Uniform Interpolation from Cyclic Proofs: The Case of Modal Mu-Calculus

Bahareh Afshari[1,2(✉)], Graham E. Leigh[2], and Guillermo Menéndez Turata[1]

[1] Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands
{b.afshari,g.m.t.menendezturata}@uva.nl
[2] Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg, Gothenburg, Sweden
graham.leigh@gu.se

Abstract. We show how to construct uniform interpolants in the context of the modal mu-calculus. D'Agostino and Hollenberg (2000) were the first to prove that this logic has the uniform interpolation property, employing a combination of semantic and syntactic methods. This article outlines a purely proof-theoretic approach to the problem based on insights from the cyclic proof theory of mu-calculus. We argue the approach has the potential to lend itself to other temporal and fixed point logics.

Keywords: Modal mu-calculus · Sequent calculus · Uniform interpolation · Cyclic proofs

1 Introduction

Uniform interpolation is frequently listed among the most desirable properties a logic may have. Let $\mathsf{Voc}(\varphi)$ denote the non-logical vocabulary of a formula φ.[1] A logic has the uniform interpolation property if given any formula φ and vocabulary $V \subseteq \mathsf{Voc}(\varphi)$, there exists a formula I with $\mathsf{Voc}(I) \subseteq V$, the *uniform interpolant*, such that for every ψ with $\mathsf{Voc}(\psi) \cap \mathsf{Voc}(\varphi) \subseteq V$ we have

$$\varphi \to \psi \text{ is valid iff } I \to \psi \text{ is valid.}$$

Upon inspection one sees that uniform interpolation is tightly knitted to deeper semantic considerations. Uniform interpolants simulate quantification. For example, in the case of propositional logic, if $\mathsf{Voc}(\varphi) \setminus V$ is a set of propositional constants $\{p_1, \dots, p_k\}$ then the formula I above expresses $\exists p_1 \cdots \exists p_k \varphi$. This also

[1] This definition depends on the choice of underlying logic. For example, in the case of polymodal logic, $\mathsf{Voc}(\varphi)$ is the set of propositional constants and modal actions occurring in φ.

Supported by the Knut and Alice Wallenberg Foundation [2015.0179] and the Swedish Research Council [2016-03502 & 2017-05111]. The authors would like to thank the anonymous referees for their valuable comments.

A. Das and S. Negri (Eds.): TABLEAUX 2021, LNAI 12842, pp. 335–353, 2021.
https://doi.org/10.1007/978-3-030-86059-2_20

somewhat goes to explain why it is a challenging question: it does not reside entirely in the realm of syntax or semantics, thereby limiting the techniques available to tackle the problem.

A considerable body of research has been devoted to studying interpolation properties of logics, and the landscape is moderately clear [13,24]. Nevertheless, though perhaps not so surprising, proofs of uniform interpolation differ wildly from one system to another, leaving the question yet open for a number of interesting logical systems. There have been efforts to find general frameworks to attack the problem. Iemhoff [25,26] identifies sufficient (but not necessary) conditions on the form of proof systems that entail uniform interpolation. In modal logics, uniform interpolation is intimately connected to the definability of bisimulation quantifiers [16,43].

From a proof-theoretic perspective, the idea that uniform interpolation is tied to provability is a natural one. Thinking about Craig interpolation for the moment, if a 'nice' proof of a valid implication $\varphi \to \psi$ is available, one may succeed in defining an interpolant by induction on the proof-tree, starting from leaves and proceeding to the implication at the root. This method has recently been applied even to fixed point logics admitting cyclic proofs [2,38]. In contrast, for uniform interpolation, there is no single proof to work from but a collection of proofs to accommodate: a witness to each valid implication $\varphi \to \psi$ where the vocabulary of ψ is constrained. Working over a set of prospective proofs and relying on the structural properties of sequent calculus is the essence of Pitts' seminal result on uniform interpolation for intuitionistic logic [35].

In this article, we adapt Pitts' technique to the modal μ-calculus, a fixed point modal logic with an elegant mathematical theory that holds a prominent place among temporal logics. Uniform interpolation for the modal μ-calculus was established by D'Agostino and Hollenberg [14]. Their proof utilises modal automata [27] to show definability of bisimulation quantifiers in modal μ-calculus [35,43] (see also [15,16]). With this form of second-order quantification, uniform interpolants can be readily defined.

One may wonder why Pitts' method has not been exploited for the modal μ-calculus. A notable appeal of such a syntactic method is the direct construction of interpolants, without recourse to intermediate structures such as automata. One possible answer is that the method applies only in a setting where one can argue inductively over the cut-free derivations of a *finitary* system. In other words, the approach is helpless in the context of non-analytic or infinitary proofs.

While it is feasible to design analytic (complete) tableaux systems for many fixed point logics, they are often infinitary, i.e., the proof-tree can have infinite branches. The first deductive system for the μ-calculus for which completeness was established is indeed a system of ill-founded (cut-free) derivations, due to Niwiński and Walukiewicz [34]. With the advance of cyclic proof theory in recent years, it has been possible to obtain finite proof graphs that can witness validity. For modal μ-calculus specifically, an annotated goal-oriented system was given by Jungteerapanich [28] and Stirling [41] and it is a reformulation of that system which is utilised in the present work. A different cyclic proof system based on the

Jungteerapanich–Stirling system was exploited to establish Lyndon interpolation for the modal μ-calculus [2], a strengthening of Craig interpolation not implied by uniform interpolation. It is unclear whether the argument in [2] can be applied to uniform interpolation as the proof system employed lacks the requisite uniformity for a Pitts-style treatment.

The main ideas and concepts we present are not specific to modal μ-calculus but do rely on two of its essential features: the existence of (cyclic) analytic tableaux and expression of fixed points. As such we anticipate that the main argument is applicable to other logics admitting regular proof trees. That project, however, is reserved for future investigation.

Outline. In the next section we give a brief account of the syntax of μ-calculus and its equivalent formulation in terms of systems of equations which greatly facilitate defining interpolants. Section 3 presents our version of the Jungteerapanich–Stirling calculus and related concepts including that of a proof invariant and proofs in normal form. In Sect. 4 we describe the overall idea of the method and a key ingredient of our approach: construction of an interpolation 'template' for a formula encoding information about prospective proofs involving this formula. Section 5 is dedicated to the definition of the uniform interpolant and the verification of the main theorem is carried out in the subsequent section. In the conclusion we expand on some points already touched on in the introduction and further research questions of interest.

2 The Modal μ-calculus

Fix a countably infinite set Var of *variables* $X, Y, \dots$ and a set Act of *modal actions* $a, b, \dots$. The *formulas of the modal μ-calculus* are given by the following grammar:

$$\varphi := \top \mid \bot \mid X \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [a]\varphi \mid \langle a \rangle \varphi \mid \mu X \varphi \mid \nu X \varphi,$$

where X ranges over Var and a over Act. The language does not contain propositional constant symbols as these may be encoded by additional modal actions.

Formulas are denoted by lower-case Greek letters $\varphi, \psi, \chi, \dots$, and finite sets of formulas by upper-case Greek letters $\Gamma, \Delta, \Sigma, \dots$. An *atom* is either $\top$ or $\bot$. A *quantifier-free* formula is one built from atoms and variables via modal operators $[a]$, $\langle a \rangle$ and connectives $\wedge$ and $\vee$. A $[a]$-*formula* is one of the form $[a]\varphi$. A $[\cdot]$-*formula* is a $[a]$-formula for any $a \in \mathsf{Act}$. $\langle a \rangle$- and $\langle \cdot \rangle$-formulas are defined analogously. A *modal* formula is either a $[\cdot]$- or a $\langle \cdot \rangle$-formula. Define $[a]\Gamma = \{[a]\gamma \mid \gamma \in \Gamma\}$ and $\langle a \rangle \Gamma$ similarly.

An occurrence of a variable X in a formula φ is *bound* if it is within the scope of a quantifier σX for some $\sigma \in \{\mu, \nu\}$, and it is *free* otherwise. A formula is *closed* if no variable occurs free in it. A set of formulas is *closed* if every element is closed. We say that φ is *well-named* if no variable in φ occurs both free and bound and no variable is bound more than once. Each bound variable X of a well-named formula φ uniquely identifies a subformula $\sigma_X X \varphi_X$ of φ. When no ambiguity arises we express this association as $X =_{\sigma_X} \varphi_X$. A formula

φ is *guarded* if, for any subformula $\sigma X\psi$ of φ, every occurrence of X in ψ is within a modal subformula of ψ. It is well known that every formula of the modal μ-calculus is equivalent to a guarded one. A finite set of formulas Γ is *closed/guarded/well-named* if and only if the conjunction of elements $\bigwedge_{\gamma\in\Gamma}\gamma$ is.

It will be convenient to utilise a more succinct notation for μ-calculus formulas. Modal/hierarchical equational systems provide an expressively equivalent formalism meeting our needs. For the purpose of this article, a *modal equational system* is a pair $(\varphi, \mathcal{E})$ where φ is a quantifier-free formula over a set of variables $V_\mathcal{E}$ and $\mathcal{E}$ is a set of equations $\{X =_{p_X} \varphi_X \mid X \in V_\mathcal{E}\}$ where φ_X is a quantifier-free formula over $V_\mathcal{E}$ and $p_X \in \mathbb{N}$ for each $X \in V_\mathcal{E}$. The system $(\varphi, \mathcal{E})$ uniquely determines a μ-calculus formula $\varphi_\mathcal{E}$ given as follows. Let $V_\mathcal{E} = \{X_0, \ldots, X_n\}$ where $p_{X_i} \leq p_{X_j}$ for each $i < j \leq n$. The formula $\varphi_\mathcal{E}$ is specified by recursively substituting $\sigma_i X_i \varphi_{X_i}$ for X_i in all equations starting from $i = n$ where $\sigma_i = \mu$ iff p_{X_i} is odd. In other words, the equations $X_i =_{p_{X_i}} \varphi_{X_i}$ of $(\varphi, \mathcal{E})$ correspond to associations $X_i =_{\sigma_i} \hat{\varphi}_{X_i}$ in $\varphi_\mathcal{E}$ where $\hat{\varphi}_{X_i}$ is a substitution instance of φ_{X_i}, and the ordering of the priorities corresponds to the subsumption order of $\varphi_\mathcal{E}$. We refer the reader to [21, §8.3.4] for details. In the following, whenever a system $(\varphi, \mathcal{E})$ is referred to as a formula we mean the formula $\varphi_\mathcal{E}$ described above.

We assume the reader is familiar with denotational semantics for formulas of μ-calculus over, for example, labelled transition systems (see, e.g. [21, §8.1.2]). A formula whose denotation over every labelled transition system is the set of states of the system is called *valid*.

3 The JS proof system

Based on tableaux for satisfiability by Jungteerapanich [28], Stirling [41] introduces a sound and complete tableau-style proof system for the modal μ-calculus in which formulas are enriched with annotations that keep track of fixed point unfoldings. In this section we present a Gentzen-style, two-sided version of the Jungteerapanich–Stirling system.

A *(plain) sequent* is a pair (Γ, Δ), henceforth written $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite sets of formulas. A sequent $\Gamma \Rightarrow \Delta$ is *closed/guarded/well-named* iff $\Gamma \cup \Delta$ is closed/guarded/well-named. A closed sequent $\Gamma \Rightarrow \Delta$ is *valid* if the induced formula $\bigwedge_{\gamma\in\Gamma}\gamma \to \bigvee_{\delta\in\Delta}\delta$ is valid, where the connective $\to$ is defined in terms of disjunction and negation (the latter expressed via de Morgan duality). When working with sequents we shall frequently abbreviate $\Gamma \cup \Delta$ to Γ, Δ and $\Gamma, \{\varphi\}$ to Γ, φ.

For every $X \in \mathsf{Var}$ let a countably infinite set $\mathsf{N}_X = \{x_0, x_1, \ldots\}$ of *names* for X be fixed such that $\mathsf{N}_X \cap \mathsf{N}_Y = \varnothing$ if $X \neq Y$. We denote names for variables $X, Y, Z, \ldots$ by $x, y, z, \ldots$ respectively (possibly with indices) and let $\mathsf{N} = \bigcup_{X\in\mathsf{Var}} \mathsf{N}_X$. An *annotation* is a finite sequence of pairwise distinct names in N. Given an annotation u we denote by $|u|$ the length of u. An *annotated* formula is a pair (u, φ), henceforth written φ^u, where φ is a formula and u is an annotation for variables occurring in φ. A name x occurs in Γ if x occurs in the annotation of some formula in Γ. An *(annotated) sequent* is a triple (Θ, Γ, Δ), henceforth written $\Theta : \Gamma \Rightarrow \Delta$, where Θ is an annotation in names for variables in $\Gamma \cup \Delta$, and Γ and Δ are finite sets of annotated formulae such that every name in Θ occurs in $\Gamma \cup \Delta$. The annotation Θ is called the *control* of $\Theta : \Gamma \Rightarrow \Delta$. We identify annotated sequents whose control is empty with plain sequents.

Let φ be closed and well-named. Fix an arbitrary linear ordering of the variables in φ, say $X_1 \lhd X_2 \lhd \cdots \lhd X_n$, compatible with the subsumption ordering on φ, i.e., such that $i < j$ implies φ_{X_i} is not a subformula of φ_{X_j}. If φ is given as an equational system, $\lhd$ can be chosen as any linear order such that $p_X < p_Y$ implies $X \lhd Y$. Given an annotation u for variables in φ, we denote by $u{\restriction}X_i$ the result of removing from u all names for $X_{i+1},\ldots,X_n$.

We now define the JS sequent calculus. The system operates on annotated sequents, i.e., expressions $\Theta : \Gamma \Rightarrow \Delta$ defined above. The axioms and rules of JS are given in Fig. 1. Applications of the rules are subject to three restrictions:

- Θ' is the subsequence of Θ given by removing any name which does not occur in the sequent whose control is Θ'. This applies to the rules LW, RW, $\mathsf{L}\mu$, $\mathsf{R}\mu$, $\mathsf{L}\nu$, $\mathsf{R}\nu$, LMod, RMod, LReset and RReset.
- In $\mathsf{L}\mu$ and $\mathsf{R}\nu$ x is a name for the variable X not occurring in Θ, and $\Theta' x$ is the concatenation of Θ' and the annotation consisting of the single name x.
- In LReset_z and RReset_z the names z, z_1, $\ldots$, z_k all name the same variable; the other annotations (u, u_1, $\ldots$, u_k) are arbitrary.
- In LReset_z the name z may not occur in Γ and in RReset_z the name z may not occur in Δ.

By Mod (Reset) we denote either LMod or RMod (resp., LReset or RReset).

Definition 1. *A derivation of a closed and well-named sequent $\Gamma \Rightarrow \Delta$ is a finite tree P of annotated sequents in accordance with the rules of* JS *(subject to the restrictions above) with root $\Gamma \Rightarrow \Delta$, together with a map $l \mapsto c_l$ which assigns to every non-axiomatic leaf $l \in P$ a vertex $c_l <_P l$, where $<_P$ denotes the ancestor relation of P, such that the sequents at l and c_l are identical. We refer to c_l as the* companion *of l and to l as a* repeat.

A proof in the JS calculus is a derivation for which all repeat leaves fulfil a correctness condition that we now define.

Definition 2 (Invariant; Successful repeat). *Let P be a* JS *derivation and l a repeat leaf of P with companion c_l. Let Θ be the longest common prefix of all controls on the path $[c_l, l]_P$ from c_l to l. The* invariant *of l, in symbols $\mathsf{inv}_P(l)$, is the shortest prefix of Θ of the form ux where Reset_x occurs on the path $[c_l, l]$. If no such prefix exists, define $\mathsf{inv}_P(l) = \Theta$. The repeat leaf l is* successful *iff $\mathsf{inv}_P(l)$ is of the first form above.*

The notion of an invariant for a repeat leaf does not feature in the presentations of the calculus in [28,41]. However, it is an easy exercise to show that a repeat is successful in the sense above if and only if it is successful in the sense of [41].

Definition 3 (JS proof). *A proof is a* JS *derivation P such that every non-axiomatic leaf is a successful repeat.*

$$\mathsf{Ax}_\bot\ \frac{}{\varTheta : \varGamma, \bot^u \Rightarrow \varDelta} \qquad \mathsf{Ax}_\top\ \frac{}{\varTheta : \varGamma \Rightarrow \top^u, \varDelta}$$

$$\mathsf{LW}\ \frac{\varTheta' : \varGamma \Rightarrow \varDelta}{\varTheta : \varGamma, \varPi \Rightarrow \varDelta} \qquad \mathsf{RW}\ \frac{\varTheta' : \varGamma \Rightarrow \varDelta}{\varTheta : \varGamma \Rightarrow \varSigma, \varDelta}$$

$$\mathsf{L}\wedge\ \frac{\varTheta : \varGamma, \varphi^u, \psi^u \Rightarrow \varDelta}{\varTheta : \varGamma, \varphi \wedge \psi^u \Rightarrow \varDelta} \qquad \mathsf{R}\vee\ \frac{\varTheta : \varGamma \Rightarrow \varphi^u, \psi^u, \varDelta}{\varTheta : \varGamma \Rightarrow \varphi \vee \psi^u, \varDelta}$$

$$\mathsf{L}\vee\ \frac{\varTheta : \varGamma, \varphi^u \Rightarrow \varDelta \qquad \varTheta : \varGamma, \psi^u \Rightarrow \varDelta}{\varTheta : \varGamma, \varphi \vee \psi^u \Rightarrow \varDelta} \qquad \mathsf{R}\wedge\ \frac{\varTheta : \varGamma \Rightarrow \varphi^u, \varDelta \qquad \varTheta : \varGamma \Rightarrow \psi^u, \varDelta}{\varTheta : \varGamma \Rightarrow \varphi \wedge \psi^u, \varDelta}$$

$$\mathsf{L}\mu\ \frac{\varTheta' x : \varGamma, \varphi^{(u\restriction X)x} \Rightarrow \varDelta}{\varTheta : \varGamma, X^u \Rightarrow \varDelta}\ X =_\mu \varphi \qquad \mathsf{R}\mu\ \frac{\varTheta' : \varGamma \Rightarrow \varphi^{u\restriction X}, \varDelta}{\varTheta : \varGamma \Rightarrow X^u, \varDelta}\ X =_\mu \varphi$$

$$\mathsf{L}\nu\ \frac{\varTheta' : \varGamma, \varphi^{u\restriction X} \Rightarrow \varDelta}{\varTheta : \varGamma, X^u \Rightarrow \varDelta}\ X =_\nu \varphi \qquad \mathsf{R}\nu\ \frac{\varTheta' x : \varGamma \Rightarrow \varphi^{(u\restriction X)x}, \varDelta}{\varTheta : \varGamma \Rightarrow X^u, \varDelta}\ X =_\nu \varphi$$

$$\mathsf{LMod}\ \frac{\varTheta' : \varGamma, \varphi^u \Rightarrow \varDelta}{\varTheta : [a]\varGamma, \varPi, \langle a\rangle\varphi^u \Rightarrow \langle a\rangle\varDelta, \varSigma} \qquad \mathsf{RMod}\ \frac{\varTheta' : \varGamma \Rightarrow \varphi^u, \varDelta}{\varTheta : [a]\varGamma, \varPi \Rightarrow [a]\varphi^u, \langle a\rangle\varDelta, \varSigma}$$

$$\mathsf{LReset}_z\ \frac{\varTheta' : \varGamma, \varphi_1^{uz}, \dots, \varphi_k^{uz} \Rightarrow \varDelta}{\varTheta : \varGamma, \varphi_1^{uzz_1u_1}, \dots, \varphi_k^{uzz_ku_k} \Rightarrow \varDelta} \qquad \mathsf{RReset}_z\ \frac{\varTheta' : \varGamma \Rightarrow \varphi_1^{uz}, \dots, \varphi_k^{uz}, \varDelta}{\varTheta : \varGamma \Rightarrow \varphi_1^{uzz_1u_1}, \dots, \varphi_k^{uzz_ku_k}, \varDelta}$$

Fig. 1. Rules of the JS system.

The role of the annotations is to keep track of unfoldings of fixpoint variables. Figure 2 shows a proof of the valid formula $\nu Z\mu X([a]Z \vee \langle a\rangle X)$ with corresponding modal equational system $(Z, \{Z =_0 X,\ X =_1 [a]Z \vee \langle a\rangle X\})$. The name z is preserved and reset in between the companion node and the repeat, so the repeat is successful.

We write $P \vdash \varGamma \Rightarrow \varDelta$ to express that P is a proof of the (closed, well-named) sequent $\varGamma \Rightarrow \varDelta$, and $\mathsf{JS} \vdash \varGamma \Rightarrow \varDelta$ if and only if there exists a proof of $\varGamma \Rightarrow \varDelta$. The following can be proved by reduction of the two-sided calculus JS to its one-sided fragment and appealing to the main result of [41].

Theorem 1. JS *is sound and complete with respect to validity for closed, well-named and guarded sequents.*

Before turning to the statement and proof of uniform interpolation, we present some important restrictions on applications of the rules of JS which does not affect the completeness theorem above. Following these observations, we establish a property of JS proofs which plays a crucial role in our proof of the uniform interpolation property. The property in question is that a finite unfolding of a JS proof – given by identifying repeat leaves with their companions – is a JS proof, and an invariant for a leaf in the unfolding is the invariant of some leaf in the original proof.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{z : \varnothing \Rightarrow X^{z}}{zz' : \varnothing \Rightarrow X^{zz'}}\ \mathsf{RReset}_z}{zz' : \varnothing \Rightarrow X^{zz'}, X^{z}}\ \mathsf{RW}}{z : \varnothing \Rightarrow Z^{z}, X^{z}}\ \mathsf{R}\nu}{z : \varnothing \Rightarrow [a]Z^{z}, \langle a\rangle X^{z}}\ \mathsf{RMod}}{z : \varnothing \Rightarrow ([a]Z \vee \langle a\rangle X)^{z}}\ \mathsf{R}\vee}{z : \varnothing \Rightarrow X^{z}}\ \mathsf{R}\mu}{: \varnothing \Rightarrow Z}\ \mathsf{R}\nu
$$

Fig. 2. A JS proof of $\nu Z \mu X([a]Z \vee \langle a\rangle X)$.

The structural rules of weakening are also implicit in our formulation of the modal rules LMod and RMod. Any JS proof can be converted into a proof without LW or RW. Although the argument is straightforward, some care is required as altering sequents in a JS proof can result in leaves and companions no longer being identical sequents. Weakening, however, serves a special purpose in the Jungteerapanich–Stirling calculus as a rule for maintaining a bound on the size of sequents in proof-search. To show completeness for JS a proof is constructed in which the left and right weakening rules are utilised in a specific form for eliminating (reading the rule from conclusion to premise) an occurrence of a formula if it occurs with two (distinct) annotations:

$$
\mathsf{LThin}\ \frac{\Theta' : \Gamma, \varphi^{u} \Rightarrow \Delta}{\Theta : \Gamma, \varphi^{u}, \varphi^{v} \Rightarrow \Delta}\ u \sqsubset_{\Theta} v \qquad \mathsf{RThin}\ \frac{\Theta' : \Gamma \Rightarrow \varphi^{u}, \Delta}{\Theta : \Gamma \Rightarrow \varphi^{u}, \varphi^{v}, \Delta}\ u \sqsubset_{\Theta} v
$$

LThin and RThin are referred to as *thinning rules*. As before, in both inferences Θ' denotes the result of removing from Θ any name which does not occur in $\Gamma \cup \Delta \cup \{\varphi^{u}\}$. The relation $\sqsubset_{\Theta}$ is a total ordering on subsequences of Θ defined as follows. If Θ is an annotation and u, v are subsequences of Θ, set $u <_{\Theta} v$ iff u precedes v in the lexicographic ordering induced by Θ. Then define $u \sqsubset_{\Theta} v$ as either $u <_{\Theta} v$ or there is some variable X such that $v{\restriction}X$ is a proper prefix of $u{\restriction}X$. We refer the reader to [29, §4.3] for the proof that $\sqsubset_{\Theta}$ is a total order on subsequences of Θ.

In the presence of the thinning rules, Stirling's completeness proof for the one-sided system shows it is possible to dispense entirely with weakening, both in the explicit form of LW and RW and implicitly in the modal inferences. Moreover, provided sequents are guarded it suffices that the conclusion to the modal rules LMod and RMod is a sequent of modal formulas and atoms only.

Further restrictions can be imposed on proofs in JS. These are outlined by the next definition. The first three restrictions mirror standard conditions that can be imposed on analytic sequent calculi for basic modal logic. Conditions 4 and 5 enforce uniformity on the rules manipulating annotations. The final condition places a similar condition on the logical inferences, with the effect that two incomparable vertices of a normal proof which are labelled by the same annotated sequent have identical sub-proof up to repeat leafs. The requirement is trivial for proofs of quantifier-free formulas; for quantified formulas it is a corollary of the fact that JS proofs are closed under unravelling repeat leaves.

Definition 4 (Normal proof). *A JS proof P is normal if the following conditions hold of P.*

1. *The only applications of weakening are the thinning rules.*
2. LMod *or* RMod *is permitted only in cases where (referring to the form of the rule in Fig. 1) Π consists of only $\top$, $\langle\cdot\rangle$-formulas and $[c]$-formulas for $c \neq a$, and Σ of only $\bot$, $[\cdot]$-formulas, and $\langle c\rangle$-formulas for $c \neq a$.*
3. *Any sequent which is an instance of an axiom is a leaf.*
4. *In instances of $\mathsf{L}\mu$ and $\mathsf{R}\nu$, x is the first name in N_X not occurring in Θ.*
5. *If a sequent $\Theta : \Gamma \Rightarrow \Delta$ in P can be realised as the conclusion of an instance of* LThin, RThin, LReset *or* RReset *then the sequent is the conclusion of this rule in P, with the thinning rules having precedence over reset rules.*
6. *Any two non-repeat vertices of P labelled by the same sequent are instances of the same rule instantiation.*

The following is a direct consequence of the completeness theorem for the one-sided fragment of JS by Stirling [41].

Theorem 2. *A closed, well-named and guarded sequent $\Gamma \Rightarrow \Delta$ is valid iff there exists a normal JS proof of $\Gamma \Rightarrow \Delta$.*

We conclude this section with two results concerning the definition of invariant of a repeat leaf.

Fix a JS derivation P and let Rep_P be the set of repeat leaves of P. We define two relations on elements of Rep_P. The first is a reflexive and transitive relation $\preceq$ given by setting $l \preceq l'$ if $\mathsf{inv}_P(l)$ is a prefix of $\mathsf{inv}_P(l')$. The second relation $\rightsquigarrow$ is defined as reachability between repeat vertices: $l \rightsquigarrow l'$ iff $c_l <_P l'$, i.e., there is a (simple) path in P from the companion of l to l'. Note that $\rightsquigarrow$ need not be symmetric or transitive.

The following two observations link invariants to the 'unfolding' of JS proofs. Both results are immediate consequences of our notion of invariant.

Proposition 1. *For every infinite $\rightsquigarrow$-chain $l_0 \rightsquigarrow l_1 \rightsquigarrow \cdots$ there exists $k \geq 0$ such that $l_k \preceq l_j$ for all $j \geq k$.*

Proposition 2. *Let P be a proof and $l \in \mathrm{Rep}_P$. The result of inserting a copy of the sub-derivation of P with root c_l at the leaf l is a JS proof. Any invariant of a repeat in the resulting proof is an invariant of a repeat in P.*

4 Uniform Interpolation

With the proof system now fixed, we present the statement of uniform interpolation that will be proved. The *vocabulary* of a formula φ, in symbols $\mathsf{Voc}(\varphi)$, is the set of modal actions occurring in φ. The vocabulary of a set of formulas Φ is $\mathsf{Voc}(\Phi) = \bigcup_{\varphi\in\Phi} \mathsf{Voc}(\varphi)$. In the following $\vdash \Gamma \Rightarrow \Delta$ expresses $\mathsf{JS} \vdash \Gamma \Rightarrow \Delta$.

Theorem 3. *Let Γ be a finite well-named set of modal μ-calculus formulas and $V \subseteq \mathsf{Voc}(\Gamma)$. There exists a formula I such that: (i) $\mathsf{Voc}(I) \subseteq V$, (ii) $\vdash \Gamma \Rightarrow I$, and (iii) for every Δ such that $\Gamma \Rightarrow \Delta$ is a well-named sequent and $\mathsf{Voc}(\Delta) \cap \mathsf{Voc}(\Gamma) \subseteq V$, if $\vdash \Gamma \Rightarrow \Delta$, then $\vdash I \Rightarrow \Delta$.*

We call the formula I of Theorem 3 the *(uniform) interpolant of Γ relative to V*. Mention of the fixed vocabulary V will be suppressed when it can be inferred from context. Note that the statement of uniform interpolation in Theorem 3 is equivalent to the version on page 1.

The Craig interpolation property is a special case of uniform interpolation.

Corollary 1. *If $\vdash \Gamma \Rightarrow \Delta$ and $\Gamma \Rightarrow \Delta$ is well-named then there exists a formula I such that $\mathsf{Voc}(I) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta)$, $\vdash \Gamma \Rightarrow I$ and $\vdash I \Rightarrow \Delta$.*

Proof. Let $\Gamma \Rightarrow \Delta$ be given and set $V = \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta)$. The uniform interpolant I for this choice of Γ and V satisfies the desired properties.

The remainder of this article is concerned with the proof of Theorem 3. Section 5 covers the construction of the interpolant in detail; the verification is the focus of Sect. 6. In the present section we overview the basic strategy in the simple case Γ is quantifier-free. In what immediately follows, sequents are expressions $\Theta : \Pi \Rightarrow \Delta$ where Π is a finite set of unannotated formulas and Θ only names variables in Δ. In particular, Θ is empty if Δ is empty, in which case the control will not be mentioned.

Let $V \subseteq \mathsf{Voc}(\Gamma)$ be fixed. The uniform interpolant I_Γ for Γ and V is constructed by recursion on the syntactic complexity of Γ. Preempting the incorporation of fixed points into Γ, we take a slightly less direct approach to the definition of I_Γ than necessary. Indeed, for quantifier-free sequents the construction of interpolants is among the simplest examples of obtaining uniform interpolants from a terminating proof system, as presented in [26].

We consider a derivation tree for the sequent $\Gamma \Rightarrow \varnothing$ according to the rules $\mathsf{Ax}_\bot$, $\mathsf{L}\wedge$, $\mathsf{L}\vee$ and a modification of the modal inferences which we detail shortly. A derivation tree with root $\Gamma \Rightarrow \varnothing$ that is maximal in the sense that every leaf is an instance of $\mathsf{Ax}_\bot$ or a sequent $\varnothing \Rightarrow \varnothing$ is called an *interpolation template for Γ*. The modal inference we utilise is an amalgamation of the left and right modal inferences, named the *global modal rule*, GMod:

$$\mathsf{GMod}\ \frac{\{\Gamma_i, \pi \Rightarrow \varnothing \mid i \leq n \wedge \pi \in \Pi_i\} \quad \{\Gamma_i \Rightarrow \varnothing \mid i \leq n \wedge a_i \in V\} \quad \varnothing \Rightarrow \varnothing}{[a_0]\Gamma_0, \langle a_0 \rangle \Pi_0, \ldots, [a_n]\Gamma_n, \langle a_n \rangle \Pi_n, \Sigma \Rightarrow \varnothing}$$

Applications of the rule are subject to the restriction that $a_0, \ldots, a_n$ are distinct actions and $\Sigma \subseteq \{\top\}$. The sets Γ_i and Π_i are permitted to be empty.

Unlike LMod and RMod, the GMod rule is branching and involves three forms of premise: sequents $\Gamma_i, \pi \Rightarrow \varnothing$ for $i \leq n$ and $\pi \in \Pi_i$, called *active premises*; sequents $\Gamma_i \Rightarrow \varnothing$ for $i \leq n$ and $a_i \in V$, called *passive premises*; and the *trivial premise* $\varnothing \Rightarrow \varnothing$. The active and passive premises encode maximal instances of LMod and RMod respectively assuming an appropriate (but unspecified) instantiation of the consequent in both premise and conclusion. The trivial premise corresponds to an instance of RMod for an action label in $\mathsf{Act} \setminus V$, as any such application of RMod yields a premise with empty antecedent. In practice, the trivial premise may be safely ignored because it takes no part in the construction of the uniform interpolant; its presence is merely a technical convenience for tracing paths in a proof of $\Gamma \Rightarrow \Delta$ onto the interpolation template.

The desired connection between the three modality rules is formally expressed by the next lemma, the proof of which is straightforward. Restricting to the case $\Delta = \varnothing$, the lemma shows that a proof of a sequent $\Gamma \Rightarrow \varnothing$ (if one exists) is encoded within an interpolation template for Γ.

Lemma 1. *Let Γ be quantifier-free and $V \subseteq \mathsf{Voc}(\Gamma)$. There exists an interpolation template T for Γ such that for every Δ, if $\Gamma \Rightarrow \Delta$ is valid and $\mathsf{Voc}(\Delta) \cap \mathsf{Voc}(\Gamma) \subseteq V$, then there exists a normal proof $P \vdash \Gamma \Rightarrow \Delta$ satisfying the following condition. For every path $(\Gamma_i \Rightarrow \Delta_i)_{i<N}$ through P there exists a path $(\Gamma_{k_i} \Rightarrow \varnothing)_{i<N'}$ through T where $(k_i)_{i<N'}$ is strictly increasing and $\Gamma_{k_i+j} = \Gamma_{k_i}$ for all $j < k_{i+1} - k_i$.*

From an interpolation template T_Γ satisfying the above lemma it is possible to read off a uniform interpolant for Γ and V. We show that each vertex of $u \in T_\Gamma$ can be associated a formula I_u fulfilling the three conditions of Theorem 3 relative to the sequent at u. The construction of I_u begins at the leaves of T_Γ. In the following, u is assumed to be labelled by the sequent $\Pi \Rightarrow \varnothing$.

If u is a leaf, then either $\bot \in \Pi$ or $\Pi = \varnothing$. In the former case set $I_u = \bot$, in the latter $I_u = \top$. Either way, $\Pi \Rightarrow I_u$ is an axiom of JS. Now suppose u is a non-leaf vertex of T_Γ. Thus $\Pi \Rightarrow \varnothing$ together with the labels of immediate successors to u corresponds to an instance of either $\mathsf{L}\wedge$, $\mathsf{L}\vee$ or GMod. We consider each case in turn. The simpler of the three inferences is $\mathsf{L}\wedge$. In this case u has a unique successor, v say, in T_Γ which we may assume is a sequent $\Pi', \varphi, \psi \Rightarrow \varnothing$ where $\Pi = \Pi' \cup \{\varphi \wedge \psi\}$. Define $I_u = I_v$. That $\vdash \Pi \Rightarrow I_u$ follows immediately from the induction hypothesis. A more informative case is $\mathsf{L}\vee$. Here u has two immediate successors in T_Γ, u_0 and u_1 say, labelled by $\Pi', \varphi \Rightarrow \varnothing$ and $\Pi', \psi \Rightarrow \varnothing$ respectively where $\Pi = \Pi' \cup \{\varphi \vee \psi\}$. Choose $I_u = I_{u_0} \vee I_{u_1}$. From $\vdash \Pi', \varphi \Rightarrow I_{u_0}$ and $\vdash \Pi', \psi \Rightarrow I_{u_1}$ we deduce $\vdash \Pi \Rightarrow I_u$ by applications of RW, $\mathsf{L}\vee$ and $\mathsf{R}\vee$.

The final case in the construction is an instance of GMod which we may assume has the form on the previous page, i.e., Π is

$$\Pi = [a_0]\Gamma_0, \langle a_0\rangle\Pi_0, \ldots, [a_n]\Gamma_n, \langle a_n\rangle\Pi_n, \Sigma \tag{1}$$

where a_0, …, a_n are distinct modal actions and $\Sigma \subseteq \{\top\}$. Let the actions be ordered such that $V = \{a_0, \ldots, a_k\}$ for some $k \leq n$. For each $i \leq n$ and $\pi \in \Pi_i$ let u_i^π be the immediate successor of u for the active premise $\Gamma_i, \pi \Rightarrow \varnothing$ and u_i the immediate successor for the passive premise $\Gamma_i \Rightarrow \varnothing$. We may ignore the trivial premise. A natural candidate for I_u is the formula

$$I_u^* = \bigwedge_{i \leq k}([a_i]I_{u_i} \wedge \bigwedge_{\pi \in \Pi_i} \langle a_i\rangle I_{u_i^\pi}) \tag{2}$$

Restricting the conjunction to $i \leq k$ ensures $\mathsf{Voc}(I_u^*) \subseteq V$. Given $\vdash \Gamma_i, \pi \Rightarrow I_{u_i^\pi}$ for each $i \leq n$ and $\pi \in \Pi_i$, an application of LMod yields $\vdash \Pi \Rightarrow \langle a_i \rangle I_{u_i^\pi}$. Likewise, $\Pi \Rightarrow [a_i] I_{u_i}$ can be deduced from $\Gamma_i \Rightarrow I_{u_i}$ by RMod. So $\vdash \Pi \Rightarrow I_u^*$.

It is not difficult to show however that requirement (iii) of Theorem 3 can fail for this choice of interpolant, for example if $\Pi = \langle b \rangle \varphi$ where $b \notin V$ and φ is unsatisfiable. The conjunction in (2) would be empty and $I_u^* = \top$. So $\vdash \Pi \Rightarrow \bot$ but not $\vdash I_u^* \Rightarrow \bot$. The failure of I^* to be a uniform interpolant of Π stems from the possibility of modal actions outside V being relevant in a proof of $\Pi \Rightarrow \Delta$. If Γ_i, π is unsatisfiable for some $\pi \in \Pi_i$, then $\bot$ suffices as the choice of I_u but I_u^* may not. With this consideration in mind, define $I_u = \bot$ if $\vdash \Gamma_i, \pi \Rightarrow \varnothing$ for some $k \leq i \leq n$ and some $\pi \in \Pi_i$. Otherwise, set $I_u = I_u^*$.[2]

Let us conclude by establishing condition (iii) of the theorem. Let Δ be any set of guarded formulas such that $\mathsf{Voc}(\Delta) \cap \mathsf{Voc}(\Gamma) \subseteq V$ and suppose $\vdash \Gamma \Rightarrow \Delta$. Lemma 1 provides a normal proof $P \vdash \Gamma \Rightarrow \Delta$ and an assignment $f \colon P \to T_\Gamma$ of vertices in the interpolation template to vertices in P such that $u \in P$ and $f(u) \in T_\Gamma$ have the same antecedent. Moreover, paths through P correspond to paths through T_Γ (the latter expanded with repetitions). If $\Theta_u : \Gamma_u \Rightarrow \Delta_u$ denotes the label of $u \in P$, we claim $\vdash \Theta_u : I_{f(u)} \Rightarrow \Delta_u$.

Suppose $u \in P$ is a leaf. Then $\Theta_u : \Gamma_u \Rightarrow \Delta_u$ is either an instance of an axiom or is a repeat leaf of P. In the case of an axiom $\vdash \Theta_u : I_{f(u)} \Rightarrow \Delta_u$ holds because $I_{f(u)} = \bot$ if $\bot \in \Gamma_u$. If u is a repeat we observe that since Δ is guarded and Γ is quantifier-free, $\Gamma_u = \varnothing$. But then $I_{f(u)} = \bot$ and $\vdash \Theta_u : I_{f(u)} \Rightarrow \Delta_u$.

The only non-leaf case which is not straightforward is the modal rules, LMod or RMod. By the normality of P, the sets Γ_u and Δ_u are modal. Suppose

$$\Gamma_u = [a_0]\Gamma_0, \langle a_0 \rangle \Pi_0, \ldots, [a_n]\Gamma_n, \langle a_n \rangle \Pi_n, \Sigma$$
$$\Delta_u = [b_0]\Lambda_0, \langle b_0 \rangle \Delta_0, \ldots, [a_m]\Lambda_m, \langle b_m \rangle \Delta_m, \Sigma'$$

where we assume $k \leq \min\{m, n\}$ is such that $a_i = b_i$ for each $i \leq k$ and $\{a_{k+1}, \ldots, a_n\} \cap \{b_{k+1}, \ldots, b_n\} = \varnothing$. The premise to this inference can take one of four forms, depending on which formula of $\Gamma_u \Rightarrow \Delta_u$ is principal and which modal action the rule effected:

1. $\vdash \Gamma_i, \pi \Rightarrow \Delta_i$ for some $i \leq k$ and $\pi \in \Pi_i$.
2. $\vdash \Gamma_i \Rightarrow \lambda, \Delta_i$ for some $i \leq k$ and $\lambda \in \Lambda_i$.
3. $\vdash \Gamma_i, \pi \Rightarrow \varnothing$ for some $k < i \leq n$ and $\pi \in \Pi_i$.
4. $\vdash \varnothing \Rightarrow \lambda, \Delta_i$ for some $k < i \leq m$ and $\lambda \in \Lambda_i$.

If $I_{f(u)} = \bot$ then $\Theta_u : I_{f(u)} \Rightarrow \Delta_u$ is an axiom. Otherwise, $I_{f(u)} = I_{f(u)}^*$ (where vertices u_i^π and u_i in (2) refer to the active and passive premises of $f(u)$ in T_Γ) and the third scenario does not apply. In the first two cases $\Theta_u : I_{f(u)} \Rightarrow \Delta_u$ is a consequence of the induction hypothesis; in case 4, by RMod and LW.

Thus, we have shown that the formula I_r where r is the root of the interpolation template T_Γ is a uniform interpolant for Γ and V, if Γ is quantifier-free.

[2] The case distinction based on the provability of $\Gamma_i, \pi \Rightarrow \varnothing$ brings into question the computational cost of constructing uniform interpolants. Lemmas 1 and 3, however, provide that provability of sequents with empty consequent is implicit in the interpolation template.

5 Constructing the Interpolant

We are, of course, interested in obtaining interpolants for formulas containing quantifiers. The basic idea behind the interpolation template remains the same and can be generalised to incorporate the fixed point inferences $\mathsf{L}\mu$ and $\mathsf{L}\nu$, and the annotation management rules LThin and LReset. The construction, and subsequent verification, of interpolants from these templates is more subtle however. In order to ensure interpolation templates remain finite trees – so that interpolants can be defined recursively from leaf to root – it is necessary to treat them as we do cyclic proofs by permitting leaves with non-trivial sequent, i.e., sequents that are neither empty nor instances of $\mathsf{Ax}_\bot$. These leaves will be subject to path-based repeat condition in the style of JS proofs. Even with a suitable repeat condition, there remains the question of how to present an interpolant to a repeat leaf prior to knowing the intended interpolant for the companion vertex. We return to this question after clarifying the interpolation templates.

Fix a finite well-named set of guarded formulas Γ and vocabulary $V \subseteq \mathsf{Voc}(\Gamma)$. An *interpolation template for* Γ is a tree of *annotated* sequents of the form $\Theta : \Pi \Rightarrow \varnothing$ with root $\Gamma \Rightarrow \varnothing$ subject to the rules $\mathsf{L}\vee$, $\mathsf{L}\wedge$, $\mathsf{L}\mu$, $\mathsf{L}\nu$, LThin, LReset and GMod. The final rule is adapted to annotated sequents in the natural way in analogy with the rules LMod and RMod. We require interpolation templates to be a normal derivation.[3] As mentioned, three kinds of leaf are permitted in T_Γ: instances of $\mathsf{Ax}_\bot$, empty sequents $\varnothing \Rightarrow \varnothing$, and 'repeat' sequents. The requirements of a repeat is as follows: A leaf $u \in T_\Gamma$ is a repeat if and only if there exists a vertex $c_u <_{T_\Gamma} u$ (called the companion of u) labelled by the same (annotated) sequent. Every repeat leaf can be assigned an invariant according to Definition 2 which is called *successful* if the invariant ends in a name that is reset on the path between companion and leaf.

The restriction to normal proofs has the effect that interpolation templates can be assumed to be finite. The conditions, in especial conditions 4 and 5 concerning annotations, ensure that a maximal path through an interpolation template reaches either an axiom, empty sequent or a sequent which is repeated on the path. Such a repeat can be treated as a repeat leaf. The argument is identical to the proof of termination of proof search in [28,41].

Lemma 2. *Every guarded and well-named sequent $\Gamma \Rightarrow \varnothing$ admits a finite interpolation template.*

Henceforth we assume a fixed interpolation template T_Γ for a set Γ and vocabulary V. The path property of Lemma 1 also generalises to the quantified case. The result is analogous although now a path through a (normal) proof of $\Gamma \Rightarrow \Delta$ will in general trace out a path through the *unravelling* of T_Γ.

Lemma 3. *If $\Gamma \Rightarrow \Delta$ is a valid well-named and guarded sequent and* $\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta) \subseteq V$*, then there exists a normal proof $P \vdash \Gamma \Rightarrow \Delta$ such that for every*

3 We assume Definition 4 is generalised to derivations with GMod. No additional restrictions are necessary to accommodate this rule

path $(\Theta_i : \Gamma_i \Rightarrow \Delta_i)_{i \leq N}$ *through* P *there is a sequence of vertices* $(u_i)_{i \leq N'}$ *from* T_Γ *with labels* $(\Theta'_{k_i} : \Gamma_{k_i} \Rightarrow \varnothing)_{i \leq N'}$ *where for every* $i < N'$*:*

1. u_{i+1} *is an immediate successor of* u_i *or* u_i *is a repeat and* u_{i+1} *is an immediate successor of the companion of* u_i*.*
2. $k_i < k_{i+1}$*, and for each* $j < k_{i+1} - k_i$ *we have* $\Gamma_{k_i+j} = \Gamma_{k_i}$ *and* Θ'_{k_i} *is the restriction of* Θ_{k_i+j} *to names for variables in* Γ_{k_i}*.*

Given an interpolation template T_Γ, we now assign an interpolant to each vertex of T_Γ. The definition proceeds in two stages. First, we assign to each $u \in T_\Gamma$ a formula I_u called the *pre-interpolant for* u. This is defined by recursion through T_Γ following essentially the same construction as before. Second, by considering the collection of all pre-interpolants it is possible to isolate a uniform interpolant for Γ.

Once we have decided on pre-interpolants for the repeat leaves of T_Γ, the construction of I_u can proceed by the same process as in the quantifier-free case. However, it is convenient to deal with trivial interpolants for instances of GMod as a special case before the more general recursive construction. Recall that if u is the conclusion of GMod we set $I_u = \bot$ if at least one active premise to this rule is valid (as a plain sequent). Lemma 3 confirms that as in the quantifier-free case, this question can be answered by inspection of the interpolation template directly (we omit the details here). With trivial instances of GMod pre-interpolated, we proceed with the recursive construction. Suppose $u \in T_\Gamma$ has not yet been assigned a pre-interpolant. If u sits above an application of GMod which has already been assigned (trivial) interpolant, $I_u = \bot$. In the case u is a repeat I_u is chosen to be a fresh variable symbol X_u uniquely associated to the leaf u. In all other cases, define I_u follows the same construction as before. The 'extra' derivation rules not covered before ($\mathsf{L}\mu$, $\mathsf{L}\nu$, LThin and LReset) are all unary and we define $I_u = I_{u_0}$ where u_0 is the immediate successor to u.

Thus we have an interpolation template T_Γ and a pre-interpolant I_u for each $u \in T_\Gamma$, both relative to a choice of vocabulary $V \subseteq \mathsf{Voc}(\Gamma)$. The pre-interpolant I_u is a quantifier-free formula in variables $X_{u_1}, \ldots, X_{u_k}$ where $u_1, \ldots, u_k$ lists the repeat leaves above u. Let Rep_Γ be the set of repeat leaves of T_Γ and set r to be the root of T_Γ. The uniform interpolant for Γ is defined as the formula represented by the modal equational system

$$I_\Gamma = (I_r, \mathcal{E}_\Gamma) \text{ where } \mathcal{E}_\Gamma = \{X_u =_{p_u} I_{c_u} \mid u \in \mathrm{Rep}_\Gamma\}.$$

It remains to define the priority function $u \mapsto p_u$ on repeats. For this fix an enumeration $\{u_1, \ldots, u_n\}$ of Rep_Γ consistent with the relation $\preceq$ introduced in Sect. 3 and such that if $\mathsf{inv}(u_i) = \mathsf{inv}(u_j)$ for $i < j$ then either u_i is successful or u_j is unsuccessful. Define $p_{u_i} = 2i - 1$ if u_i is successful and $p_{u_j} = 2i$ otherwise.

6 Verifying the Interpolant

In this section we present the argument that I_Γ fulfils the requirements of Theorem 3. Inspection of the definition confirms that the interpolant is in the appropriate language: $\mathsf{Voc}(I_\Gamma) \subseteq V$. We address (ii) and (iii) of the theorem in turn beginning with the former. Due to constraints of space, we omit the technical details. Let T_Γ be a finite interpolation template satisfying Lemma 3.

Proposition 3. $\mathsf{JS} \vdash \Gamma \Rightarrow I_\Gamma$.

Proof. We begin by removing from T_Γ all vertices above the instances of GMod which were assigned trivial pre-interpolant by the construction. Let T^* be this tree. We assume the label of $u \in T^*$ is $\Theta_u : \Pi_u \Rightarrow \varnothing$. The strategy is to show $\vdash \Theta_u : \Pi_u \Rightarrow I_u$ for every $u \in T^*$ where I_u identifies the formula expressed by the equational system $(I_u, \mathcal{E}_\Gamma)$. For vertices of T^* lacking a repeat leaf as a successor, the claim follows the argument given in Sect. 4. Repeats in T^* will be accounted for via the equations $X_l = I_{c_l}$ and the creation of proof cycles.

The claimed JS proof $P \vdash \Gamma \Rightarrow I_\Gamma$ is defined as follows. Begin by identifying the repeat leaves of T^* with their companions to form a graph which we unravel to an infinite tree T^ω. At each sequent in T^ω instantiate the consequent by the assigned pre-interpolant and insert rules from JS between vertices to obtain an infinite JS derivation. Vertices arising from leaves of T^* will have the form $\Theta_l : \Pi_l \Rightarrow X_l$, whereby inserting (unannotated) instances of $\mathsf{R}\mu$ or $\mathsf{R}\nu$ allows them to be connected with their 'companion' sequent $\Theta_l : \Pi_l \Rightarrow I_{c_l}$. Applications of $\mathsf{R}\nu$ inserted in this way are annotated according to the normality condition 4 of Definition 4 and the names propagate through the derivation according to the JS rules (with applications of RReset inserted whenever possible). P is formed by pruning this tree on each path at the first repeated annotated sequent where a variable X_l of the pre-interpolant is principal provided that $l \preceq l'$ for every subsequent variable $X_{l'}$ active on the path.

That the process results in a finite JS derivation is straightforward to verify. To see that P is a proof, consider an arbitrary repeat leaf $l \in P$ and its companion $c_l \in P$. The path $[c_l, l]_P$, when projected to a sequence of vertices of T_Γ, identifies a $\rightsquigarrow$-chain $l_0 \rightsquigarrow l_1 \rightsquigarrow \cdots \rightsquigarrow l_n$ of repeat leaves of T_Γ. Given that repeats in T_Γ are associated unique variables, l and c_l must correspond to the same vertex of T_Γ, meaning $l_n \rightsquigarrow l_0$. Proposition 1 allows us to assume, without loss of generality, that $l_0 \preceq l_i$ for every $i \leq n$ and $p_{l_0} = \min\{p_{l_i} \mid i \leq n\}$. If p_{l_0} is even, meaning l_0 is unsuccessful in T_Γ, then X_{l_0} is of type ν and thus its unfoldings in P introduce names for X_{l_0}. Since $p_{l_0} \leq p_{l_i}$ for all $i \leq n$, the unfoldings of $X_{l_1}, \ldots, X_{l_n}$ preserve the names for X_{l_0} and each unfolding of X_{l_0} removes the names for the other variables, whence we can find a name for X_{l_0} preserved and reset in $[c_l, l]_P$. And if p_{l_0} is odd, then l_0 is successful in T_Γ, so, since $l_0 \preceq l_i$ for every $i \leq n$, the name on the antecedent witnessing the success of l_0 in T_Γ is preserved and reset in $[c_l, l]_P$.

The third clause of Theorem 3 is proved by a similar argument. We transform, using Lemma 3, a given proof $P \vdash \Gamma \Rightarrow \Delta$ into a derivation $P_I \vdash I_\Gamma \Rightarrow \Delta$ by replacing the antecedent of each sequent with the chosen pre-interpolant and observe that any repeat leaf of P_I whose success in P is on account of the variables from Γ is successful by virtue of a variable in I_Γ.

Proposition 4. *Suppose $\Gamma \Rightarrow \Delta$ is a well-named and guarded sequent such that* $\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta) \subseteq V$. *Then* $\vdash \Gamma \Rightarrow \Delta$ *implies* $\vdash I_\Gamma \Rightarrow \Delta$

Proof. Fix Δ satisfying the requirements and a normal JS proof $P \vdash \Gamma \Rightarrow \Delta$ given by Lemma 3. A derivation $P_I \vdash I_\Gamma \Rightarrow \Delta$ is obtained in two steps. First, P is converted into an ill-founded 'derivation' P_I^ω with the desired root sequent by unravelling P, replacing every antecedent by the corresponding pre-interpolant formula given by Lemma 3, correcting the 'left'-rules between vertices and adjusting the control of each vertex accordingly. Observe that the control of a vertex $u \in P_I^\omega$ and the control of the sequent corresponding to u in P are identical when restricted to variables from the consequent, i.e., variables in Δ. They differ, however, in that names of variables from Γ are no longer present in P_I^ω and that names for variables of the pre-interpolant have been inserted. Following the annotation strategy for normal proofs, it follows that every infinite path in P_I^ω contains a repeated annotated sequent which can be marked as a repeat leaf, thus obtaining a finite JS derivation P_I.

It remains to show that P_I is a proof. Let $l \in P_I$ be a repeat leaf with companion $c_l \in P_I$. As in the previous argument, tracing $[c_l, l]_{P_I}$ onto T_Γ yields a $\rightsquigarrow$-cycle $l_0 \rightsquigarrow l_1 \rightsquigarrow \cdots \rightsquigarrow l_n \rightsquigarrow l_0$ of repeat leaves of T_Γ and by Proposition 1 we may assume that $l_0 \preceq l_i$ for every $i \leq n$ and $p_{l_0} = \min\{p_{l_i} \mid i \leq n\}$. If p_{l_0} is odd, then l is a successful repeat because the unfoldings of X_{l_0} remove the names for any other variable X_{l_i} and introduce a new name for X_{l_0}, whence a name for X_{l_0} is preserved and reset in $[c_l, l]_{P_I}$. Therefore, suppose p_{l_0} is even, meaning that l_0 is not a successful leaf of T_Γ. But given that P is a proof, it follows that l is successful due to a name for a variable in Δ being preserved and reset on the path $[c_l, l]_{P_I}$.

7 Conclusion

We introduced the notion of an *interpolation template* for formulas of the modal μ-calculus and showed that these describe uniform interpolants. Interpolation templates are finite (cyclic) derivation trees in a sequent calculus based on the Jungteerapanich–Stirling annotated proof system of [28,41]. Uniform interpolants arise from encoding the structure of interpolation templates as formulas of the μ-calculus.

Interpolation templates can be given for a wide range of modal and temporal logics simply by embedding them into the modal μ-calculus. This holds even for logics that lack the uniform interpolation property, such as propositional dynamic logic PDL and the alternation-free fragment of the μ-calculus. Although uniform interpolants can still be constructed by the method given, these will be formulas of the μ-calculus that can not be expressed within the syntax of the logic in question. Nevertheless, it would be informative to investigate the interpolants that arise from templates for these logics as a way of shedding light on the complexity and expressivity of fragments of the μ-calculus. A natural question is whether uniform interpolants generated from, say, alternation-free formulas are of bounded complexity or exhaust the quantifier hierarchy of the modal μ-calculus.

We also expect our approach can be directly applied to logics without a detour via modal μ-calculus. A key requirement is the existence of a sound and complete cyclic sequent calculus from which one can define interpolation templates. While there are many examples of cyclic sequent calculi[4] not all are directly amenable to our methods. Our notion of interpolation templates presupposes an analytic calculus satisfying an appropriate sub-formula property. On the other hand, constructing interpolants from templates assumes a highly expressive logic, namely the definability of fixed points for the formulas arising from interpolation templates. From these perspectives, two logics in particular stand out as interesting candidates for investigation: Gödel–Löb provability logic GL and bisimulation quantifier logic BQL. The former logic admits an analytic cyclic sequent calculus which has been utilised to prove Craig interpolation for GL [38]. Bisimulation quantifier logic is the extension of propositional dynamic logic PDL with bisimulation quantifiers [15]. Although BQL is expressively equivalent to μ-calculus [14], the reduction is highly complex and a direct treatment of BQL will be an interesting contribution of cyclic proof theory to modal logic.

Uniform interpolation is intimately connected with quantification [14,35,43]. For modal logics the appropriate notion of quantification is (propositional) bisimulation quantifiers. In the present framework where propositional constants are replaced by modal actions, it is natural to consider quantifiers ranging over modal actions. Thus the formula $\exists a\varphi$ expresses that $\varphi[b/a]$, the result of substituting the modal action b for a in φ, holds for some $b \in \mathsf{Act}$, and similarly for the universal quantifier $\forall a\varphi$. To basic modal logic we may add the logical axiom $\vdash \varphi[b/a] \to \exists a\varphi$ and the rule 'from $\vdash \varphi \to \psi$ infer $\vdash \exists a\varphi \to \psi$,' where ψ is any formula in which a does not occur free (using the usual definition of free occurrences of variables). The universal quantifier is treated symmetrically. Having access to action quantifiers, the formula $\exists a\varphi$ is nothing more than a uniform interpolant for φ with respect to the vocabulary $\mathsf{Voc}(\varphi) \setminus \{a\}$. As such, our construction of uniform interpolants provides a new proof of the definability of bisimulation quantifiers in the modal μ-calculus, a result first established in [14] via automata theoretic methods.

Another direction of research arises in the area of description logics and knowledge representation, where uniform interpolation plays an important tool in reducing the search space for querying ontologies [6,12,33]. In the case of acyclic TBoxes uniform interpolants often exist but there seems to be no uniform approach for dealing with cyclic constraints [5]. We have left aside complexity considerations in the present study but it is an important factor for the methodology to have practical applications for tableaux-based algorithms. At this stage, our expectation is that the complexity may well not be favourable for modal μ-calculus. Nevertheless we hope the approach can contribute to database-related reasoning problems in the context of less expressive logics.

[4] A non-exhaustive list of cyclic proofs systems include: *first-order logic with inductive definitions* [8,9,11], *arithmetic* [7,17,39], *linear logic* [3,4,20], *modal and dynamic logics* [1,22,23,28,30,38,40,41,44], *program semantics* [37], *automated theorem proving* [10,36,42], *higher-order logic* [31] and *algebras and lattices* [18,19,32].

References

1. Afshari, B., Leigh, G.E.: Cut-free completeness for modal mu-calculus. In: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, 20–23 June 2017, pp. 1–12. IEEE Computer Society (2017)
2. Afshari, B., Leigh, G.E.: Lyndon interpolation for modal mu-calculus. In: Post-Proceedings of TbiLLC 2019. (to appear)
3. Baelde, D., Doumane, A., Kuperberg, D., Saurin, A.: Bouncing threads for infinitary and circular proofs (2020). https://arxiv.org/abs/2005.08257
4. Baelde, D., Doumane, A., Saurin, A.: Infinitary proof theory: the multiplicative additive case. In: Talbot, J., Regnier, L. (eds.) 25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29–September 1 2016, Marseille, France. LIPIcs, vol. 62, pp. 42:1–42:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
5. Benedikt, M.: How can reasoners simplify database querying (and why haven't they done it yet)? In: den Bussche, J.V., Arenas, M. (eds.) Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, 10–15 June 2018, pp. 1–15. ACM (2018)
6. Benedikt, M., ten Cate, B., Vanden Boom, M.: Interpolation with decidable fixpoint logics. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, 6–10 July 2015, pp. 378–389. IEEE Computer Society (2015)
7. Berardi, S., Tatsuta, M.: Equivalence of inductive definitions and cyclic proofs under arithmetic. In: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, 20–23 June 2017, pp. 1–12. IEEE Computer Society (2017)
8. Berardi, S., Tatsuta, M.: Classical system of Martin-Löf's inductive definitions is not equivalent to cyclic proofs. Log. Methods Comput. Sci. **15**(3), 1:1–1:39 (2019)
9. Brotherston, J.: Cyclic proofs for first-order logic with inductive definitions. In: Beckert, B. (ed.) TABLEAUX 2005. LNCS (LNAI), vol. 3702, pp. 78–92. Springer, Heidelberg (2005). https://doi.org/10.1007/11554554_8
10. Brotherston, J., Gorogiannis, N., Petersen, R.L.: A generic cyclic theorem prover. In: Jhala, R., Igarashi, A. (eds.) APLAS 2012. LNCS, vol. 7705, pp. 350–367. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35182-2_25
11. Brotherston, J., Simpson, A.: Sequent calculi for induction and infinite descent. J. Log. Comput. **21**(6), 1177–1216 (2011)
12. ten Cate, B., Franconi, E., Seylan, I.: Beth definability in expressive description logics. J. Artif. Intell. Res. **48**, 347–414 (2013)
13. D'Agostino, G.: Interpolation in non-classical logics. Synthese **164**(3), 421–435 (2008)
14. D'Agostino, G., Hollenberg, M.: Logical questions concerning the μ-calculus. J. Symb. Log. **65**(1), 310–332 (2000)
15. D'Agostino, G., Lenzi, G.: An axiomatization of bisimulation quantifiers via the mu-calculus. Theor. Comput. Sci. **338**(1–3), 64–95 (2005)
16. D'Agostino, G., Lenzi, G.: Bisimulation quantifiers and uniform interpolation for guarded first order logic. Theor. Comput. Sci. **563**, 75–85 (2015)
17. Das, A.: On the logical complexity of cyclic arithmetic. Log. Methods Comput. Sci. **16**(1), 10:1–10:25 (2020)
18. Das, A., Pous, D.: A cut-free cyclic proof system for Kleene algebra. In: Schmidt, R.A., Nalon, C. (eds.) TABLEAUX 2017. LNCS (LNAI), vol. 10501, pp. 261–277. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66902-1_16

19. Das, A., Pous, D.: Non-wellfounded proof theory for (kleene+action)(algebras+lattices). In: Ghica, D.R., Jung, A. (eds.) 27th EACSL Annual Conference on Computer Science Logic, CSL 2018, 4–7 September 2018, Birmingham, UK. LIPIcs, vol. 119, pp. 19:1–19:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)
20. De, A., Saurin, A.: Infinets: the parallel syntax for non-wellfounded proof-theory. In: Cerrito, S., Popescu, A. (eds.) TABLEAUX 2019. LNCS (LNAI), vol. 11714, pp. 297–316. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29026-9_17
21. Demri, S., Goranko, V., Lange, M.: Temporal Logics in Computer Science: Finite-State Systems, vol. 58. Cambridge University Press, Cambridge (2016)
22. Docherty, S., Rowe, R.N.S.: A non-wellfounded, labelled proof system for propositional dynamic logic. In: Cerrito, S., Popescu, A. (eds.) TABLEAUX 2019. LNCS (LNAI), vol. 11714, pp. 335–352. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29026-9_19
23. Enqvist, S., Hansen, H.H., Kupke, C., Marti, J., Venema, Y.: Completeness for game logic. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, 24–27 June 2019, pp. 1–13. IEEE (2019)
24. Gabbay, D.M., Maksimova, L.: Interpolation and Definability. Modal and Intuitionistic Logic. Oxford University Press, Oxford (2005) Logic. Oxford University Press, Oxford (2005)
25. Iemhoff, R.: Uniform interpolation and sequent calculi in modal logic. Arch. Math. Log. **58**(1–2), 155–181 (2019)
26. Iemhoff, R.: Uniform interpolation and the existence of sequent calculi. Ann. Pure Appl. Log. **170**(11), 102711 (2019)
27. Janin, D., Walukiewicz, I.: Automata for the modal μ-calculus and related results. In: Wiedermann, J., Hájek, P. (eds.) MFCS 1995. LNCS, vol. 969, pp. 552–562. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60246-1_160
28. Jungteerapanich, N.: A tableau system for the modal μ-calculus. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 220–234. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02716-1_17
29. Jungteerapanich, N.: Tableau systems for the modal μ-calculus. Ph.D. thesis, University of Edinburgh (2010)
30. Kokkinis, I., Studer, T.: Cyclic proofs for linear temporal logic. In: Probst, D., Schuster, P. (eds.) Concepts of Proof in Mathematics, Philosophy, and Computer Science, Ontos Mathematical Logic, vol. 6, pp. 171–192. De Gruyter (2016)
31. Kori, M., Tsukada, T., Kobayashi, N.: A cyclic proof system for HFL_N. In: Baier, C., Goubault-Larrecq, J. (eds.) 29th EACSL Annual Conference on Computer Science Logic, CSL 2021, 25–28 January 2021, Ljubljana, Slovenia (Virtual Conference). LIPIcs, vol. 183, pp. 29:1–29:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
32. Kuznetsov, S.: Half a way towards circular proofs for Kleene lattices (2019). circularity in Syntax and Semantics
33. Lutz, C., Wolter, F.: Foundations for uniform interpolation and forgetting in expressive description logics. In: Walsh, T. (ed.) IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 16–22 July 2011, pp. 989–995. IJCAI/AAAI (2011)
34. Niwinski, D., Walukiewicz, I.: Games for the mu-calculus. Theor. Comput. Sci. **163**(1 & 2), 99–116 (1996)
35. Pitts, A.M.: On an interpretation of second order quantification in first order intuitionistic propositional logic. J. Symb. Log. **57**(1), 33–52 (1992)

36. Rowe, R.N.S., Brotherston, J.: Realizability in cyclic proof: extracting ordering information for infinite descent. In: Schmidt, R.A., Nalon, C. (eds.) TABLEAUX 2017. LNCS (LNAI), vol. 10501, pp. 295–310. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66902-1_18
37. Santocanale, L.: A calculus of circular proofs and its categorical semantics. In: Nielsen, M., Engberg, U. (eds.) FoSSaCS 2002. LNCS, vol. 2303, pp. 357–371. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45931-6_25
38. Shamkanov, D.: Circular proofs for the Gödel-Löb provability logic. Math. Notes **96**, 575–585 (2014)
39. Simpson, A.: Cyclic arithmetic is equivalent to Peano Arithmetic. In: Esparza, J., Murawski, A.S. (eds.) FoSSaCS 2017. LNCS, vol. 10203, pp. 283–300. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54458-7_17
40. Sprenger, C., Dam, M.: On the structure of inductive reasoning: circular and tree-shaped proofs in the μ-calculus. In: Gordon, A.D. (ed.) FoSSaCS 2003. LNCS, vol. 2620, pp. 425–440. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36576-1_27
41. Stirling, C.: A tableau proof system with names for modal mu-calculus. In: Voronkov, A., Korovina, M.V. (eds.) HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday, EPiC Series in Computing, vol. 42, pp. 306–318. EasyChair (2014)
42. Tellez, G., Brotherston, J.: Automatically verifying temporal properties of pointer programs with cyclic proof. J. Autom. Reason. **64**(3), 555–578 (2020)
43. Visser, A.: Bisimulations, model descriptions and propositional quantifiers. Logic Group Preprint Series no. 161, Utrecht (1996)
44. Visser, A.: Cyclic Henkin logic (2021). https://arxiv.org/abs/2101.11462v1