Available online at www.sciencedirect.com
ScienceDirect
Journal of Applied Logic 4 (2006) 256–278
JOURNAL OF APPLIED LOGIC
www.elsevier.com/locate/jal

On modal μ-calculus with explicit interpolants

G. D'Agostino [a,*], G. Lenzi [b]

[a] *Department of Mathematics and Computer Science, University of Udine, Viale delle Scienze 206, 33100 Udine, Italy*
[b] *Department of Mathematics, University of Pisa, Via Buonarroti 2, 56127 Pisa, Italy*

Available online 21 July 2005

Abstract

This paper deals with the extension of Kozen's μ-calculus with the so-called "existential bisimulation quantifier". By using this quantifier one can express the uniform interpolant of any formula of the μ-calculus. In this work we provide an explicit form for the uniform interpolant of a disjunctive formula and see that it belongs to the same level of the fixpoint alternation hierarchy of the μ-calculus than the original formula. We show that this result cannot be generalized to the whole logic, because the closure of the third level of the hierarchy under the existential bisimulation quantifier is the whole μ-calculus. However, we prove that the first two levels of the hierarchy are closed. We also provide the μ-logic extended with the bisimulation quantifier with a complete calculus.

Keywords: μ-calculus; Bisimulation quantifier; Uniform interpolation

1. Introduction

Bisimulation quantifiers were first considered in [4] and [12] as a tool for proving uniform interpolation for modal logic, and in [2] to show uniform interpolation for the modal μ-calculus. Given a formula ϕ, the language of ϕ is the set of all propositional constants appearing in the formula. The uniform interpolant of a formula ϕ with respect to a sublanguage L' of the language of ϕ is a formula ψ in the language L' which *behaves* like ϕ when L' is concerned, in the sense that ψ has the same L'-logical consequences as ϕ. A logic

* Corresponding author.
E-mail addresses: dagostin@dimi.uniud.it (G. D'Agostino), lenzi@mail.dm.unipi.it (G. Lenzi).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2005.06.008

that contains all uniform interpolants of its formulas is said to enjoy uniform interpolation. Since uniform interpolation implies Craig interpolation, which in turn implies properties as the Beth one, a logic having uniform interpolation has a good interplay between syntactical and semantical behaviour. Moreover, uniform interpolation allows *modularization*: if we are interested only in L'-consequences of ϕ then we may consider the (hopefully simpler) uniform interpolant ψ instead of ϕ, and *derive* all its L'-consequences (which are the same as the L'-consequences of ϕ) in an appropriate calculus: this would be like a module for this subtask. Notice that modularization is not possible if only Craig interpolation is present.

The relation between uniform interpolation and the existential bisimulation quantifier (which we denote by $\widetilde{\exists}P$) works as follows. The semantics of $\widetilde{\exists}P$ tells us that $\widetilde{\exists}P\phi(P)$ is true in a model if we can find a subset P satisfying $\phi(P)$ not only within the model, but also in any other model which is bisimilar to the given one. One can prove that any logic which is invariant under bisimulation and closed under the existential bisimulation quantifier (that is: given ϕ in the logic, there exists a formula ψ in the logic with the same semantics as $\widetilde{\exists}P\phi$) enjoys uniform interpolation: the uniform interpolant of a formula $\phi(P)$ with respect to the language $L(\phi) \setminus \{P\}$ is simply $\widetilde{\exists}P\phi(P)$. This is the way uniform interpolation is proved for the μ-calculus in [2]. However, this result does not give an explicit form for uniform interpolants in the original logic: given a μ-formula ϕ and a sublanguage L' of $L(\phi)$, we know that a uniform interpolant of ϕ with respect to L' exists in the μ-calculus, but we don't know how to construct it from ϕ. Hence, we cannot *use* the uniform interpolant to derive all L'-consequences of ϕ.

In this paper we study the relations between bisimulation quantifiers and the μ-calculus more closely. First of all we restrict our attention to disjunctive μ-formulas. The disjunctive formulas (see e.g. [7,8]) form an important subset of the μ-calculus, because any μ-formula is equivalent to a disjunctive one, with the advantage that disjunctive formulas behave more nicely than in the general case: e.g. the problem of satisfiability for disjunctive formulas is linear. We shall see that this docility of the disjunctive formulas is confirmed when the existential bisimulation quantifier (or uniform interpolants) is concerned: if ϕ is disjunctive then $\widetilde{\exists}P\phi$ is equivalent to the formula obtained from ϕ by the simultaneous substitution of P and $\neg P$ by $\top$. This result allows us to calculate the uniform interpolant of a disjunctive formula explicitly, and to show that the uniform interpolant is not more complicate than the original formula: a good measure of the complexity of a μ-calculus formula is given by the fixpoint alternation hierarchy of the μ-calculus (proved to be strict in [1]) and from the result above it is clear that the uniform interpolant of a disjunctive formula ϕ belongs to the same level as ϕ.

We will then consider the general problem: is it true that all levels of the μ-calculus are closed under the existential bisimulation quantifier, or, equivalently, is it true that the uniform interpolant of a formula in a certain level of the hierarchy belongs to the same level? This is not a mere curiosity, because the best model checking algorithm known so far for μ-calculus formulas depends on the fixpoint alternation level of the formula: the lower the level, the easier it is to check whether the formula is true in a finite model (see e.g. [9]). For this reason it is sometime preferable to consider not the whole μ-calculus but only formulas up to a certain level (in practice, all temporal logics used in applications can be embedded into the low levels of the hierarchy). Then the question of whether the levels of the hierarchy are closed or not under the existential bisimulation quantifier becomes relevant, because an affirmative answer would give a certification of the possibility of modularizations for the level under consideration. In particular, it would be good to know whether the low levels are closed. This closure property is already known for the 0-level of the fixpoint hierarchy, that is, it is already known that modal logic is closed under the existential bisimulation quantifier [4,12]. In this paper we generalize this property to the levels 1 and 2 of the hierarchy. On the other hand, we see that level 3 is not closed, and more than this, that any μ-formula is obtained by considering the uniform interpolant of a formula of the third level.

Since the third level is not closed under the existential bisimulation quantifier it follows immediately that no simple rule such as $\widetilde{\exists} P\phi \leftrightarrow \phi[P/\top, \neg P/\top]$, which is valid for disjunctive formulas, can possibly hold for the whole μ-calculus. However, although we are not able to simplify so easily the existential bisimulation quantifier in the general case, we can still try to understand more precisely how this quantifier behaves w.r.t. the connectives and the operators of the μ-calculus. One way to do so is to enrich the original μ-language with an existential bisimulation quantifier $\widetilde{\exists} P$ with the appropriate semantics and provide this extended logic with a complete calculus. We shall see that to derive all validities in the extended logic we need some standard principles allowing introduction and elimination of the bisimulation quantifier, plus some natural principles of commutativity between the existential bisimulation quantifier and the operators of the μ-calculus.

Notice that, at least in principle, modularization for the μ-calculus could be obtained from modularization of the extended logic: to derive all L'-logical consequences of a μ-sentence ϕ we may go to the extended logic, write the uniform interpolant using bisimulation quantifiers and use the extended calculus to derive all consequences of it. The new logic, denoted by $\widetilde{\mu}$, is not more powerful then the original one, but in the extended language we gain the possibility to express uniform interpolants explicitly, and the complete calculus allows also to *work* with them.

The paper is organized as follows. In Section 2 we introduce the μ-calculus, give the definition of bisimulation quantifiers, and summarize the results already known about these quantifiers in the μ-calculus context. In Section 3 we calculate the uniform interpolant for disjunctive formulas. In Section 4 we prove the results concerning the fixpoint alternation hierarchy and the bisimulation quantifier. In the last section we find a complete calculus for the μ-logic extended with the existential bisimulation quantifier.

2. Notation and preliminaries

2.1. The μ-calculus

First of all, we recall the definition of the extension of modal logic known as the modal μ-calculus.

Definition 2.1. The *μ-calculus* is defined as the least set which contains a set of propositional constants *Prop*, a set of variables *Var*, and satisfies: if $\phi, \psi \in \mu$ then $\neg\phi$, $\phi \vee \psi$, $\Diamond\phi$

belong to μ; if $X \in \mathit{Var}$ and X occurs just positively in ϕ (that is: under an even number of negations) then $\mu X \phi$ belongs to μ.

Remark 2.2. The μ-calculus is usually defined as the extension of *multi*-modal logic, where a set of actions A and a corresponding set of operators $\Diamond_a$ are considered. For simplicity we restrict ourselves to the case in which only one action is present, although our results can easily be extended to the general case (by generalizing the covers-syntax as it is done in [7]).

The derived operators $\phi \wedge \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$, $\Box\phi$, and $\nu X.\phi$ are defined as usual. The variable X is said to be bound in $\mu X.\phi$, $\nu X.\phi$. Free variables in a formula and sentences are defined as usual. If ϕ is a formula, then $L(\phi)$ is defined as the set of propositional constants and free variables occurring in ϕ. We call ϕ a *modal formula* if it is constructed without using the fixpoint operators.

A μ-calculus formula is interpreted in structures for the language $\{r, R\} \cup \mathit{Prop}$; these are Kripke-structures, i.e., tuples of the form

$$M = (D^M, r^M, R^M, P_1^M, \ldots),$$

where the non-empty set D^M is the domain of M, r^M is an element of this domain, R^M is a binary relation on D^M, and P^M is a subset of D^M, for any $P \in \mathit{Prop}$. Given a structure M and a valuation $V : \mathit{Var} \rightarrow \wp(D^M)$, a μ-formula is interpreted in M as a subset $\|\phi\|_V$ of D^M, defined as follows:

$$\begin{aligned}
&\|P\|_V := P^M,\\
&\|X\|_V := V(X),\\
&\|\neg\phi\|_V := D^M \setminus \|\phi\|_V,\\
&\|\phi \vee \psi\|_V := \|\phi\|_V \cup \|\psi\|_V,\\
&\|\Diamond\phi\|_V := \big\{s \in D^M \mid \|\phi\|_V \cap \{t\colon sR^M t\} \neq \emptyset\big\},\\
&\|\mu X.\phi\|_V := \bigcap\big\{S \subseteq D^M \mid \|\phi\|_{V[X:=S]} \subseteq S\big\},
\end{aligned}$$

where $V[X := S]$ is equal to the valuation function V except that S is assigned to X. Note that $\|\mu X.\phi\|_V$ is the least fixpoint of the monotone operator $S \mapsto \|\phi\|_{V[X:=S]}$.

In the following, we denote $s \in \|\phi\|_V$ by $(M, s, V) \models \phi$ and we may leave out the valuation, if ϕ is a sentence. $(M, V) \models \phi$ is used to denote $(M, r^M, V) \models \phi$.

$\Gamma \models \phi$ denotes logical consequence: if $(M, V) \models \Gamma$ then $(M, V) \models \phi$ for every model M and for every valuation V.

An alternative syntax for the μ-calculus is obtained by substituting the $\Diamond$ operator with a set of *cover operators*, one for each natural n. For $n \geqslant 1$ these operators are defined as follows: if $\phi_1, \ldots, \phi_n$ are formulas, then

$$\mathit{Cover}(\phi_1, \ldots, \phi_n)$$

is a formula. We also allow the constant operator $\mathit{Cover}(\emptyset)$. The cover operators are interpreted in a Kripke structure M as follows: $\mathit{Cover}(\emptyset)$ is true in M if and only if the root of M

does not have any successor, while $Cover(\phi_1, \dots, \phi_n)$ is true in M if and only if the successors of the root are *covered* by $\phi_1, \dots, \phi_n$. More formally, $(M, s, V) \models Cover(\phi_1, \dots, \phi_n)$ if and only if:

(1) for every $i = 1, \dots, n$ there exists t with $(s, t) \in R^M$ and $(M, t, V) \models \phi_i$;
(2) for every t with $(s, t) \in R^M$ there exists $i \in \{1, \dots, n\}$ with $(M, t, V) \models \phi_i$.

We call this syntax the *covers-syntax* to distinguish it form the original $\Diamond$-syntax.
Since $Cover(\phi_1, \dots, \phi_n)$ is equivalent to

$$\Diamond(\phi_1) \wedge \cdots \wedge \Diamond(\phi_n) \wedge \Box(\phi_1 \vee \cdots \vee \phi_n),$$

cover operators are definable in the $\Diamond$ syntax. Conversely,

$$\Diamond\phi \Leftrightarrow Cover(\phi, \top).$$

Hence, the μ-calculus obtained from the covers-syntax is equivalent to the familiar μ-calculus constructed using the $\Diamond$-syntax. In this paper we use the covers-syntax because, as we shall see, cover operators behave nicely with respect to the existential bisimulation quantifier.

We now introduce an important class of μ-formulas.

Definition 2.3. The class of *disjunctive μ-formulas* is the least class containing $\top$, $\bot$, and non-contradictory conjunction of literals which is closed under:

(1) disjunctions;
(2) special conjunctions: if $\phi_1, \dots, \phi_n$ are formulas in the class and σ is a non-contradictory conjunction of literals, then $\sigma \wedge Cover(\phi_1, \dots, \phi_n)$ is in the class;
(3) fixpoint operators: if ϕ is disjunctive, ϕ does not contain $X \wedge \gamma$ as a subformula for any formula γ, and X is positive in ϕ, then $\mu X\phi$, $\nu X\phi$ are in the class.

The disjunctive formulas are representative of the whole μ-calculus:

Theorem 2.4. [7] *Any μ-calculus formula is equivalent to a disjunctive one.*

The same is true (but the proof is easier) for the class of modal formulas with respect to disjunctive modal formulas (which are defined as in Definition 2.3 but without closing for fixpoint operators).

2.2. Bisimulation quantifiers and uniform interpolation

To introduce bisimulation quantifiers we first need the notion of *bisimulation*.

Definition 2.5. Let M, N be structures with D^M, D^N as respective domains. Let $Prop' \subseteq Prop$. A relation $Z \subseteq D^M \times D^N$ is a *$Prop'$-bisimulation* between M and N if:

(1) $r^M Z r^N$;

(2) if wZv then $w \in P^M$ iff $v \in P^N$, for every $P \in Prop'$;
(3) if wZv and $wR^M w'$, then there exists a v' such that $vR^N v'$ and $w'Zv'$;
(4) if wZv and $vR^N v'$, then there exists a w' such that $wR^M w'$ and $w'Zv'$.

Two structures M, N are *Prop′-bisimilar* (notation: $M \sim_{Prop'} N$) if there exists a *Prop′*-bisimulation between them (but we write $M \sim N$ if $Prop' = Prop$).

If $Var' \subseteq Var$ and V_1, V_2 are valuations of the variables in Var' in the structures M, N, respectively, a bisimulation Z between (M, V_1) and (N, V_2) is a bisimulation between M and N such that if wZv then $w \in V_1(X)$ iff $v \in V_2(X)$, for every $X \in Var'$. We use the notation $(M, V_1) \sim (N, V_2)$ accordingly.

An *existential bisimulation quantifier* is defined as a classical monadic second order quantifier, except that we look for a subset satisfying a certain property not only within the model, but also in any other model which is bisimilar to the given one:

Definition 2.6 *(Existential bisimulation quantifier).* We enrich the μ-grammar with a propositional quantifier $\widetilde{\exists} P\phi$, for any $P \in Prop$, whose semantics is defined as follows. If M is a structure and V is a valuation of the free variables of ϕ, then (M, V) satisfies the formula $\widetilde{\exists} P\phi$ iff:

(1) there exists a structure N and a valuation V' of the set $Free(\phi)$ of the free variables of ϕ with $(M, V) \sim_{Prop\setminus\{P\}} (N, V')$;
(2) there exists a subset $P \subseteq D^N$ such that $(N, P, V') \models \phi$, where (N, P, V') stands for the structure which is like (N, V') except for the propositional constant P that receives now a new interpretation (denoted again by P).

The set of formulas obtained using the extended grammar is denoted by $\widetilde{\mu}$.

Example 1. The sentence $\widetilde{\exists} P(\Diamond P \wedge \Diamond\neg P)$ is true in M iff the root has at least one successor. In fact, in a model, there exists a P such that $\Diamond P \wedge \Diamond\neg P$ holds if and only if the root has at least two successors (just put one successor of the root in P and one out of P); and a model M is bisimilar to one where the root has two successors if and only if, in M, the root has at least one successor.

Example 2. As we shall see (Theorem 2.9) the logic $\widetilde{\mu}$ has the same expressive power as the μ-calculus. However, the bisimulation quantifier can help to express a certain property in a way that is closer to natural language than the formulation of the property in the μ-calculus. For example, suppose we are interested in the existence of an infinite path starting from the root where P holds infinitely often. If we think of X as the path we are looking for, it is not difficult to see that this property is expressed by the monadic second order formula $\exists X F(X, P)$, where $\exists$ is the standard monadic second order existential quantifier,

$$F(X, P) = X \wedge \Box^*\big(X \to \Diamond^+(X \wedge P)\big),$$

the operator $\Box^*\phi$ is a shorthand for the μ-calculus formula $\nu X(\phi \wedge \Box X)$ and means that ϕ is true in every descendant of the current point, and $\Diamond^+\phi$ is a shorthand for $\mu X\Diamond(\phi \vee X)$ and means that ϕ is true in some proper descendant of the current point.

Since the existence of such a path is a property which is invariant under bisimulation, one can easily see the same property can be expressed in $\widetilde{\mu}$ by $\widetilde{\exists} X F(X, P)$. Compare now this formula with the μ sentence expressing the same property:

$$\nu X \mu Y\big((P \wedge \Diamond X) \vee \Diamond Y\big).$$

The μ formula is shorter, but it is not so easy for a non-specialist to find it or even just to check that the formula works.

Next, we consider the notion of uniform interpolant.

Definition 2.7. Given a μ-sentence ϕ and a language $L' \subseteq L(\phi)$, the *uniform interpolant* of ϕ with respect to L' is a μ-sentence θ such that:

(1) $\phi \models \theta$;
(2) whenever $\phi \models \psi$ and $L(\phi) \cap L(\psi) \subseteq L'$ then $\theta \models \psi$;
(3) $L(\theta) \subseteq L'$.

In [2] it is proved that the μ-calculus enjoys uniform interpolation, in the sense that for any μ-sentence ϕ and $L' \subseteq L(\phi)$ there exists a μ-sentence θ satisfying the above properties. Notice that uniform interpolation is stronger than Craig interpolation, which states that for any two formulas ϕ, ψ with $\phi \models \psi$ there exists a formula θ in the common language (called the Craig interpolant of ϕ, ψ) with $\phi \models \theta$ and $\theta \models \psi$: if the logic enjoys uniform interpolation then the Craig interpolant of ϕ, ψ does not depend on ψ but only on the common language: it is simply the uniform interpolant of ϕ relative to $L' = L(\phi) \cap L(\psi)$. This explains why we call this formula a *uniform* interpolant: no information is needed about the formula ψ except which non-logical symbols it has in common with ϕ.

The proof of uniform interpolation for the μ-calculus is obtained by considering first the case of $L' = L(\phi) \setminus \{P\}$ and then iterating the construction. In the case of $L' = L(\phi) \setminus \{P\}$ it can be proved that any μ-formula which is equivalent (in the $\widetilde{\mu}$-semantics) to $\widetilde{\exists} P\phi$ is a uniform interpolant of ϕ with respect to the language L':

Theorem 2.8. [2] *If ϕ is a μ-formula and θ is such that*

$$\models \theta \leftrightarrow \widetilde{\exists} P\phi,$$

then θ is a uniform interpolant of ϕ w.r.t. $L(\phi) \setminus \{P\}$.

Then, uniform interpolation for the μ-calculus follows if one proves that the μ-calculus is closed under the existential bisimulation quantifier:

Theorem 2.9. [2] *If ϕ is a μ-formula, there exists a μ-formula θ with $L(\theta) \subseteq L(\phi) \setminus \{P\}$ such that*

$$\models \theta \leftrightarrow \widetilde{\exists} P\phi.$$

3. Explicit uniform interpolants for disjunctive formulas

In this section we prove our first result concerning the existential bisimulation quantifier and the μ-calculus. We restrict our attention to the class of disjunctive formulas and prove that if $\phi(P, \neg P)$ belongs to this class, then its uniform interpolant $\widetilde{\exists} P\phi$ is equivalent to $\phi[P/\top, \neg P/\top]$. This can be proved using the correspondence between disjunctive formulas and nondeterministic automata as defined in [7]. We first recall the definition of this kind of automata.

Definition 3.1. A nondeterministic parity modal automaton is a tuple

$$A = \langle Q_A, \Sigma_A, q_{0,A}, \delta_A, \Omega_A\rangle,$$

such that:

- Q_A is a finite set of states;
- Σ_A is a finite alphabet (the powerset of a finite subset *Prop*$'$ of *Prop*);
- $q_{0,A} \in Q_A$ is the initial state;
- Ω_A is a function from Q_A to the natural numbers;
- δ_A is a function which associates to every $q \in Q_A$ and $\sigma \in \Sigma_A$ a set $\{D_1, \dots, D_n\}$ of subsets of Q_A.

We omit the A-subscript when possible.

The acceptance condition of nondeterministic modal automata is usually defined by a game, but if we restrict to ω-expanded trees we can also describe acceptance in terms of labellings (a proof of this result can be found in [5, Lemmas 3.4.3 and 3.4.4]. Here we adopt the labeling condition as a definition of acceptance directly.

Recall that *an ω-expanded tree* is a tree T in which for every $s, s' \in T$ if $s' \in Succ(s)$ then there are at least ω distinct successors of s which are bisimilar to s'. A Σ-valued tree is a tree in which nodes are labeled by elements in Σ (we denote the label of s by $T(s)$).

Definition 3.2. Let $A = \langle Q, \Sigma, q_0, \delta, \Omega\rangle$ be a nondeterministic modal automaton and T a Σ-valued tree. A total function $l : T \to Q$ is an A-*labeling* (also called an *accepting run* for A over T) if:

(1) $l(r^T) = q_0$;
(2) If $l(s) = q$ then $\{q' \in Q\colon \exists t \in Succ(s), l(t) = q'\}$ belongs to $\delta(q, T(s))$;
(3) For any infinite T-path $s_0 = r^M, s_1, \dots$:

$$\min\bigl\{\Omega(q) \mid \text{there are infinitely many } s_i \text{ in the path s.t. } l(s_i) = q\bigr\}$$

is even.

Definition 3.3. Given a nondeterministic modal automaton A, we say that an ω-expanded tree T is *accepted* by A iff it has an A-labeling.

Nondeterministic modal automata are automata-theoretic counterparts of disjunctive μ-formulas, in the sense that to any such formula it corresponds a nondeterministic automaton accepting the same ω-expanded trees, and vice versa. To describe the explicit form of the uniform interpolant of a disjunctive formula we shall need a direct translation from automata to disjunctive formulas, as described in [6]. Let us describe how this translation is achieved. Given an automaton, we first put it in *tree normal form*:

Definition 3.4. The automaton $A = \langle Q, \Sigma, q_0, \delta, \Omega\rangle$ is said to be in tree normal form if there is an order relation $\leqslant_A$ between the states of A which satisfies the following properties:

(1) q_0 is the minimum state of Q with respect to $\leqslant_A$;
(2) if the states q_1, q_2, q_3 are such that $q_2, q_3 \leqslant_A q_1$, then q_2, q_3 are $\leqslant_A$-comparable;
(3) if $q_2 \in D \in \delta(q_1, \sigma)$, then either q_2 is an immediate $\leqslant_A$ successor of q_1 or $q_2 \leqslant_A q_1$ and $\Omega(q_2) \leqslant \Omega(q)$ for all q such that $q_2 \leqslant_A q \leqslant_A q_1$.

One can prove that for any automaton A there exists an equivalent automaton in tree normal form (see [6]). If A is in tree normal form and q is an A-state, we define the disjunctive formula $\phi_{A,q}$ by induction on the tree like form of A, from the leaves towards the root:

$$\phi_{A,q} = (\sigma_q X_q) \bigvee_{\sigma\in\Sigma_A} \bigvee_{D\in\delta(q,\sigma)} \hat{\sigma} \wedge Cover\bigl(\{\beta_{q,q'}\colon q' \in D\}\bigr),$$

where:

(1) $\hat{\sigma} = \bigwedge_{P\in\sigma} P \wedge \bigwedge_{P\notin\sigma} \neg P$;
(2) $\sigma_q = \nu$ if $\Omega(q)$ is even, $\sigma_q = \mu$ if $\Omega(q)$ is odd;
(3) $\beta_{q,q'} = X_{q'}$ if $q' \leqslant_A q$, $\beta_{q,q'} = \phi_{A,q'}$, otherwise.

Let $\phi_A = \phi_{A,q_0}$: then it is possible to prove (see [6]) that ϕ_A is equivalent to A.

Regarding bisimulation quantifiers, in [2] it is proved that they correspond to projections in the automata settings. If P is a proposition and $A = \langle Q, \Sigma, q_0, \delta, \Omega\rangle$ is an automaton, we define the automaton $\widetilde{\exists} PA = \langle Q, \Sigma', q_0, \delta', \Omega\rangle$ as follows:

(1) $\Sigma' = \{\sigma'\colon \sigma' \in \Sigma, P \notin \sigma'\}$;
(2) $\delta'(q, \sigma') = \delta(q, \sigma') \cup \delta(q, \sigma' \cup \{P\})$.

Then:

Theorem 3.5. [2] *An ω-expanded tree T is* accepted *by $\widetilde{\exists} PA$ if and only if there exists an ω-expanded tree S which is accepted by A and is bisimilar to T with respect to $Prop \setminus \{P\}$.*

We can now prove:

Theorem 3.6. *The uniform interpolant $\widetilde{\exists} P\phi$ of a disjunctive μ-formula ϕ is equivalent to the μ-formula $\phi[P/\top, \neg P/\top]$, where $\phi[P/\top, \neg P/\top]$ is defined from ϕ by simultaneously substituting the literals P and $\neg P$ with $\top$.*

Proof. Let A be a nondeterministic automaton in tree normal form which is equivalent to ϕ. From Theorem 3.5 it follows that the uniform interpolant of ϕ is equivalent to the automaton $A' = \widetilde{\exists} PA$; hence, we are just left to verify that $\phi_{A'}$ is equivalent to $\phi[P/\top, \neg P/\top]$. We prove by induction on the tree structure of the set of states of A that

$$\phi_{(A',q)} \text{ is equivalent to } \phi_{A,q}[P/\top, \neg P/\top], \quad \text{for all } q \in Q$$

and when $q = q_0$ we obtain the desired result. We have:

$$\phi_{A',q} = (\sigma_q X_q) \bigvee_{\sigma' \in \Sigma'} \bigvee_{D' \in \delta(q,\sigma') \cup \delta(q,\sigma' \cup \{P\})} \hat{\sigma}' \wedge Cover\bigl(\{\beta'_{q,q'}\colon q' \in D'\}\bigr),$$

where

$$\beta'_{q,q'} = \begin{cases} X_{q'} & \text{if } q' \leqslant_A q; \\ \phi_{A',q'} & \text{otherwise.} \end{cases}$$

On the other hand, the formula $\phi_{A,q}[P/\top, \neg P/\top]$ is

$$(\sigma_q X_q) \bigvee_{\sigma \in \Sigma} \bigvee_{D \in \delta(q,\sigma)} \hat{\sigma}[P/\top, \neg P/\top] \wedge Cover\bigl(\{\beta_{q,q'}[P/\top, \neg P/\top]\colon q' \in D\}\bigr),$$

where

$$\beta_{q,q'} = \begin{cases} X_{q'} & \text{if } q' \leqslant_A q; \\ \phi_{A,q'} & \text{otherwise.} \end{cases}$$

Using induction the equivalence between $\phi_{(A',q)}$ and $\phi_{A,q}[P/\top, \neg P/\top]$ easily follows. $\square$

From Theorem 3.6 it is intuitively clear that the uniform interpolant of a disjunctive formula is not more *complex* than the original formula. Before stating the result, however, we need a precise notion of complexity. One possibility is given by the syntactical hierarchy, whose definition is recalled in the next section.

4. The existential bisimulation quantifiers and the fixed point hierarchy

We consider now the behaviour of the existential bisimulation quantifier with respect to the fixpoint alternation levels of the μ-calculus. As an easy corollary of Theorem 3.6 we have that the uniform interpolant of a disjunctive formula ϕ belongs to the same level as ϕ. As we shall see, this is not true in general: in this section we prove that levels 0, 1 and 2 are closed under the existential bisimulation quantifier, while the third level is not: the closure of this level is the whole μ-calculus.

In this section we consider the μ-formulas as constructed from a set of propositional constants *Prop*, their negations $\{\neg P\colon P \in Prop\}$, a set of variables *Var*, using the following operators: if $\phi_1, \dots, \phi_n \in \mu$ and X in *Var* then $\phi_1 \vee \phi_2$, $\phi_1 \wedge \phi_2$, $Cover(\phi_1, \dots, \phi_n)$, $\mu X \phi_1$, and $\nu X \phi_1$ belong to μ.

Since $\Box(\phi)$ is semantically the same as $Cover(\emptyset) \vee Cover(\phi)$, this definition is equivalent to the one adopted in the previous sections, but it avoids the use of explicit negation.

Definition 4.1. The *fixpoint alternation-depth hierarchy* of the μ-calculus is the sequence $N_0 = M_0, N_1, M_1, \dots$ of sets of μ-formulas defined inductively as follows.

(1) $N_0 = M_0$ is defined as the set of all modal fixpoint free formulas over the covers-signature.
(2) N_{k+1} is the closure of $N_k \cup M_k$ under the operations described in (a), (b) below.
   (a) (Positive Substitution) If $\phi(P_1, \dots, P_n), \phi_1, \dots, \phi_n$ are in N_{k+1}, then $\phi(\phi_1, \dots, \phi_n)$ is in N_{k+1}, provided $P_1, \dots, P_n$ are positive in ϕ and no occurrence of a variable which was free in one of the ϕ_i becomes bound in $\phi(\phi_1, \dots, \phi_n)$.
   (b) If ϕ is in N_{k+1}, then $\nu X.\phi \in N_{k+1}$.
(3) Likewise, M_{k+1} is the closure of $N_k \cup M_k$ under positive substitution and the μ-operator.

From this definition and Theorem 3.6 it follows easily:

Corollary 4.2. *If ϕ is a disjunctive formula which belongs to the level N_k of the fixpoint alternation-depth hierarchy and $L' \subseteq L(\phi)$, then the uniform interpolant of ϕ w.r.t. L' belongs to the same level N_k.*

To generalize this result from disjunctive formulas to arbitrary μ-formulas we need to restrict ourselves to the levels N_0, N_1, N_2: we shall prove that for $i = 0, 1, 2$ any $\phi \in N_i$ is equivalent to a disjunctive formula in N_i, and then use Theorem 3.6. The fact that any μ-formula of these levels is equivalent to a disjunctive one of the same level is well known but we found no reference in the literature except for the zero level. For the sake of completeness we sketch here a proof which is an adaptation of the proof in [6,7] that any μ-formula is equivalent to a disjunctive one. This result is proved by using tableaux for μ-formulas. We first recall some definitions.

A μ-formula γ is *guarded* if every occurrence of the bound variable X in every subformula $\tau X.\alpha$ (for $\tau \in \{\mu, \nu\}$) is under the scope of a cover, and it is *well-named* if, for every bound variable X, there exists only one subformula of type $\tau X.\alpha$ in γ. It is possible to prove that any μ-formula is equivalent to a positive, well-named, and guarded formula [8,10], and that this formula can be found in linear time [11]. In this section we will only deal with this kind of formulas (henceforth simply called *formulas*). A bound variable X of such a formula γ is a μ-variable (ν-variable) if $\mu X.\alpha$ is a subformula of γ ($\nu X.\alpha$, respectively). We define a partial order $\preceq_\gamma$ on the set of the bound variables of a formula γ as the least partial order such that if $\tau X.\alpha$ is a γ-subformula, $\tau' Y.\beta$ is an α-subformula, and X occurs in β then $X \preceq_\gamma Y$. In other words, the bound variable Y is an immediate successor of the bound variable X if the scope of X contains Y and X occurs free in the scope of Y.

Remark 4.3. It follows easily from Definition 4.1 that a formula $\gamma \in N_1$ only contains ν-variables, while if a formula γ belongs to N_2 then there is no pair of variables (X, Y) appearing in γ where X is a μ-variable, Y is a ν-variable, and $X \preceq_\gamma Y$.

Definition 4.4. A *tableau* $\mathbf{T} = (T, L)$ for a μ-formula γ is a tree T (which we think as growing upwards) with a labeling function L such that the root is labeled by the set $\{\gamma\}$ and the sons of every node are created and labeled with sets of formulas according to the following rules:

$$\frac{\{\alpha\} \cup \Gamma \quad \{\beta\} \cup \Gamma}{\{\alpha \vee \beta\} \cup \Gamma}; \quad \text{(or)}$$

$$\frac{\{\alpha, \beta\} \cup \Gamma}{\{\alpha \wedge \beta\} \cup \Gamma}; \quad \text{(and)}$$

$$\frac{\{\alpha\} \cup \Gamma}{\{\tau X.\alpha\} \cup \Gamma}; \quad \text{(fixed points)}$$

$$\frac{\{\alpha\} \cup \Gamma}{\{X\} \cup \Gamma} \quad \text{where } \tau X.\alpha \text{ is a subformula of } \gamma; \quad \text{(reg)}$$

if $\Gamma = \{Cover(\mathcal{F}_1), \ldots, Cover(\mathcal{F}_n)\} \cup \Delta$, where Δ contains only propositional constants or negated propositional constants, then

$$\frac{\ldots \{\alpha\} \cup \{\bigvee \mathcal{F}_j \colon\ j \neq i\} \ldots}{\Gamma} \quad \text{(mod)}$$

where we have a successor labelled $\{\alpha\} \cup \{\bigvee \mathcal{F}_j \colon\ j \neq i\}$ for each $i \in \{1, \ldots, n\}$ and $\alpha \in \mathcal{F}_i$.

For example, an instance of the last rule is:

$$\frac{\{\alpha_1, \alpha_3\} \quad \{\alpha_2, \alpha_3\} \quad \{\alpha_1 \vee \alpha_2, \alpha_3\}}{\{Cover(\alpha_1, \alpha_2), Cover(\alpha_3), P, \neg Q\}};$$

if a node n in the tableau is labelled by $\{Cover(\alpha_1, \alpha_2), Cover(\alpha_3), P, \neg Q\}$ it will have 3 sons labelled respectively by $\{\alpha_1, \alpha_3\}\, \{\alpha_2, \alpha_3\}$, $\{\alpha_1 \vee \alpha_2, \alpha_3\}$.

We say that a variable X is *regenerated* in a node n if the regeneration rule (*reg*) is applied to n.

Definition 4.5. A *trace* on an infinite path $\mathcal{P}$ of a tableau $\mathbf{T} = (T, L)$ is a function F taking value on nodes n on an initial path of $\mathcal{P}$ such that $F(n) \in L(n)$ and whenever F is defined on n and m is the son of n in $\mathcal{P}$ then:

(1) if $F(n)$ is not reduced from n to m then $F(m) = F(n)$;
(2) if $F(n)$ is reduced from n to m then $F(n)$ is one of the result of this reduction where e.g.:
if the rule (*or*) is applied to n, $L(n) = \{\alpha \vee \beta\} \cup \Gamma$, $F(n) = \alpha \vee \beta$ and $L(m)$ is $\{\alpha\} \cup \Gamma$, then $F(m) = \alpha$;
if $L(n) = \{Cover(\mathcal{F}_1), \ldots, Cover(\mathcal{F}_n)\} \cup \Delta$ where Δ contains only propositional constants or negated propositional constants, $F(n) = Cover(\mathcal{F}_i)$, and $L(m)$ is $\{\alpha\} \cup \{\bigvee \mathcal{F}_j \colon\ j \neq i\}$ for $\alpha \in \mathcal{F}_i$, then $F(m) = \alpha$;

if $L(n) = \{Cover(\mathcal{F}_1), \ldots, Cover(\mathcal{F}_n)\} \cup \Delta$, where Δ contains only propositional constants or negated propositional constants, $F(n) = Cover(\mathcal{F}_h)$ and $L(m)$ is $\{\alpha\} \cup \{\bigvee \mathcal{F}_j\colon\ j \neq i\}$ for an $\alpha \in \mathcal{F}_i$ and $h \neq i$, then $F(m) = \bigvee \mathcal{F}_h$.

A trace is called a μ-trace (ν-trace) if it is an infinite trace in which the least variable (with respect to the order of dependence $\preceq_\gamma$) which is regenerated infinitely often is a μ-variable (a ν-variable, respectively).

It is then possible to prove that every infinite trace is either a μ- or a ν-trace, because along every trace there is always a least variable which is regenerated infinitely often.

Tableaux can be used to show that two formulas are equivalent: one can define a notion of tableau equivalence in such a way that if two tableaux are equivalent then the corresponding formulas are equivalent (see [7] for the definition).

Let us now summarize the main steps of the proof which allows to transform a μ-formula γ into an equivalent disjunctive formula $\hat{\gamma}$. What we will do in addition to the proof given in [7] is just to check that this transformation will not leave N_2 if the μ-formula is in N_2 (we leave the easier cases of levels N_0, N_1 to the reader). Let (T, L) be a tableau for the formula γ. The first step is to build a finite tree with "back edges" (T', L') (that is, a graph obtained from a finite tree by adding edges from some nodes to their ancestors), such that:

(1) (T', L') unwinds to (T, L);
(2) every node of (T', L') to which a back edge points (a "back node") is colored magenta or navy in such a way that for any infinite path from the unwinding of (T', L') we have: there exists a μ-trace on the path if and only if the highest node of (T', L') which appears infinitely often in the path is colored magenta;
(3) (only in the N_2-case) for no pair (m, n) of nodes in T' it holds: m is a back node colored magenta, n is a back node colored navy, and n lies on the path from m to a node k from which the back edge leading to m starts.

(T', L') can be constructed as follows: since γ is in N_2, a trace in T is a ν-trace if and only if it contains an infinite number of regenerations of ν-variables. Then one can build a deterministic Buchi automaton A on infinite words reading (the labels of the) infinite paths in T, and accepting only those paths having only ν-traces on them. The main idea for the construction of A is that the automaton must stay in a state of priority 1 until all traces have reached a new regeneration of a ν-variable, and when this happens, it will go to a state of 0-priority. Then it will start again, waiting until all traces have reached a new regeneration of a ν-variable and so on. A complete description of the automaton is given in Appendix A. Suppose we have such an automaton A, and let Ω be its parity function. If we run A on the infinite paths of the tableau, we may associate to each node n of the tableau a state $S(n)$ of the automaton in such a way that:

- if n_0 is the root of T, then $S(n_0)$ is the initial state of A;
- If m is a son of n then $(S(n), L(m), S(m))$ is a transition of the automata.

Then a path in T contains a μ-trace if and only if the least priority of the states appearing infinitely often on the path is odd. The set of nodes of (T', L') is then defined as the least subset of the set $\{(n, S(n))\colon n \in T\}$ such that:

(1) $(n_0, S(n_0))$ belongs to T', where n_0 is the root of T;
(2) if $(n, S(n)) \in T'$, m is a son of n in T, and there exists a node $(m', S(m'))$ such that:
 - $L(m') = L(m)$, $S(m') = S(m)$;
 - m' is an ancestor of m in T and for all nodes n'' on the path between m' and m we have $\Omega(m') \leqslant \Omega(n'')$;

 then we forget the node $(m, S(m))$ and build a back edge from $(n, S(n))$ to $(m', S(m'))$. If the preceding conditions are not fulfilled, then we add $(m, S(m))$ in T'.

We let $L'(m, S(m)) = L(m)$. We then color every back node $(m, S(m))$ magenta if $\Omega(S(m)) = 1$, or navy, if $\Omega(S(m)) = 0$, and doing so we see that the first two properties we required on (T', L') are fulfilled. As for the third property, suppose there exist a back node m colored magenta and a back node n colored navy which lies on the path from m to a node k from which the back edge leading to m starts. But this is impossible because, since there is a back edge from k to m, we should have $\Omega(S(m)) \leqslant \Omega(S(n))$, while $\Omega(S(m)) = 1$ and $\Omega(S(n)) = 0$. The second step in the proof is to construct the disjunctive formula $\hat{\gamma}$ from (T', L'): the construction starts from the leaves of the tree to the root; to all leaves from which a back edge starts leading to a node n we assign the variable X_n; this variable is then closed with a fixed point when we reach n, and the type of fixed point depends on the color of n: it will be a least fixed point if n is colored magenta, a greatest fixed point if n is colored navy. At the end of the construction we reach the root and the formula corresponding to the root will be the disjunctive formula $\hat{\gamma}$. Then one can prove that $\hat{\gamma}$ has a tableau which is equivalent to the tableau of γ, and hence γ is equivalent to $\hat{\gamma}$.

We will not enter in the details of this construction here, but we check that level N_2 of the syntactical hierarchy is preserved from γ to $\hat{\gamma}$. This is a consequence of the third property of (T', L'): $\hat{\gamma}$ will be in N_2 unless there exists a back node m colored magenta and a back node n colored navy which lies on the path from m to a node k from which the back edge leading to m starts, and we know that there are no such m, n. From the above discussion it follows:

Lemma 4.6. *A formula in N_k is equivalent to a disjunctive formula in N_k, for $k = 0, 1, 2$.*

4.1. Closure under bisimulation quantifiers

In [2] it is proved that the μ-calculus is closed under existential bisimulation quantifiers: if ϕ is a sentence of the μ-calculus, there exists a μ-sentence ψ which behaves like $\widetilde{\exists} P\phi$, that is:

$$M \models \psi \Leftrightarrow \exists N, \exists P \subseteq D^N \text{ with } N \sim_{Prop\setminus\{P\}} M \text{ and } (N, P) \models \phi.$$

The same result is known for modal logic, i.e., for level N_0 of the μ-calculus. Here we prove that the same holds for levels N_1 and N_2 of the μ-calculus hierarchy.

Theorem 4.7. *N_1 and N_2 are closed under existential bisimulation quantifiers on arbitrary models.*

Proof. Fix $k \in \{1, 2\}$ and $\phi \in N_k$. By Lemma 4.6 we know that ϕ is equivalent to a disjunctive formula ψ in N_k. By Theorem 3.6 we know that $\widetilde{\exists} P\psi$ is equivalent to $\psi[P/\top, \neg P/\top]$ which is still a formula in N_k. □

Corollary 4.8. *The uniform interpolant of a μ-formula ϕ in N_1 or N_2 belongs to the same level as ϕ.*

4.2. The power of two alternations

In the previous section we proved that N_1 and N_2 are closed under existential bisimulation quantifiers. Our next task is to show that this is not true after level N_2, because the whole μ-calculus is contained in the closure of level M_2. To prove this we shall use again the correspondence between the μ-calculus and nondeterministic automata introduced in Section 3. Since any μ-formula is equivalent to a disjunctive formula and disjunctive formulas correspond to nondeterministic automata we have:

Theorem 4.9. [7] *For any μ-sentence ϕ there exists a nondeterministic automaton A such that an ω-expanded tree satisfies ϕ if and only if it is accepted by A. Conversely, any nondeterministic automaton is equivalent to a μ-sentence.*

We now prove that any μ-sentence can be obtained from a sentence in M_2 by using a certain number of existential bisimulation quantifiers. We shall do this in two steps: in Lemma 4.10 we prove the analogous result over the class of ω-expanded trees using monadic second order existential quantifiers instead of existential bisimulation quantifiers. Then in Corollary 4.13 we go from ω-expanded trees to arbitrary models by considering bisimulation quantifiers instead of monadic quantifiers.

Lemma 4.10. *For any μ-sentence $\phi(P_1, \ldots, P_m)$ there exists a μ-sentence $\theta(P_1, \ldots, P_m, Q_1, \ldots, Q_n)$ with $\theta \in M_2$ such that for any ω-expanded tree T it holds*

$$T \models \phi \leftrightarrow \exists Q_1 \ldots \exists Q_n \theta.$$

Proof. By Theorem 4.9 for any μ-calculus formula ϕ there exists a nondeterministic automaton A which is equivalent to ϕ over ω-expanded trees. We now prove that the existence of an A-labeling over an ω-expanded tree can be expressed by using a finite number of monadic existential quantifiers $\exists Q_1 \ldots \exists Q_n$ over a formula θ in M_2.

By Definition 3.3 a nondeterministic automaton accepts an ω-expanded tree T iff T has an A-labeling. This labeling defines subsets $Q_0, \ldots, Q_n$ of the tree T (where Q_i corresponds to the set of points labelled by the state q_i) having the following properties:

(1) the sets $Q_0, \ldots, Q_n$ form a partition of T and the root of the tree belongs to Q_0;
(2) If $l(s) = q$ then $\{q' \in Q\colon \exists t \in Succ(s), l(t) = q'\}$ belongs to $\delta(Q, T(s))$;

(3) if $s_0, s_1, \ldots$ is an infinite path starting from the root and for every j the index i_j is such that $s_j \in Q_{i_j}$, then the least number appearing infinitely often in the sequence

$$\Omega(q_{i_0}), \quad \Omega(q_{i_1}), \quad \ldots,$$

is even.

Conversely, the existence of subsets $Q_0, \ldots, Q_n$ satisfying the above properties clearly allows us to construct an A-labeling of T.

It follows that A accepts T iff $T \models \exists Q_1 \ldots \exists Q_n(\phi_1 \wedge \phi_2 \wedge \phi_3)$, where ϕ_1, ϕ_2, ϕ_3 are formulas expressing the above points. We now show that ϕ_1, ϕ_2, ϕ_3 can be chosen to be in N_1, N_1, M_2, respectively. This is obvious for ϕ_1, which is equivalent to

$$Q_0 \wedge \nu X\left(\bigwedge_{i \neq j} \neg(Q_i \wedge Q_j) \wedge \Box X\right) \wedge \nu X\left(\left(\bigvee_i Q_i\right) \wedge \Box X\right).$$

As for ϕ_2, for any q_i, σ we consider the modal formula

$$f_{q_i,\sigma} = \bigvee_{\{Q_1,\ldots Q_n\} \in \delta(q_i,\sigma)} \Diamond(Q_1) \wedge \cdots \Diamond(Q_n) \wedge \Box(Q_1 \vee \cdots \vee Q_n).$$

We can then define ϕ_2 as the N_1-formula

$$\nu X\left(\left(\bigwedge_{i,\sigma}(Q_i \wedge \hat{\sigma} \to f_{q_i,\sigma})\right) \wedge \Box X\right),$$

where

$$\hat{\sigma} = \bigwedge_{P \in \sigma} P \wedge \bigwedge_{P \notin \sigma} \neg P.$$

To express ϕ_3 with a formula in M_2 we proceed as follows. First of all, notice that the existence of an infinite chain starting from a node w in which P appears infinitely often and Q appears in any point can be described by a μ-formula $\psi(P, Q)$ of the second level N_2:

$$\psi(P, Q) := \nu X \mu Y\big(P \wedge Q \wedge \Diamond(X)\big) \vee \big(Q \wedge \Diamond(Y)\big).$$

Fix an index k and substitute Q_k for P and the conjunction of

$$\big\{\neg Q_i\colon\ \Omega(q_i) < \Omega(q_k)\big\}$$

for Q in the formula $\psi(P, Q)$; we obtain a formula $\psi_k \in N_2$ which is true in w iff from w starts a chain in which infinitely many points are in Q_k, and no point is in any of the Q_i, for $\Omega(q_i) < \Omega(q_k)$. Finally, notice that point (3) above is expressed by the M_2-formula

$$\bigwedge_{\Omega(q_k)\ \text{odd}} \nu X(\neg\psi_k \wedge \Box X).$$

This proves that any μ-formula ϕ is equivalent over ω-expanded trees to a formula of type $\exists Q_1, \ldots, \exists Q_n \theta$, with $\theta \in M_2$. $\Box$

To prove our next step, we show in Lemma 4.12 that bisimulation quantifiers applied to bisimulation invariant formulas behave like *MSO*-quantifiers on the class of ω-expanded trees. But first we remark:

Lemma 4.11. *If the trees T, T' are ω-expanded and bisimilar, then they satisfy the same MSO-sentences.*

Proof. This holds because on ω-expanded trees any *MSO*-sentence is equivalent to a μ-calculus sentence [8], and bisimilar structures satisfy the same μ-sentences. □

Lemma 4.12. *If T is an ω-expanded tree and ϕ is a μ-sentence, then*

$$T \models \exists P\phi \leftrightarrow \widetilde{\exists} P\phi.$$

Proof. The implication from left to right is trivial and does not require that T is ω-expanded. Conversely, suppose $T \models \widetilde{\exists} P\phi$, i.e., that there exists N with $N \sim_{Prop\setminus\{P\}} T$ and $P \subseteq N$ such that $(N, P) \models \phi$. If N^ω is the ω-expansion of N we still have: there exists a $P \subseteq N^\omega$ with $(N^\omega, P) \models \phi$, that is: $N^\omega \models \exists P\phi$. But $\exists P\phi$ can be expressed as a second order property (in the language $Prop \setminus \{P\}$) of the structure N^ω, which is bisimilar to T w.r.t. this language. Hence, from Lemma 4.11 it holds: $T \models \exists P\phi$. □

Corollary 4.13. *For any μ-sentence $\phi(P_1, \dots, P_m)$ there exists a μ-sentence $\theta(P_1, \dots, P_m, Q_1, \dots, Q_n)$ with $\theta \in M_2$ such that*

$$\models \phi \leftrightarrow \widetilde{\exists} Q_1, \dots, \widetilde{\exists} Q_n\theta.$$

Proof. Lemma 4.10 implies there exists $\theta \in M_2$ such that ϕ and $\exists Q_1, \dots, \exists Q_n\theta$ are equivalent over ω-expanded models. Then Lemma 4.12 allows us to conclude that ϕ is equivalent to $\widetilde{\exists} Q_1, \dots, \widetilde{\exists} Q_n\theta$ over arbitrary models. □

5. A complete system for the μ-calculus with explicit uniform interpolants

Although by Corollary 4.13 there cannot be a simple rule for uniform interpolation of arbitrary formulas, we can still try to understand better how the existential bisimulation quantifier behaves w.r.t. the connectives and the operators of the μ-calculus.

To do so, we extend the original μ-language with the quantifier $\widetilde{\exists} P$ with the appropriate semantics (see Definition 2.6) and provide this extended logic $\widetilde{\mu}$ with a complete calculus. We shall see that to derive all validities in $\widetilde{\mu}$ we only need some standard principles allowing introduction and elimination of the bisimulation quantifier, plus some natural principles of commutativity between the existential bisimulation quantifier and the operators of the μ-calculus.

By inspecting the semantics of this quantifier we recognize easily that it enjoys at least the standard properties regarding substitutions and free variables. As usual, we say that the substitution of ψ for P in ϕ is *admissible* if no free variable of ψ becomes bound after the substitution for P in ϕ. If this is the case, we denote by $\phi[P/\psi]$ the formula obtained after the substitution.

The axiom and the rule for the existential bisimulation quantifiers are:

Ax1: $\phi[P/\psi] \to \widetilde{\exists} P\phi$ is provable, provided the substitution of ψ for P in ϕ is admissible;

R1: if $\phi \to \psi$ is provable, then $\widetilde{\exists} P\phi \to \psi$ is provable, provided P is not free in ψ.

The proof of the soundness of the above axiom and rule is left to the reader.

One could think that adding **Ax1** and **R1** to a Hilbert system which is complete for the μ-calculus (such as the Kozen system, proved to be complete in [13]) would give us the complete calculus for the extended logic, but this is not the case. Consider for example a valid principle as

$$\phi = \Diamond\top \to \widetilde{\exists} P\big(\Diamond(P) \wedge \Diamond(\neg P)\big).$$

It is easy to see that the system $\mu + \mathbf{Ax1} + \mathbf{R1}$ cannot prove ϕ: this is because all axioms and rules of $\mu + \mathbf{Ax1} + \mathbf{R1}$ are valid when we interpret the bisimulation quantifier $\widetilde{\exists}$ as a standard second order quantifier, while ϕ is not valid under this interpretation. Hence we need to add some more principles to $\mu + \mathbf{Ax1} + \mathbf{R1}$ in order to obtain a complete system. In this section we show that it is enough to add some simple *commutativity axioms*, relating the bisimulation quantifier $\widetilde{\exists}$ to disjunction, cover operators, and fixpoint operators.

First of all we prove that the existential bisimulation quantifier $\widetilde{\exists} P$ commutes with disjunctions and special conjunctions. In the next lemma, we denote $\sigma[P/\top, \neg P/\top]$ the formula obtained from a conjunction σ of literals by replacing every occurrence of P or $\neg P$ (if any) with $\top$.

Lemma 5.1. *If σ is a conjunction of a set of literals not containing both P and $\neg P$, and $\phi_1, \ldots, \phi_n$ are μ-formulas, then the following are valid formulas.*

$$\widetilde{\exists} P\sigma \leftrightarrow \sigma[P/\top, \neg P/\top], \qquad \widetilde{\exists} P(\phi_1 \vee \phi_2) \leftrightarrow \widetilde{\exists} P\phi_1 \vee \widetilde{\exists} P\phi_2$$
$$\widetilde{\exists} P\big(\sigma \wedge Cover(\phi_1, \ldots, \phi_n)\big) \leftrightarrow \widetilde{\exists} P\sigma \wedge Cover(\widetilde{\exists} P\phi_1, \ldots, \widetilde{\exists} P\phi_n).$$

Proof. To prove the validity of $\sigma[P/\top, \neg P/\top] \to \widetilde{\exists} P\sigma$, suppose a model M satisfies $\sigma[P/\top, \neg P/\top]$; then the model M' which is like M except that P is interpreted as $\{r^M\}$, if P belongs to the conjunction σ, and as $\emptyset$ otherwise, is bisimilar to M if we do not consider P in the language, and satisfies σ.

On the other hand, if a model M satisfies $\widetilde{\exists} P\sigma$, then σ is true in a model M' which is bisimilar to M w.r.t. the language of M minus P; then any propositional constant which is different from P and is true in M' must be also true in M and M satisfies $\sigma[P/\top, \neg P/\top]$.

The verification of commutativity of $\widetilde{\exists}$ with disjunction is left to the reader. To prove that

$$\widetilde{\exists} P\sigma \wedge Cover(\widetilde{\exists} P\phi_1, \ldots, \widetilde{\exists} P\phi_n) \to \widetilde{\exists} P\sigma \wedge Cover(\phi_1, \ldots, \phi_n),$$

suppose $\widetilde{\exists} P\sigma \wedge Cover(\widetilde{\exists} P\phi_1, \ldots, \widetilde{\exists} P\phi_n)$ holds in a model (M, V), where V is a valuation of the free variables in $\sigma, \phi_1, \ldots, \phi_n$. Fix a successor v of the root r^M of M and consider all formulas of type $\widetilde{\exists} P\phi_i$ it satisfies: for any such formula there exists a model $N_{v,i}$ and a valuation $V_{v,i}$ of the free variables of ϕ_i such that $(N_{v,i}, V_{v,i})$ is $Prop \setminus \{P\}$-bisimilar to (M, V) and $(N_{v,i}, V_{v,i}) \models \phi_i$. Consider a new model M' with a new root satisfying the same propositional constants as r^M and connected to all these $N_{v,i}$, when v varies in the successors of r^M. Define a valuation V' over a variable X as

$$V'(X) = \bigcup_{v,i} V_{v,i}(X), \quad \text{if } r^M \notin V(X),$$

and

$$V'(X) = \{r^M\} \cup \bigcup_{v,i} V_{v,i}(X), \quad \text{otherwise.}$$

Then (M', V') is $Prop \setminus \{P\}$-bisimilar to (M, V) and verifies the formula $\sigma \wedge Cover(\phi_1, \ldots, \phi_n)$.

The verification of the validity of the reverse arrow is left to the reader. □

As a first step towards a complete calculus for $\widetilde{\mu}$, let us prove that the principles discovered so far are complete if we do not consider fixpoint operators, that is, if we only consider modal logic K extended with the existential bisimulation quantifier. Let us denote the extended logic by $\widetilde{K}$.

Theorem 5.2. *Consider the Hilbert calculus for $\widetilde{K}$ consisting of the following axioms and rules*:

(1) *a complete Hilbert system K of axioms and rules for modal logic*;
(2) *the axiom* **Ax1** *and the rule* **R1**;
(3) *if σ is a non contradictory conjunction of literals and $\phi_1, \ldots, \phi_n$ are formulas, the axiom*

$$\widetilde{\exists} P\big(\sigma \wedge Cover(\phi_1, \ldots, \phi_n)\big) \leftrightarrow \sigma[P/\top, \neg P/\top] \wedge Cover(\widetilde{\exists} P\phi_1, \ldots, \widetilde{\exists} P\phi_n).$$

Then this calculus is sound and complete for modal logic extended with the existential bisimulation quantifier.

Proof. To prove that $\widetilde{K}$ is complete it is enough to show that for any formula ψ of the logic there exists a modal formula ψ^- such that $\psi \leftrightarrow \psi^-$ is provable in $\widetilde{K}$: if this is true, to derive a valid formula ψ in $\widetilde{K}$ we can derive the (provably equivalent and) valid modal formula ψ^- instead; but $\widetilde{K}$ proves ψ^- because K proves it (being a complete calculus for modal formulas) and $\widetilde{K}$ is an extension of K.

We find the formula ψ^- by induction on the structural complexity of ψ, the only interesting case being $\psi = \widetilde{\exists} P\phi$. By induction, we suppose that ϕ is provably equivalent to a modal formula ϕ^-. Now, any modal formula is semantically equivalent to a disjunctive formula ϕ^-_d, and $\widetilde{K}$ can prove this equivalence since our system contains the complete system K. By the existential rule and axiom we have

$$\widetilde{\exists} P\phi \leftrightarrow \widetilde{\exists} P\phi^-_d,$$

and we only have to prove that $\widetilde{\exists} P\phi_d^-$ is provably equivalent to a modal formula. By Theorem 3.6, we know that this formula is the modal formula $\phi_d^-[P/\top, \neg P/\top]$. Moreover, if fixpoint operators are not present the (semantical) equivalence between $\widetilde{\exists} P\phi_d^-$ and $\phi_d^-[P/\top, \neg P/\top]$ can be easily proved inside $\widetilde{K}$ by using induction on the structural complexity of the disjunctive formula ϕ_d^-, the cover axioms, **Ax1**, and **R1**.

So we let $\psi^- = \phi_d^-[P/\top, \neg P/\top]$. □

We now go back to the μ-calculus. The strategy to find a complete calculus for this logic is the same as for $\widetilde{K}$, that is: we use the explicit form (in the original μ-language) of uniform interpolants of disjunctive μ-formulas (see Theorem 3.6). First we prove that the existential bisimulation quantifier $\widetilde{\exists} P$ commutes with the fixpoint operators μX, νX, provided the *contexts* $\mu X.\phi$, $\nu X.\phi$ are disjunctive.

Corollary 5.3. *If $\mu X.\phi$ and $\nu X.\phi$ are disjunctive formulas then*

$$\models \widetilde{\exists} P\mu X.\phi \leftrightarrow \mu X.\widetilde{\exists} P\phi \qquad \models \widetilde{\exists} P\nu X.\phi \leftrightarrow \nu X.\widetilde{\exists} P\phi.$$

Proof. By Theorem 3.6, the formula $\widetilde{\exists} P\mu X.\phi$ is equivalent to $(\mu X.\phi)[P/\top, \neg P/\top]$, which is the same as $\mu X.(\phi[P/\top, \neg P/\top])$, which is equivalent to $\mu X.\widetilde{\exists} P\phi$. The proof for the operator ν is similar. □

Notice that these equivalences are not true without the disjunctivity hypothesis. Consider for example the formula $\phi = P \wedge \Diamond(\neg P) \wedge \Box X$. We have $\widetilde{\exists} P\phi = \Diamond\top \wedge \Box X$, hence $\nu X.\widetilde{\exists} P\phi = \nu X.\Diamond\top \wedge \Box X$, which is true in a model M iff all nodes accessible from the root satisfy $\Diamond\top$. Thus, the formula $\nu X.\widetilde{\exists} P\phi$ is satisfiable. On the other hand, the formula $\nu X.\phi$ is equivalent to $\perp$, and so is $\widetilde{\exists} P\nu X.\phi$.

In the next theorem we show that adding commutativity between $\widetilde{\exists}$ and fixpoint operators in a disjunctive context to the principles presented in Theorem 5.2 is enough to obtain a complete system for $\widetilde{\mu}$.

Theorem 5.4. *Consider the Hilbert calculus $\widetilde{\mu}$ consisting of the following axioms and rules*:

(1) *axioms and rules of the system $\widetilde{K}$ (see Theorem 5.2)*;
(2) *a complete system of Hilbert axioms and rules for the μ-calculus (e.g. the Kozen system, proved to be a complete system in [13])*;
(3) *if $\mu X.\phi$ is disjunctive, the axiom $\widetilde{\exists} P\mu x.\phi \leftrightarrow \mu x.\widetilde{\exists} P\phi$*;
(4) *if $\nu X.\phi$ is disjunctive, the axiom $\widetilde{\exists} P\nu x.\phi \leftrightarrow \nu x.\widetilde{\exists} P\phi$.*

Then $\widetilde{\mu}$ is sound and complete for the μ-calculus extended with the existential bisimulation quantifier, that is: a formula ϕ of this logic is valid if and only if it is derivable within the system $\widetilde{\mu}$.

Proof. To prove that $\widetilde{\mu}$ is complete, it is enough to show that for any formula ψ of the logic there exists a μ-formula ψ^- such that $\psi \leftrightarrow \psi^-$ is provable in $\widetilde{\mu}$. This can be achieved by induction on the structural complexity of the formula, the only interesting case being $\psi = \widetilde{\exists} P\phi$. By induction, we suppose that ϕ is provably equivalent to a μ-formula ϕ^-, which is in turn provably equivalent to a disjunctive μ-formula ϕ_d^- since our system contains the complete system μ. By the existential rule and axiom we have

$$\widetilde{\exists} P\phi \leftrightarrow \widetilde{\exists} P\phi_d^-.$$

By induction on the structure of the disjunctive formula ϕ_d^- we prove using the axioms and the rules above that $\widetilde{\exists} P\phi_d^- \leftrightarrow \phi_d^-[P/\top, \neg P/\top]$. Hence $\widetilde{\exists} P\phi$ is provably equivalent to the μ-formula $\phi_d^-[P/\top, \neg P/\top]$.

So we let $\psi^- = \phi_d^-[P/\top, \neg P/\top]$. □

6. Conclusions and related work

In this paper we gave a simple rule for the uniform interpolant of a disjunctive formula, and studied the behaviour of the existential bisimulation quantifiers w.r.t. the alternation-depth hierarchy of the μ-calculus. We also gave an axiomatization of the μ-calculus extended with the existential bisimulation quantifier, allowing in this way the possibility of computing with a logic as powerful as the μ-calculus in which we have a way to denote uniform interpolants explicitly. A related question is the axiomatization of *BQL* [5] which is defined as the bisimulation quantifier closure of Propositional Dynamic Logic *PDL*. The logic *BQL* is semantically equivalent to the μ-calculus, but has the advantage of replacing all fixpoint operators (which are difficult to read, especially when nesting of two or more operators occur) by the Kleene star and the existential bisimulation quantifier. The structure of formulas expressing a certain property are closer to natural language in *BQL* than in the μ-calculus (see Example 2 in Section 2.2). A complete system of axioms and rules for *BQL* is presented in [3].

Appendix A

Given a μ-formula γ with a tableau T, we construct a deterministic Buchi automaton A on infinite words which recognises exactly the infinite paths in the tableau having only ν-traces on them. We suppose that the tableau's nodes (except the root) are labelled by a pair in which the first component gives the formula which has been reduced in the father of the node, and the second component gives the label of the node in the tableau. E.g. if we are in node n which was created because of the rule *and* applied to the father m to the formula $\alpha = \beta \wedge \gamma$ then the label of n will be $(\alpha, \{\beta, \gamma, \ldots\})$ The automaton A is defined as follows. The set of states is given by all sets of the form $\{(\alpha_1, u_1), \ldots, (\alpha_n, u_n)\}$ where u_i is either s (for "search") or f (for found) and $\{\alpha_1, \ldots, \alpha_n\}$ is the second component of a label of the tableau, where all propositional constants have been removed. This implies in particular that all α_j are different. When defining the transitions of the automaton we identify pairs $(\alpha, s), (\alpha, f)$ with the single element (α, s).The initial position is $\{(\gamma, s)\}$. The states having all element with second component equal to f are of priority 0, while the other states have priority 1. The automaton reads paths in T (but it skips the label of the root). We will define A in such a way that when A reads a path $(L(root))L(n_1), \dots, L(n_i)\dots$ of the tableau, it will be in a state $\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\}$ just after reading a label $\{\alpha_1, \dots, \alpha_n\}$. If A is in state $\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\}$, where not all u_i are equal to f, it reads a label (θ, L), and there exists i such that $\theta = \alpha_i$, then:

(1) if $\alpha_i = \beta \wedge \delta$ and $L = (\{\alpha_1, \dots, \alpha_n\} \setminus \{\alpha_i\}) \cup \{\beta, \delta\}$ then A goes to the state

$$\big\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\big\} \setminus \big\{(\alpha_i, u_i)\big\} \cup \big\{(\beta, u_i), (\delta, u_i)\big\};$$

(2) if $\alpha_i = \beta \vee \delta$ and $L = (\{\alpha_1, \dots, \alpha_n\} \setminus \{\alpha_i\}) \cup \{\epsilon\}$ (where ϵ is either β or δ), then A goes to the state

$$\big\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\big\} \setminus \big\{(\alpha_i, u_i)\big\} \cup \big\{(\epsilon, u_i)\big\};$$

(3) if $\alpha_i = \sigma X \alpha$ where σ is ν or μ and $L = (\{\alpha_1, \dots, \alpha_n\} \setminus \{\alpha_i\}) \cup \{\alpha\}$, then A goes to the state

$$\big\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\big\} \setminus \big\{(\alpha_i, u_i)\big\} \cup \big\{(\alpha, u_i)\big\}$$

(4) if $\alpha_i = X$, the binding of X is α, and $L = (\{\alpha_1, \dots, \alpha_n\} \setminus \{X\}) \cup \{\alpha\}$, then A goes to the state

$$\big\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\big\} \setminus \big\{(\alpha_i, u_i)\big\} \cup \big\{(\alpha, s)\big\},$$

if X is a ν-variable, and to the state

$$\big\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\big\} \setminus \big\{(\alpha_i, u_i)\big\} \cup \big\{(\alpha, u_i)\big\},$$

if X is a μ-variable;

(5) if $\alpha_1 = Cover(\mathcal{F}_1), \dots, \alpha_n = Cover(\mathcal{F}_n)$ and there exists a formula $\alpha \in \mathcal{F}_i$ such that $L = \{\alpha\} \cup \{\bigvee \mathcal{F}_i\colon\ j \neq i\}$ then A goes to the state

$$\big\{(\alpha, u_i)\big\} \cup \Big\{\Big(\bigvee \mathcal{F}_j, u_j\Big)\colon\ j \neq i\Big\}.$$

If A is in state $\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\}$, where all u_i are equal to f, we just consider the previous transitions, but as if we where starting from the state $\{(\alpha_1, s), \dots, (\alpha_n, s)\}$ instead of $\{(\alpha_1, u_1), \dots, (\alpha_n, u_n)\}$. If none of the above condition is fulfilled, the automaton stops.

References

[1] J. Bradfield, The modal μ-calculus alternation hierarchy is strict, Theoret. Comput. Sci. 195 (1998) 133–153.

[2] G. D'Agostino, M. Hollenberg, Logical questions concerning the μ-calculus: Interpolation, Lyndon and Los-Tarski, J. Symbolic Logic 65 (2000) 310–332.

[3] G. D'Agostino, G. Lenzi, An axiomatization of bisimulation quantifiers via the mu-calculus, Theoret. Comput. Sci. 338 (1–3) (2005) 64–95.

[4] S. Ghilardi, M. Zawadowski, A sheaf representation and duality for finitely presented Heyting algebras, J. Symbolic Logic 60 (1995) 911–939.

[5] M. Hollenberg, Logic and bisimulation, PhD Thesis, University of Utrecht, vol. XXIV, Zeno Institute of Philosophy, 1998.

[6] D. Janin, Propriété logique du non determinisme et μ-calcul modal, Thèse du doctorat, Université Bordeaux I.
[7] D. Janin, I. Walukiewicz, Automata for the modal μ-calculus and related results, in: Proceedings of the Conference MFCS'95, in: Lecture Notes in Computer Science, vol. 969, Springer, Berlin, 1995, pp. 552–562.
[8] D. Janin, I. Walukiewicz, On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic, in: Proceedings of the Conference CONCUR '96, in: Lecture Notes in Computer Science, vol. 1119, Springer, Berlin, 1996, pp. 263–277.
[9] M. Jurdziński, Small progress measures for solving parity games, in: Proceedings of the Conference STACS 2000, in: Lecture Notes in Computer Science, vol. 1770, Springer, Berlin, 2000, pp. 290–301.
[10] D. Kozen, Results on the propositional μ-calculus, Theoret. Comput. Sci. 27 (1983) 333–354.
[11] O. Kupferman, M. Vardi, P. Wolper, An automata-theoretic approach to branching time model checking, J. ACM 47 (2) (2000) 312–360.
[12] A. Visser, Bisimulations, model descriptions and propositional quantifiers, Logic Group Preprint Series 161, Department of Philosophy, Utrecht University, 1996.
[13] I. Walukiewicz, Completeness of Kozen's axiomatization of the propositional μ-calculus, Inform. and Comput. 157 (2000) 142–182.