rightarrow \phi_2 \vee \phi_3}$$

provided that either $\phi_1$ or $\phi_2$ contain $x$. $\phi_3$ may be empty.

**World Introduction** (Int W):

$$\frac{N, W_i \Rightarrow \bigcirc_a\phi_1 \vee \phi_2}{N, W_i \Rightarrow \bigcirc_a W_j \vee \phi_2, W_j \Rightarrow \phi_1}$$

provided that

(i) $\bigcirc \in \{\Box, \Diamond\}$,
(ii) $\phi_1$ must contain $x$,
(iii) if $\phi_2$ contains $x$ then $x$ must occur under a modal operator, and
(iv) $W_j$ is a fresh $W$-symbol, and $Corr(W_j) = \{W_j\}$.

$\phi_2$ may be empty.

Fig. 1: The preprocessing rules for $UI_{K_n}$ calculus for the modal logic $K_n$. The rules are replacement rules: each rule replaces the premise with an equisatisfiable formula. In each rule, $x$ is assumed to be the maximal symbol specified by the given ordering $\succ$ on the symbols outside $\Sigma$ occurring in the premises.

**Literal Resolution** (RES):

$$\frac{W_i \Rightarrow \psi_1 \vee x \quad W_i \Rightarrow \psi_2 \vee \neg x}{W_i \Rightarrow \psi_1 \vee \psi_2}$$

$\psi_1$ and/or $\psi_2$ may be empty.

**World Resolution** (RES W):

$$\frac{W_i \Rightarrow \psi \quad W_j \Rightarrow W_i}{W_j \Rightarrow \psi}$$

provided that $\psi$ contains $x$. $\psi$ may be a $W$-symbol.

**$\Box\bigcirc$ Resolution** (RES $\Box\bigcirc$):

$$\frac{W_i \Rightarrow \psi_1 \vee \Box_a W_n \quad W_i \Rightarrow \psi_2 \vee \bigcirc_a W_m}{W_i \Rightarrow \psi_1 \vee \psi_2 \vee \bigcirc_a W_j,\ W_j \Rightarrow W_n,\ W_j \Rightarrow W_m}$$

provided that:
(i) $\bigcirc \in \{\Box, \Diamond\}$,
(ii) $Corr(W_n) \cap Corr(W_m)$ is empty,
(iii) if there is a $W_k$ such that $Corr(W_k) = Corr(W_n) \cup Corr(W_m)$ then $W_j = W_k$, otherwise $W_j$ is a fresh $W$-symbol, and $Corr(W_j) = Corr(W_n) \cup Corr(W_m)$.

$\psi_1$ and/or $\psi_2$ may be empty.

Fig. 2: The resolution rules of the $UI_{K_n}$ calculus for the modal logic $K_n$. The rules given here are saturation rules: they add conclusions to the current working clause set. We use $x$ to refer to the maximal symbol in the working set specified by a given ordering $\succ$ on the symbols outside $\Sigma$.

**Positive Purification** (+PUR):

$$\frac{N, W_i \Rightarrow \psi \vee x}{N, W_i \Rightarrow \psi \vee \top}$$

provided that no more non-purification inference rules can be applied. $\psi$ may be empty.

**Negative Purification** (-PUR):

$$\frac{N, W_i \Rightarrow \psi \vee \neg x}{N, W_i \Rightarrow \psi \vee \top}$$

provided that no more non-purification inference rules can be applied. $\psi$ may be empty.

**World Elimination** (ELM W):

$$\frac{N, W_i \Rightarrow \psi_1, \ldots, W_i \Rightarrow \psi_n}{N^{W_i}_{(\psi_1 \wedge \cdots \wedge \psi_n)}}$$

provided that $i \neq 0$, $\psi_1, \ldots, \psi_n$ do not contain $x$ or any $W$-symbol, and $N$ only contains $W_i$ on the right hand side of $\Rightarrow$ clauses. The expression $N^{\phi}_{\psi}$ denotes the set of clauses that is obtained by replacing each occurrence of $\phi$ in $N$ by $\psi$.

Fig. 3: The purification and elimination rules of the $UI_{K_n}$ calculus for modal logic $K_n$. The rules are replacement rules: each rule replaces the premise with an equisatisfiable formula. In each rule, $x$ is assumed to be the maximal symbol specified by the given ordering $\succ$ on the symbols outside $\Sigma$ occurring in the premises.

hand side occurrences of the $W$-symbol with the conjunction of these formulas, effectively eliminating the $W$-symbol from the set of clauses.

### 4.3 Examples

In the following examples, we demonstrate how the $UI_{K_n}$ system is used to compute a uniform interpolant with respect to $\Sigma = \{p, q\}$. Starting from $i = 0$, we use $N_i$ to refer to the clause set that is obtained after applying the $i$th step in the derivation.

*Example 1.* Consider a formula $\phi = (\neg p \vee \Diamond x) \wedge (\neg x \vee \Box q)$.

The input to the system is the set $N_0 = \{W_0 \Rightarrow (\neg p \vee \Diamond x) \wedge (\neg x \vee \Box q)\}$. The only rule applicable to $N_0$ is the clausification rule which gives

$$N_1 = \{W_0 \Rightarrow \neg p \vee \Diamond x, W_0 \Rightarrow \neg x \vee \Box q\}.$$

Now we apply the world introduction rule to get

$$N_2 = \{W_0 \Rightarrow \neg p \vee \Diamond W_1, W_1 \Rightarrow x, W_0 \Rightarrow \neg x \vee \Box q\}.$$

The only applicable rules are the positive and negative purification rules. We achieve

$$N_3 = \{W_0 \Rightarrow \neg p \vee \Diamond W_1, W_1 \Rightarrow \top, W_0 \Rightarrow \top \vee \Box q\}.$$

Eliminating $W_1$, we obtain

$$N_4 = \{W_0 \Rightarrow \neg p \vee \Diamond \top, W_0 \Rightarrow \top \vee \Box q\}.$$

The $\Sigma$-uniform interpolant is $\phi' = (\neg p \vee \Diamond \top) \wedge (\top \vee \Box q)$.

Notice that this example illustrates the local flavour of the system. We see that the occurrences of $x$ at two different modal levels do not interact via any resolution rule.

*Example 2.* Consider a formula $\phi = (\neg p \vee \Diamond x) \wedge \Box(\neg x \vee \Box q)$. We start with the set $N_0 = \{W_0 \Rightarrow (\neg p \vee \Diamond x) \wedge \Box(\neg x \vee \Box q)\}$. Applying clausification to $N_0$ we get

$$N_1 = \{W_0 \Rightarrow \neg p \vee \Diamond x, W_0 \Rightarrow \Box(\neg x \vee \Box q)\}.$$

By applying the world introduction rule twice, we have

$$N_3 = \{W_0 \Rightarrow \neg p \vee \Diamond W_1, W_1 \Rightarrow x, W_0 \Rightarrow \Box W_2, W_2 \Rightarrow \neg x \vee \Box q\}.$$

The only applicable rule is the $\Box\Diamond$ rule, and it yields

$$N_4 = N_3 \cup \{W_0 \Rightarrow \neg p \vee \Diamond W_3, W_3 \Rightarrow W_1, W_3 \Rightarrow W_2\}.$$

By applying the world resolution rule twice, we obtain

$$N_6 = N_4 \cup \{W_3 \Rightarrow x, W_3 \Rightarrow \neg x \vee \Box q\}.$$

Now, we can apply the literal resolution rule which yields

$$N_7 = N_6 \cup \{W_3 \Rightarrow \Box q\}.$$

We apply the positive and negative purification rules (4 applications) and achieve

$$\begin{aligned} N_{11} = \{\, & W_0 \Rightarrow \neg p \vee \Diamond W_1, && W_1 \Rightarrow \top, && W_0 \Rightarrow \Box W_2, \\ & W_2 \Rightarrow \top \vee \Box q, && W_0 \Rightarrow \neg p \vee \Diamond W_3, && W_3 \Rightarrow W_1, \\ & W_3 \Rightarrow W_2, && W_3 \Rightarrow \top, && W_3 \Rightarrow \top \vee \Box q, \\ & W_3 \Rightarrow \Box q \,\}. \end{aligned}$$

Now, $x$ does not appear anywhere. We eliminate the world variables $W_1$, $W_2$, $W_3$ via the world elimination rule.

To eliminate $W_1$, we look for clauses labelled with $W_1$, in this case we only have $W_1 \Rightarrow \top$. We remove $W_1 \Rightarrow \top$ and replace each occurrence of $W_1$ on the right hand side of $\Rightarrow$ with $\top$ as follows:

$$\begin{aligned} N_{12} = \{\, & W_0 \Rightarrow \neg p \vee \Diamond \top, && W_0 \Rightarrow \Box W_2, && W_2 \Rightarrow \top \vee \Box q, \\ & W_0 \Rightarrow \neg p \vee \Diamond W_3, && W_3 \Rightarrow \top, && W_3 \Rightarrow W_2, \\ & W_3 \Rightarrow \top \vee \Box q, && W_3 \Rightarrow \Box q \,\}. \end{aligned}$$

Similarly for $W_2$, we remove $W_2 \Rightarrow \top \vee \Box q$, and replace the other occurrences of $W_2$ with $\top \vee \Box q$.

$$\begin{aligned} N_{13} = \{\, & W_0 \Rightarrow \neg p \vee \Diamond \top, && W_0 \Rightarrow \Box(\top \vee \Box q), && W_0 \Rightarrow \neg p \vee \Diamond W_3, \\ & W_3 \Rightarrow \top, && W_3 \Rightarrow \top \vee \Box q, && W_3 \Rightarrow \Box q \,\}. \end{aligned}$$

Finally, we eliminate $W_3$,

$$\begin{aligned} N_{14} = \{\, & W_0 \Rightarrow \neg p \vee \Diamond \top, && W_0 \Rightarrow \Box(\top \vee \Box q), \\ & && W_0 \Rightarrow \neg p \vee \Diamond(\top \wedge (\top \vee \Box q) \wedge \Box q) \,\}. \end{aligned}$$

The uniform interpolant is

$$\phi' = (\neg p \vee \Diamond \top) \wedge (\Box(\top \vee \Box q)) \wedge (\neg p \vee \Diamond(\top \wedge (\top \vee \Box q) \wedge \Box q)),$$

which is equivalent to $\phi' = (\neg p \vee \Diamond \Box q)$ by standard simplifications.

## 4.4 Correctness

The output $\phi'$ is correct if it is a uniform interpolant of a formula $\phi$ and a signature $\Sigma$, produced in a finite number of steps. There are three issues at hand: termination, soundness and completeness. We state the theorems and lemmas that are relevant to these topics. For the proofs, we refer the reader to the full version of the paper[1].

First are lemmas which are relevant to termination. We prove termination by showing that any derivation uses a finite number of symbols, and we argue that because of this, the calculus will stop generating new clauses.

**Lemma 1.** *For a given formula $\phi$ and a signature $\Sigma$, the $UI_{K_n}$ calculus introduces a finite number of $W$-symbols.*

[1] https://personalpages.manchester.ac.uk/staff/ruba.alassaf/publications.html

**Lemma 2.** *For a given formula $\phi$ and a signature $\Sigma$, the $UI_{K_n}$ calculus will stop generating new clauses.*

**Lemma 3.** *For a given formula $\phi$ and a signature $\Sigma$, the $UI_{K_n}$ system will not reintroduce a $W$-symbol that was eliminated before.*

From Lemma 1, 2 and 3, we conclude the following theorem.

**Theorem 1 (Termination).** *Given a formula $\phi$ and a signature $\Sigma$, the uniform interpolation system $UI_{K_n}$ computes a formula $\phi'$ in a finite number of steps.*

The following lemma addresses the space complexity of our system.

**Lemma 4.** *The space complexity of the $UI_{K_n}$ calculus is double exponentially bounded in the length of the input.*

The idea of the proof is to show each clause is exponentially bounded in the length of the input $n$, and that the number of clauses produced by the system is double exponentially bounded by $n$.

The next lemmas argue that the signature of $\phi'$ is $\Sigma$.

**Lemma 5.** *The $UI_{K_n}$ system will always be able to eliminate every $W$-symbol that is not $W_0$, using the world elimination rule.*

**Lemma 6.** *The $UI_{K_n}$ system will always be able to eliminate symbols in the signature of $\phi$ that are not in $\Sigma$.*

Next, we state the soundness theorem.

**Theorem 2 (Soundness).** *Given a formula $\phi$ and a signature $\Sigma$, the uniform interpolation system $UI_{K_n}$ computes a formula $\phi'$ such that for any formula $\psi$ over $\Sigma$, we have that*

$$\text{if } \models \phi' \to \psi \text{ then } \models \phi \to \psi.$$

For our completeness proof, we are interested in understanding models that are invariant up to the satisfaction of $\Sigma$-modal formulas. $\Sigma$-modal formulas are modal formulas described using a signature of propositional symbols $\Sigma$. For this purpose, we use the following notion.

**Definition 3 ($\Sigma-$bisimulation).** *Let $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ be two Kripke models where $\mathcal{M} = (\mathcal{W}, R, V)$ and $\mathcal{M}' = (\mathcal{W}', R', V')$. A $\Sigma-$*bisimulation *between $\mathcal{M}$ and $\mathcal{M}'$ is a relation $\rho \subseteq \mathcal{W} \times \mathcal{W}'$ such that $w \rho w'$, and whenever $u \rho u'$, the following holds:*

***atoms*** *$u$ and $u'$ satisfy the same propositional symbols from $\Sigma$;*

***forth*** *For all $a$, if $uR_a t$, then there is a $t'$ such that $u'R'_a t'$ and $t \rho t'$;*

***back*** *For all $a$, if $u'R'_a t'$, then there is a $t$ such that $uR_a t$ and $t \rho t'$.*

The following is our completeness theorem.

**Theorem 3 (Completeness).** *Given a formula $\phi$ and a signature $\Sigma$, the uniform interpolation system $UI_{K_n}$ computes a formula $\phi'$ such that, for any formula $\psi$ over $\Sigma$, we have that*

$$\textit{if } \models \phi \to \psi \textit{ then } \models \phi' \to \psi.$$

Using proof by contradiction, we assume that $\models \phi \to \psi$ but $\not\models \phi' \to \psi$. The assumption implies that there exists a counter-model $\mathcal{M}'$ and a world $w_0$ such that, $\mathcal{M}', w_0 \models \phi'$ and $\mathcal{M}', w_0 \not\models \psi$. We use $\Sigma$-bisimulation to prove by induction that this is not possible.

## 5 Conclusion

The paper presented a resolution-based method to compute uniform interpolants for the multi-agent modal logic $K_n$. It has been shown that our method terminates, and is sound and complete. The space complexity was proven to be at most double exponential in the length of the input. This work is intended to be the basis of our future work. We would like to study logics which are known to have the uniform interpolation property, and show that the presented system can be extended to solve the uniform interpolation problem for more modal logics. An implementation is being developed to demonstrate practicality.

## References

1. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Mathematische Annalen 110, 390–413 (1935)
2. Bílková, M.: Uniform interpolation and propositional quantifiers in modal logics. Studia Logica: An International Journal for Symbolic Logic 85(1), 1–31 (2007)
3. Doherty, P., Łukaszewicz, W., Szałas, A.: Computing circumscription revisited: A reduction algorithm. Journal of Automated Reasoning 18(3), 297–336 (1997)
4. Fang, L., Liu, Y., Van Ditmarsch, H.: Forgetting in Multi-agent Modal Logics. In: Proc. IJCAI 2016. pp. 1066–1073. IJCAI/AAAI Press (2016)
5. Fitting, M.: Modal proof theory. In: Handbook of Modal Logic, vol. 3, pp. 85–138. Elsevier (2007)
6. Gabbay, D.M., Ohlbach, H.J.: Quantifier elimination in second-order predicate logic. In: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning. pp. 425–435. KR'92, Morgan Kaufmann (1992)
7. Ghilardi, S.: An algebraic theory of normal forms. Annals of Pure and Applied Logic 71(3), 189 – 245 (1995)
8. Ghilardi, S., Zawadowski, M.: Undefinability of propositional quantifiers in the modal system S4. Studia Logica 55(2), 259–271 (1995)
9. Herzig, A., Mengin, J.: Uniform interpolation by resolution in modal logic. In: European Workshop on Logics in Artificial Intelligence. pp. 219–231. Springer (2008)

10. Horrocks, I., Hustadt, U., Sattler, U., Schmidt, R.: Computational modal logic. In: Handbook of Modal Logic, vol. 3, pp. 181–245. Elsevier (2007)
11. Koopmann, P., Schmidt, R.A.: Uniform interpolation and forgetting for $\mathcal{ALC}$ Ontologies with ABoxes. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. vol. 29, p. 175–181 (2015)
12. Ludwig, M., Konev, B.: Practical uniform interpolation and forgetting for $\mathcal{ALC}$ TBoxes with applications to logical difference. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014. pp. 318–327 (2014)
13. Schmidt, R.A.: The Ackermann approach for modal logic, correspondence theory and second-order reduction. Journal of Applied Logic 10(1), 52–74 (2012)
14. Visser, A.: Bisimulations, model descriptions and propositional quantifiers. Logic Group Preprint Series 161 (1996)
15. Wolter, F.: Fusions of modal logics revisited. In: Advances in Modal Logic. pp. 361–379. CSLI (1998)
16. Zhao, Y., Schmidt, R.A.: Concept forgetting in $\mathcal{ALCOI}$-ontologies using an $\mathcal{A}$ckermann approach. In: Proc. ISWC 2015. Lecture Notes in Computer Science, vol. 9366, pp. 587–602. Springer (2015)