THE JOURNAL OF SYMBOLIC LOGIC
Volume 65, Number 1, March 2000

LOGICAL QUESTIONS CONCERNING THE μ-CALCULUS: INTERPOLATION, LYNDON AND ŁOŚ-TARSKI

GIOVANNA D'AGOSTINO AND MARCO HOLLENBERG

§1. **Introduction.** The (modal) μ-calculus ([14]) is a very powerful extension of modal logic with least and greatest fixed point operators. It is of great interest to computer science for expressing properties of processes such as termination (every run is finite) and fairness (on every infinite run, no action is repeated infinitely often to the exclusion of all others).

The power of the μ-calculus is also evident from a more theoretical perspective. The μ-calculus is a fragment of monadic second-order logic (MSO) containing only formulae that are *invariant for bisimulation*, in the sense that they cannot distinguish between bisimilar states. Janin and Walukiewicz prove the converse: any property which is invariant for bisimulation and MSO-expressible is already expressible in the μ-calculus ([13]). Yet the μ-calculus enjoys many desirable properties which MSO lacks, like a complete sequent-calculus ([29]), an exponential-time decision procedure, and the finite model property ([25]). Switching from MSO to its bisimulation-invariant fragment gives us these desirable properties.

In this paper we take a classical logician's view of the μ-calculus. As far as we are concerned a new logic should not be allowed into the community of logics without at least considering the standard questions that any logic is bothered with. In this paper we perform this rite of passage for the μ-calculus. The questions we will be concerned with are the following.

First, there is the question of interpolation. That is, if ϕ, ψ are μ-formulae with $\phi \models \psi$, is there then a μ-formula χ such that $\phi \models \chi \models \psi$ and furthermore such that χ is in their common language (i.e., uses only the non-logical symbols that occur both in ϕ and ψ)? Such a formula χ we would then call the *interpolant*. This question was originally considered by Craig for first-order logic, and answered positively ([3]). For MSO it is easily shown to be false.

In this paper we demonstrate that the μ-calculus belongs to the class of logics that *do* have interpolation. Something even stronger can be proved: the interpolant can be constructed using only one of the formulae ϕ or ψ in combination with their common language. We say that the μ-calculus enjoys *uniform interpolation*. In this sense it is a nicer logic than first-order logic, which does not have this stronger form of interpolation. This property puts the μ-calculus into the class of logics also

Received June 6, 1997; revised June 23, 1998.

containing ordinary modal logic ([28]) and propositional classical and intuitionistic logic (respectively [9] and [20]).

We can express uniform interpolation in terms of modularization. Suppose we have a specification of a process in the form of a μ-calculus formula ϕ but for the moment we are only interested in a particular subset of the language of ϕ. Then we would like to extract a formula ψ that only deals with this sublanguage, yet is equivalent to ϕ as far as this sublanguage is concerned. This would be like a module for this subtask. Uniform interpolation tells us that we can always find such a formula ψ: the μ-calculus allows modularization.

We prove uniform interpolation by demonstrating that certain non-standard second-order quantifiers, namely so-called *bisimulation quantifiers* ([28]), are definable within the μ-calculus. These quantifiers are of interest by themselves. As an application, we show that adding bisimulation-quantifiers to Propositional Dynamic Logic (PDL, [22, 7]) results in a logic equal in expressivity to the μ-calculus.

The second question we consider concerns the Lyndon Theorem. Lyndon ([16], Theorem 10.3.3 in [10]) proved that a first-order formula $\phi(R)$ is monotone in the interpretation of R if and only if it is equivalent to a formula where R occurs only positively. Most logics seem to have a Lyndon Theorem. In fact, the only logic that we know of that doesn't is first-order logic restricted to finite models ([4, 8]). Nevertheless, whenever we encounter a new logic, a proof must be supplied.

The Lyndon Theorem is of particular importance to the μ-calculus, as monotonicity ensures the existence of a least fixed-point. This is the reason there is a call to positivity embedded in some definitions of the μ-calculus language. This appeal to positivity cannot be replaced by one to monotonicity, as this would make the syntax dependent on the semantics. In this paper we provide the μ-calculus with a Lyndon Theorem. This shows that we do not in fact lose anything by using positivity instead of monotonicity in the definition of the language.

Third, and finally, we ask whether the μ-calculus has a Łoś-Tarski Theorem: is there a characterization of formulae preserved under substructures? For first-order logic there is a very nice characterization: the fragment of formulae whose truth is preserved under substructures corresponds modulo logical equivalence to the set of universal formulae ([26, 15], Theorem 3.2.2 of [1]). For modal logic the characterization works as follows ([23, 24]): any modal formula that is preserved under substructures is equivalent to a modal formula whose first-order translation is universal (there's also a definition in purely modal terms, avoiding the use of diamonds, instead using boxes). In this paper we extend the latter result to the μ-calculus.

What all our results have in common is the use of *μ-automata* ([12]) in the proofs. These are a generalization of Rabin tree-automata ([27]) from infinite binary trees to arbitrary labeled transition systems. These automata were proved to be equivalent in recognizing power to the μ-calculus in [12].

The paper is structured as follows. Section 2 introduces the main concepts and tools of the paper, most notably the μ-calculus itself and μ-automata. The three sections thereafter deal with uniform interpolation, the Lyndon Theorem and the Łoś-Tarski Theorem respectively. We conclude the paper with some general remarks and suggestions for future research.

§**2. Preliminaries.** We consider languages $\mathscr{L}$ built up from a set PROP of proposition constants and a set A of atomic actions as follows: there is a single constant r (for 'root'), each $p \in$ PROP is a unary predicate symbol of $\mathscr{L}$ and for each $a \in A$ there is a binary relation symbol R_a in $\mathscr{L}$.

If $\mathscr{L}$ is such a language, then a *process graph* for $\mathscr{L}$ is nothing but a structure for this language: we will accordingly also refer to process graphs for $\mathscr{L}$ as '$\mathscr{L}$-structures'. A process graph without information about its root is a *labeled transition system*. A process graph will sometimes be denoted as $(\mathscr{M}, s)$, where $\mathscr{M}$ is a transition system and s is some element in the domain of $\mathscr{M}$, intended to be the root. If $p \in$ PROP and S is a subset of the domain of a process graph $\mathscr{M}$, we denote by $\mathscr{M}[p := S]$ the process graph which is like $\mathscr{M}$ except that p is now interpreted as S.

If the $\mathscr{L}$-structure $\mathscr{M}$ is clear from the context, we write $s \xrightarrow{a} t$ for $(s,t) \in R_a^{\mathscr{M}}$, where the latter is the interpretation of R_a in $\mathscr{M}$. For each $a \in A$, the set of a-successors of a point s in a structure $\mathscr{M}$ is defined as: $\text{SUCC}_a(s) := \{t \mid s \xrightarrow{a} t\}$. The set of successors of s (not relativized to a specific action) is defined as $\text{SUCC}(s) := \bigcup_{a \in A} \text{SUCC}_a(s)$.

DEFINITION 2.1. Given a language $\mathscr{L}$ as above and an infinite set of variables VAR, the set of formulae of the *modal μ-calculus* is defined as follows:

$$\phi ::= p \mid \neg p \mid X \mid \phi \vee \phi \mid \phi \wedge \phi \mid \langle a \rangle \phi \mid [a]\phi \mid \mu X.\phi \mid \nu X.\phi$$

where $p \in$ PROP, $X \in$ VAR and $a \in A$.

We will refer to these formulae as *μ-formulae*. If we want to stress the language, we speak of $\mathscr{L}$-formulae. The *language of a formula* ϕ is simply the set of proposition constants and atomic actions that are mentioned in ϕ. We denote it by $\mathscr{L}(\phi)$.

A μX- and νX-operator binds occurrences of X in the usual way: the notion of *closed* μ-formula, or *μ-sentence*, is then obvious.

We adopt the following convention: whenever a formula is introduced as

$$\phi(p_1, \dots, p_n)$$

this is just to indicate the formula ϕ with special emphasis on the propositional constants $p_1, \dots, p_n$. Later on in the text we may use 'ϕ' to indicate this same formula. Furthermore, in the same context, $\phi(\psi_1, \dots, \psi_n)$ indicates the formula $\phi[p_1 := \psi_1, \dots, p_n := \psi_n]$, that is: the formula ϕ where each p_i is simultaneously replaced by ψ_i. The same convention will be used for variables.

Given a structure $\mathscr{M}$ and a valuation $V : \text{VAR} \to \wp(\mathscr{M})$ (where we denote the domain of $\mathscr{M}$ with $\mathscr{M}$ itself), a μ-formula is interpreted in $\mathscr{M}$ as $[\![\phi]\!]_V$, defined as follows:

$$\begin{aligned}
[\![p]\!]_V &:= p^{\mathscr{M}} \\
[\![\neg p]\!]_V &:= \mathscr{M} \setminus p^{\mathscr{M}} \\
[\![X]\!]_V &:= V(X) \\
[\![\phi \vee \psi]\!]_V &:= [\![\phi]\!]_V \cup [\![\psi]\!]_V \\
[\![\phi \wedge \psi]\!]_V &:= [\![\phi]\!]_V \cap [\![\psi]\!]_V \\
[\![\langle a \rangle \phi]\!]_V &:= \{s \in \mathscr{M} \mid [\![\phi]\!]_V \cap \text{SUCC}_a(s) \neq \emptyset\}
\end{aligned}$$

$$[\![[a]\phi]\!]_V := \{s \in \mathscr{M} \mid \text{SUCC}_a(s) \subseteq [\![\phi]\!]_V\}$$
$$[\![\mu X.\phi]\!]_V := \bigcap\{S \subseteq \mathscr{M} \mid [\![\phi]\!]_{V[X:=S]} \subseteq S\}$$
$$[\![\nu X.\phi]\!]_V := \bigcup\{S \subseteq \mathscr{M} \mid S \subseteq [\![\phi]\!]_{V[X:=S]}\}$$

where $V[X := S]$ is equal to the valuation function V except that X is assigned to S. Note that $[\![\mu X.\phi]\!]_V$ is the least fixed point of the monotone operator $S \mapsto [\![\phi]\!]_{V[X:=S]}$. Similarly $[\![\nu X.\phi]\!]_V$ is the greatest fixed point of this operator.

Note that if ϕ is a formula containing as free variables only $X_1, \dots, X_n$ the valuation need only choose a value for these variables. In such a case we may write $[\![\phi]\!]_{[X_1:=A_1,\dots,X_n:=A_n]}$ instead of $[\![\phi]\!]_V$. We may leave out the valuation altogether if ϕ is a sentence.

We sometime denote $s \in [\![\phi]\!]_V$ by $\mathscr{M}, V, s \models \phi$. If ϕ is a sentence we may leave out the valuation. $\mathscr{M} \models \phi$ is used to denote $\mathscr{M}, r^{\mathscr{M}} \models \phi$, that is: the root satisfies ϕ. Logical consequence is then defined accordingly: if Γ is a set of μ-sentences and ϕ is a μ-sentence, then $\Gamma \models \phi$ if and only if for all structures $\mathscr{M}$, if $\mathscr{M} \models \Gamma$ then $\mathscr{M} \models \phi$.

Our definition of the μ-calculus avoids the use of negation everywhere but on proposition constants. Actually, negation is definable on *sentences*. An alternative definition of the μ-calculus uses explicit negation. But in such a formulation the μX-operator is restricted to apply only to formulae where X occurs just positively, that is: under an even number of negations. This is to ensure monotonicity of the corresponding operation. Once negation is available on sentences, we can also define implication on sentences, as $\phi \to \psi := \neg\phi \vee \psi$.

The modal μ-calculus is a very expressive formalism, into which many other well-studied modal formalisms can be embedded. Examples are Linear Temporal Logic (LTL, [21]), Computation Tree Logic (CTL, [2]), CTL* ([5, 6]), Propositional Dynamic Logic (PDL, [22, 7]) and modal logic. By the latter we mean the logic we get by leaving variables and fixed point operators out of the μ-calculus. We will also refer to this basic system as just *modal logic*.

DEFINITION 2.2. Let $\mathscr{M}$ and $\mathscr{N}$ be two $\mathscr{L}$-structures and let $Z \subseteq \mathscr{M} \times \mathscr{N}$. Z is a *bisimulation* between $\mathscr{M}$ and $\mathscr{N}$ (notation $Z : \mathscr{M} \sim \mathscr{N}$) if:

(1) $r^{\mathscr{M}} Z r^{\mathscr{N}}$;
(2) Z-connected points agree on the proposition constants: if sZt then $\mathscr{M}, s \models p$ if and only if $\mathscr{N}, t \models p$ for every unary p in $\mathscr{L}$.
(3) $s \xrightarrow{a} s'$ and sZt implies that there is a t' such that $s'Zt'$ and $t \xrightarrow{a} t'$;
(4) Vice versa: if sZt and $t \xrightarrow{a} t'$ then there is an s' with $s \xrightarrow{a} s'$ and $s'Zt'$.

Two structures $\mathscr{M}$ and $\mathscr{N}$ are *bisimilar* (notation $\mathscr{M} \sim \mathscr{N}$) if there exists a bisimulation between them. If $\mathscr{M}$ and $\mathscr{N}$ are structures of respectively $\mathscr{L}_1$ and $\mathscr{L}_2$, both of which contain $\mathscr{L}$, then an *$\mathscr{L}$-bisimulation* between $\mathscr{M}$ and $\mathscr{N}$ is a relation Z satisfying the above clauses just for the symbols in $\mathscr{L}$. The notion of $\mathscr{L}$-bisimilar structures is defined accordingly (notation $Z : \mathscr{M} \sim_{\mathscr{L}} \mathscr{N}$, $\mathscr{M} \sim_{\mathscr{L}} \mathscr{N}$). ⊣

A sentence ϕ in some language that can speak about transition systems is *invariant for bisimulation* if whenever $Z : \mathscr{M} \sim \mathscr{N}$ then ϕ holds in $\mathscr{M}$ if and only if it holds in $\mathscr{N}$. The notion is thus a generalization of the clause for proposition constants in the definition of bisimulation. It is known that μ-sentences are invariant for bisimulation, and, more precisely, that any $\mathscr{L}$-sentence (that is: any μ-sentence that only uses symbols from the language $\mathscr{L}$) is invariant for $\mathscr{L}$-bisimulation. We will not use it, but we would like to mention an interesting converse: Janin and Walukiewicz ([13]) prove that any monadic second-order logic sentence that is invariant for bisimulation is equivalent to a μ-sentence.

Consider the full binary tree, each of whose nodes carries a unique label from some finite set Σ. Such a tree is called a Σ-*valued tree*. Tree automata theory ([27]) studies automata that operate on such infinite trees. Σ-valued trees are easily seen to be process graphs, where the root is simply the root of the tree, Σ corresponds to the set of propositional constants and we have two atomic actions: each edge from a node to its daughter on the left is labeled with l and each edge to its right-daughter is labeled with r. Niwiński ([19]) has shown that on such structures the μ-calculus corresponds to *Rabin automata* (and hence to monadic second-order logic): for each μ-sentence ϕ there is a Rabin-automaton that recognizes exactly those trees at whose root ϕ holds. The converse also holds, to each Rabin automaton $\mathscr{A}$ there is a μ-sentence that holds precisely at all roots of trees that are accepted by $\mathscr{A}$.

Rabin automata are defined to operate only on infinite binary trees, so they are of no direct use to the present enterprise. Janin and Walukiewicz ([12]) define a notion of automaton (a so-called μ-*automaton*) that operates on arbitrary process graphs and which corresponds exactly to the μ-calculus on such graphs.

DEFINITION 2.3. A μ-*automaton* ([12]) $\mathscr{A}$ is a tuple $(Q, \Sigma, \Lambda, q_0, \delta, \Omega)$ such that:

(1) Q is a finite set of states;
(2) Σ is a finite subset of PROP;
(3) Λ is a finite subset of A;
(4) $q_0 \in Q$ is the initial state;
(5) $\delta : Q \times \wp(\Sigma) \to \wp\wp(\Lambda \times Q)$ is the transition function;
(6) $\Omega : Q \to \mathbb{N}$ is the *parity function*. ⊣

We will also refer to such an automaton as a (Σ, Λ)-automaton, to stress the language of the automaton.

Roughly speaking, the transition function δ of an automaton $\mathscr{A}$ gives a set of rules for *labeling* with Q-elements the set of successors of a node s of an $\mathscr{L}$-structure $\mathscr{M}$, which has been previously labeled by q. To label such a set, choose a D in $\delta(q, \{p \in \Sigma \mid s \models p\})$. D gives necessary conditions that the labeling of $\text{SUCC}(s)$ must fulfill: for any action $a \in \Lambda$, the set of states which appear as labels of a-successors of s must be exactly the set $\{(a, q) \mid (a, q) \in D\}$. This is not entirely correct, though, because if for example s have only one a-successor t and D contains $(a, q_1), (a, q_2)$ for $q_1 \neq q_2$, we are allowed to label t twice, with q_1 and q_2: we are labeling the set of successors of s by *nonempty* subsets of Q.

The formal definition of acceptance of an $\mathscr{L}$-structure $\mathscr{M}$ by the automaton $\mathscr{A}$ is given by means of two players, the Duplicator and the Spoiler, which play on the structure $\mathscr{M}$ following the rules given by $\mathscr{A}$: a μ-*game* of $\mathscr{A}$ on $\mathscr{M}$ (or: the $\mathscr{A}$-game on $\mathscr{M}$, notation: $\mathscr{G}(\mathscr{A}, \mathscr{M})$) is defined as follows.

(1) The starting position is $(r^{\mathscr{M}}, q_0)$;
(2) If we are in a position (s, q) then the Duplicator has to make a move. Define $L(s)$ as $\{p \in \Sigma \mid \mathscr{M}, s \models p\}$. (If Σ is not clear from the context we subscript L with the automaton $\mathscr{A}$ to disambiguate it.) A legal move for the Duplicator

consists of a *marking* $m : \Lambda \times Q \to \wp(\mathrm{SUCC}(s))$ such that $m(a, q') \subseteq \mathrm{SUCC}_a(s)$ for every $(a, q') \in \Lambda \times Q$ and such that there exists a $D \in \delta(q, L(s))$ with:

(a) For all $a \in \Lambda$ and all $t \in \mathrm{SUCC}_a(s)$: there is an $(a, q') \in D$ with $t \in m(a, q')$.

(b) If $(a, q') \in D$ then $m(a, q') \neq \emptyset$.

(3) If the Duplicator has just made a move, namely a marking m, the Spoiler picks a $t \in m(a, q)$ (for some $a \in \Lambda, q \in Q$). The new position becomes (t, q).

Either player wins the game if the other player cannot make a move. An infinite game $(r^{\mathscr{M}}, q_0), m_0, (s_1, q_1), m_1, (s_2, q_2), m_2 \ldots$ is won by the Duplicator if:

$$\min\{\Omega(q) \mid q \text{ appears infinitely often in the sequence}\}$$

is even. We say that $\mathscr{M}$ is *accepted* by $\mathscr{A}$ if and only if there exists a winning strategy for the Duplicator (in such a case we just say that the Duplicator *has* a winning strategy).

If the Duplicator has a winning strategy, he also has a winning strategy in which each time a marking m is chosen and it is correct with respect to some D, then $m(a, q') \neq \emptyset$ implies $(a, q') \in D$. So without changing which structures are accepted by $\mathscr{A}$, we will build this condition into the definition of a correct move for the Duplicator.

Janin and Walukiewicz ([12]) prove that μ-automata correspond precisely to μ-sentences: for every μ-automaton $\mathscr{A}$ there is a μ-sentence $\phi_{\mathscr{A}}$ such that $\mathscr{A}$ accepts $\mathscr{M}$ if and only if $\mathscr{M} \models \phi_{\mathscr{A}}$. Moreover, if $\mathscr{A}$ is a (Σ, Λ)-automaton then the corresponding formula is also in this language. The converse also holds: for every μ-sentence there is an automaton $\mathscr{A}_\phi$ such that $\mathscr{A}_\phi$ accepts $\mathscr{M}$ if and only if $\mathscr{M} \models \phi$. If the set of proposition constants in ϕ is Σ and its set of atomic actions is Λ, then the corresponding automaton may be assumed to be a (Σ, Λ)-automaton.

Our automata differ slightly from those given in [12]: they are more akin to the definition given in [13]. The automata in their various guises are equivalent, however.

μ-automata correspond closely to *disjunctive normal forms* of μ-formulae ([12]), or rather to *tableaux* of such formulae. Perhaps some of the results of this paper may be proved using solely disjunctive formulae, or their tableaux.

Most readers will not be familiar with μ-automata (especially the transition function is nonconventional) so some examples are in order at this point. We only give the automata and the corresponding formulae without proving the equivalence. The reader is encouraged to convince himself of the correspondence.

The first example is a μ-automaton that corresponds to the Propositional Dynamic Logic (PDL) formula $\langle a^* \rangle p$ (in μ-calculus notation: $\mu X.(p \vee \langle a \rangle X)$. Define $\mathscr{A} = (\{q_0, q_1\}, \{p\}, \{a\}, q_0, \delta, \Omega)$ where:

$$\delta(q, L) := \begin{cases} \{\{(a, q_0), (a, q_1)\}\} & \text{if } q = q_0 \text{ and } L = \emptyset; \\ \{\emptyset, \{(a, q_1)\}\} & \text{else,} \end{cases}$$

and the parity function is defined as $\Omega(q_0) = 1, \Omega(q_1) = 0$. Note that the Duplicator can win from any state (s, q_1). So q_1 corresponds to the tautology $\top$.

Our second example is an automaton corresponding to the formula $\nu X.\langle a \rangle X$, which is true at all nodes from which an infinite a-path emerges. Define $\mathscr{B}$ as

$(\{q_0, q_1\}, \emptyset, \{a\}, q_0, \delta, \Omega)$ where:

$$\begin{array}{lcl} \delta(q_0, \emptyset) & := & \{\{(a, q_0), (a, q_1)\}\} \\ \delta(q_1, \emptyset) & := & \{\emptyset, \{(a, q_1)\}\} \end{array}$$

and $\Omega(q_0) = \Omega(q_1) = 0$. Again q_1 corresponds to $\top$.

In general markings may not always be taken disjointly. That is, in general the Duplicator may be forced to choose a marking m such that for some (a, q) and (a', q'), $m(a, q)$ and $m(a', q')$ overlap. We define *labelings* to deal with the special and more manageable cases where it is possible to avoid this overlap.

DEFINITION 2.4. Given a structure $\mathscr{M}$ and a μ-automaton $\mathscr{A}$, an *$\mathscr{A}$-labeling* of $\mathscr{M}$ is a total function $l : \mathscr{M} \to Q$ such that:

(1) $l(r^{\mathscr{M}}) = q_0$;
(2) If $l(s) = q$ then:

$$D(s) := \{(a, q') \in \Lambda \times Q \mid \exists t \in \text{SUCC}_a(s).\, l(t) = q'\} \in \delta(q, L(s)).$$

(3) For any infinite path $r^{\mathscr{M}} = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots$ in $\mathscr{M}$:

$\min\{\Omega(q) \mid \text{there are infinitely many } s_i \text{ in the path such that } l(s_i) = q\}$

is even.

LEMMA 2.5. *If a structure $\mathscr{M}$ has an $\mathscr{A}$-labeling then $\mathscr{A}$ accepts $\mathscr{M}$.*

PROOF. A winning strategy for the Duplicator ensures that the Spoiler may only play positions (s, q) with $l(s) = q$. On such positions, the Duplicator plays the marking m defined by $m(a, q') := \{t \in \text{SUCC}_a(s) \mid l(t) = q'\}$, which is a correct move with respect to $D(s)$. ⊣

The converse of Lemma 2.5 does not hold in general, but it does hold modulo $\Sigma \cup \Lambda$-bisimulation.

THEOREM 2.6. *If $\mathscr{M}$ is accepted by $\mathscr{A}$ then there is a $\Sigma \cup \Lambda$-bisimilar structure $\overline{\mathscr{M}}$ which has an $\mathscr{A}$-labeling.*

PROOF. If $\mathscr{M}$ is accepted by $\mathscr{A}$ then the Duplicator has a winning strategy for the game $\mathscr{G}(\mathscr{A}, \mathscr{M})$. Let $\overline{\mathscr{M}}$ be the following structure:

(1) Its domain consists of all finite initial segments of games played according to the Duplicator's winning strategy, ending in a move by the Spoiler (or in the starting position):

$$(r^{\mathscr{M}}, q_0), m_1, (s_1, q_1), \dots, m_n, (s_n, q_n).$$

(2) p holds at a sequence $\sigma, (s, q)$ if $\mathscr{M}, s \models p$.
(3) For every $a \in \Lambda$, an a-successor of a sequence σ is a sequence $\sigma, m, (s, q)$ with $s \in m(a, q)$.
(4) The root of $\overline{\mathscr{M}}$ is $(r^{\mathscr{M}}, q_0)$, i.e., the starting position of any game.

First we prove that $\mathscr{M}$ is $\Sigma \cup \Lambda$-bisimilar to $\overline{\mathscr{M}}$ via the relation Z that connects any $s \in \mathscr{M}$ precisely to those sequences that end in a position of the form (s, q).

(1) Trivially the roots are connected. The condition for proposition constants is satisfied by definition.

(2) Suppose $a \in \Lambda$, $s \xrightarrow{a} t$ and $sZ\sigma, (s,q)$. Let the Duplicator pick a marking m according to his winning strategy, continuing the game $\sigma, (s,q)$. Then, by definition of a marking, $t \in m(a,q')$ for some $q' \in Q$. So the Spoiler may choose (t,q') as his next move, which makes $\sigma, (s,q), m, (t,q')$ a legitimate element of $\overline{\mathscr{M}}$, which is an a-successor of σ, (s,q) and moreover, is connected to t.

(3) Let a again be an action in Λ. Suppose s is Z-connected to $\sigma, (s,q)$ and let $\sigma, (s,q), m, (t,q')$ be an a-successor of $\sigma, (s,q)$, that is: $t \in m(a,q')$. Then $s \xrightarrow{a} t$ and t is Z-connected to $\sigma, (s,q), m, (t,q')$, so t is the element we are looking for.

Now let l be a function that assigns q to every sequence $\sigma, (s,q)$ in $\overline{\mathscr{M}}$. This constitutes an $\mathscr{A}$-labeling of $\overline{\mathscr{M}}$. The root is labeled with the initial state, and the Ω-condition is trivially satisfied, so we only need to verify the second condition of labelings.

So consider $\sigma = \tau, (s,q)$. The Duplicator's winning strategy gives him a marking m to continue this game on $\mathscr{M}$. Suppose that it is a correct move with respect to $D \in \delta(q, L(s))$. As $t \in m(a,q')$ if and only if $\sigma, m, (t,q')$ is an a-successor of σ, the desired result follows from the condition on correct markings that $(a,q') \in D$ if and only if $m(a,q') \neq \emptyset$. ⊣

When $\mathscr{A}$ is a μ-automaton and $q \in Q$ then $(\mathscr{A}, q)$ is defined to be the automaton that differs from $\mathscr{A}$ only in that q is the initial state. A state q in a μ-automaton $\mathscr{A}$ is *nonempty* if the machine $(\mathscr{A}, q)$ accepts some structure. In logical terms: a state is nonempty if the corresponding μ-sentence is satisfiable. For any automaton $\mathscr{A}$, either the initial state q_0 is empty, or it is equivalent to one all of whose states are nonempty.

§3. **Uniform interpolation.** In this section we prove the uniform interpolation theorem for the modal μ-calculus. It is done via an interpretation of second-order quantifiers within the μ-calculus itself. This approach is inspired by [28] and [20].

Let us begin by stating what we mean by ordinary (Craig) interpolation and uniform interpolation.

DEFINITION 3.1. Let ϕ and ψ be two μ-sentences such that $\phi \models \psi$. Then θ is an *interpolant* of ϕ, ψ if and only if:

(1) $\phi \models \theta \models \psi$;
(2) $\mathscr{L}(\theta) \subseteq \mathscr{L}(\phi) \cap \mathscr{L}(\psi)$. ⊣

In words: if $\phi \models \psi$, an interpolant of ϕ, ψ is a formula in the common language of ϕ and ψ (where by the language of a formula we count both the proposition constants and atomic actions that occur in it) such that $\phi \models \theta \models \psi$.

DEFINITION 3.2. Given a μ-sentence ϕ and a language $\mathscr{L}' \subseteq \mathscr{L}(\phi)$, the *uniform interpolant* of ϕ with respect to $\mathscr{L}'$ is a formula θ such that:

(1) $\phi \models \theta$;
(2) Whenever $\phi \models \psi$ and $\mathscr{L}(\phi) \cap \mathscr{L}(\psi) \subseteq \mathscr{L}'$ then $\theta \models \psi$.
(3) $\mathscr{L}(\theta) \subseteq \mathscr{L}'$. ⊣

When we say that the μ-calculus has (uniform) interpolation we mean that we can always find a (uniform) interpolant when the appropriate conditions are satisfied. Clearly, if the μ-calculus has uniform interpolation, it must also have Craig interpolation. For if $\phi \models \psi$, simply choose $\mathscr{L}' = \mathscr{L}(\phi) \cap \mathscr{L}(\psi)$. The interpolant is then the uniform interpolant of ϕ relative to $\mathscr{L}'$. 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.

We start by proving that the μ-calculus allows existential quantification on proposition constants, modulo bisimulation.

THEOREM 3.3. *Let ϕ be a μ-sentence and $\mathscr{L}$ its language. Let p be an arbitrary proposition constant. Then there is a μ-sentence ψ in the language $\mathscr{L} \setminus \{p\}$ such that:*

$$\mathscr{M} \models \psi \text{ if and only if there is a structure } \mathscr{N} \text{ with } \mathscr{M} \sim_{(\mathscr{L}\setminus\{p\})} \mathscr{N} \models \phi.$$

PROOF. Let $\mathscr{A} = (Q, \Sigma, \Lambda, q_0, \delta, \Omega)$ be an automaton for ϕ, with $\mathscr{L} = \Sigma \cup \Lambda$. Clearly, if $p \notin \Sigma$ then ϕ itself does not contain p, so we may simply choose $\psi := \phi$. What remains is the case that $p \in \Sigma$.

Let $\mathscr{B}$ be the automaton $(Q, \Sigma \setminus \{p\}, \Lambda, q_0, \delta', \Omega)$ with:

$$\delta'(q, L) := \delta(q, L) \cup \delta(q, L \cup \{p\}).$$

This is well-defined precisely because p is assumed to be an element of Σ. $\mathscr{B}$ corresponds to a μ-sentence $\phi_{\mathscr{B}}$ in the language $\mathscr{L} \setminus \{p\}$. We will prove that this formula satisfies the requirements.

Any structure accepted by $\mathscr{A}$ (that is: any structure of ϕ) is also accepted by $\mathscr{B}$. For if the Duplicator has a winning strategy for the $\mathscr{A}$-game on a structure $\mathscr{M}$, that same strategy is winning for the $\mathscr{B}$-game on $\mathscr{M}$. To see this, consider a marking m played on a position (s, q), chosen according to the Duplicator's winning strategy for the $\mathscr{A}$-game. This m is correct with respect to some condition $D \in \delta(q, L_{\mathscr{A}}(s))$. If $\mathscr{M}, s \models p$ then $\delta(q, L_{\mathscr{A}}(s)) = \delta(q, L_{\mathscr{B}}(s) \cup \{p\})$, otherwise $\delta(q, L_{\mathscr{A}}(s)) = \delta(q, L_{\mathscr{B}}(s))$: in both cases $D \in \delta'(q, L_{\mathscr{B}}(s))$. So this condition D is also available when we play the $\mathscr{B}$-game and reach position (s, q). Thus the same marking may be played. Infinite games are no problem, because the Ω-function of $\mathscr{B}$ is the same as that of $\mathscr{A}$.

Next, suppose $\mathscr{B}$ accepts a structure $\mathscr{M}$. We must prove that it is $(\mathscr{L} \setminus \{p\})$-bisimilar with some structure accepted by $\mathscr{A}$.

By Theorem 2.6 there is a structure $\overline{\mathscr{M}}$, $(\mathscr{L} \setminus \{p\})$-bisimilar to $\mathscr{M}$, with a $\mathscr{B}$-labeling l. We define a subset P of the domain of $\overline{\mathscr{M}}$: $s \in P$ if and only if:

$$D(s) := \{(a, q) \in \Lambda \times Q \mid \exists t \in \textsc{succ}_a(s).\, l(t) = q\} \in \delta(l(s), L_{\mathscr{B}}(s) \cup \{p\}).$$

If $(\overline{\mathscr{M}}, P)$ is the structure that differs from $\overline{\mathscr{M}}$ only in that the interpretation of p is P, then $(\overline{\mathscr{M}}, P)$ is still $(\mathscr{L} \setminus \{p\})$-bisimilar to $\mathscr{M}$, but now l is an $\mathscr{A}$-labeling for $(\overline{\mathscr{M}}, P)$. For consider a point s in $\overline{\mathscr{M}}$ with $l(s) = q$. By definition of labelings, $D(s)$ must be in $\delta'(q, L_{\mathscr{B}}(s))$. There are two cases to consider:

(1) $D(s) \in \delta(q, L_{\mathscr{B}}(s) \cup \{p\})$. Then $s \in P$, so $\delta(q, L_{\mathscr{A}}(s)) = \delta(q, L_{\mathscr{B}}(s) \cup \{p\})$.
(2) $D(s) \notin \delta(q, L_{\mathscr{B}}(s) \cup \{p\})$. Then $D(s) \in \delta(q, L_{\mathscr{B}}(s))$, and $s \notin P$ so: $D(s) \in \delta(q, L_{\mathscr{A}}(s)) = \delta(q, L_{\mathscr{B}}(s))$.

The other conditions of labelings are even more trivially satisfied, simply because l is a $\mathscr{B}$-labeling and the initial state and the Ω-function are unchanged. $(\overline{\mathscr{M}}, P)$ is thus accepted by $\mathscr{A}$. ⊣

Remark. There is an alternative proof of this fact, using the notion of ω-expansion and a related lemma by Janin and Walukiewicz ([13]). The ω-expansion of a structure $\mathscr{M}$ is the usual unraveling of $\mathscr{M}$ into a tree-like structure, but with at least ω many copies of each successor. The ω-expansion of a structure is a bisimilar to the original. Janin and Walukiewicz ([13]) prove that for every MSO sentence ϕ there is a μ-sentence $\mu(\phi)$ such that ϕ holds in the ω-expansion of some structure $\mathscr{M}$ if and only if $\mu(\phi)$ holds in $\mathscr{M}$ itself. Now ψ in the above lemma must be equivalent to $\mu(\exists p.\phi)$ ($\exists p.\phi$ is an MSO sentence as μ-sentences are MSO-expressible).

Corollary 3.4. *For any μ-sentence ϕ and every proposition constant p there is a μ-sentence $\widetilde{\exists} p.\phi$ such that*:

(1) $\phi \models \widetilde{\exists} p.\phi$;
(2) $\mathscr{L}(\widetilde{\exists} p.\phi) = \mathscr{L}(\phi) \setminus \{p\}$;
(3) *If $\phi \models \theta$ and $p \notin \mathscr{L}(\theta)$ then $\widetilde{\exists} p.\phi \models \theta$.*

Note that the three conditions in the corollary state that $\widetilde{\exists} p.\phi$ is the uniform interpolant of ϕ with respect to ϕ's language without p. The reason for the existential notation is that the uniform interpolant behaves like a second-order quantification of ϕ, as the conditions demonstrate: item 1 is like $\exists$-introduction and item 3 is like $\exists$-elimination in natural deduction.

To prove the corollary we need the following result:

Lemma 3.5. *Let $\mathscr{M}$, $\mathscr{N}$ be structures for the languages $\mathscr{L}$ and $\mathscr{L}'$, respectively. If $\mathscr{M} \sim_{\mathscr{L}\cap\mathscr{L}'} \mathscr{N}$, then there is an $(\mathscr{L} \cup \mathscr{L}')$-structure $\mathscr{K}$ which is $\mathscr{L}$-bisimilar to $\mathscr{M}$ and $\mathscr{L}'$- bisimilar to $\mathscr{N}$.*

Proof. Fix an $\mathscr{L} \cap \mathscr{L}'$-bisimulation Z between $\mathscr{M}$ and $\mathscr{N}$. $\mathscr{K}$ is defined as follows.

(1) The domain is the disjoint union of $\{(m, n) \in \mathscr{M} \times \mathscr{N} : mZn\}$, $\mathscr{M}$, and $\mathscr{N}$; the root is $r^{\mathscr{K}} := (r^{\mathscr{M}}, r^{\mathscr{N}})$.

(2) If $k, k' \in \mathscr{M}$ and a is an action in $\mathscr{L}$, then $kR_a^{\mathscr{K}}k'$ if and only if $kR_a^{\mathscr{M}}k'$.

If $k, k' \in \mathscr{N}$ and a is an action in $\mathscr{L}'$, then $kR_a^{\mathscr{K}}k'$ if and only if $kR_a^{\mathscr{N}}k'$.

If a is an action in $\mathscr{L} \cap \mathscr{L}'$, then $(m, n)R_a^{\mathscr{K}}k$ if and only if $k = (m', n')$, $mR_a^{\mathscr{M}}m'$, and $nR_a^{\mathscr{N}}n'$.

If a is an action in $(\mathscr{L} \setminus \mathscr{L}')$ then $(m, n)R_a^{\mathscr{K}}k$ if and only if $k \in \mathscr{M}$ and $mR_a^{\mathscr{M}}k$.

If a is an action in $(\mathscr{L}' \setminus \mathscr{L})$ then $(m, n)R_a^{\mathscr{K}}k$ if and only if $k \in \mathscr{N}$ and $nR_a^{\mathscr{N}}k$.

(3) For any $p \in \mathscr{L} \cup \mathscr{L}'$ define: $\mathscr{K}, (m, n) \models p$ if and only if $p \in \mathscr{L}$ and $\mathscr{M}, m \models p$ or $p \in \mathscr{L}'$ and $\mathscr{N}, n \models p$. Note that this is well-defined, because when $p \in \mathscr{L} \cap \mathscr{L}'$ then m and n must agree on p, since they are $\mathscr{L} \cap \mathscr{L}'$-bisimilar. If $k \in \mathscr{M}$ ($k \in \mathscr{N}$) then $\mathscr{K}, k \models p$ if and only if $\mathscr{M}, k \models p$ ($\mathscr{N}, k \models p$, respectively).

We claim that $\mathscr{K}$ is $\mathscr{L}$-bisimilar to $\mathscr{M}$ via the relation $S = \{(m, m) : m \in \mathscr{M}\} \cup \{((m, n), m) : (m, n) \in \mathscr{K}\}$, and $\mathscr{L}'$-bisimilar to $\mathscr{N}$ via the relation $T = \{(n, n) : n \in \mathscr{N}\} \cup \{((m, n), n) : (m, n) \in \mathscr{K}\}$. By symmetry we only need to prove the claim for the relation S.

(1) The roots are connected by definition. If kSm and $k = m \in \mathscr{M}$ then k and m satisfy the same proposition constants by definition of the valuation in $\mathscr{K}$. The same holds if $k = (m, n)$, because then m and n are $\mathscr{L} \cap \mathscr{L}'$-bisimilar and thus satisfy the same proposition constants in $\mathscr{L} \cap \mathscr{L}'$.

(2) Suppose that kSm. If $k \in \mathscr{M}$ then $k = m$, and the back and forth conditions for an $\mathscr{L}$-bisimulation are trivially satisfied. If $k = (m, n)$ with mZn, the back and forth conditions for an action a in $\mathscr{L} \setminus \mathscr{L}'$ are satisfied because $(m, n)R_a^{\mathscr{K}} k'$ if and only if $k' \in \mathscr{M}$ and $mR_a^{\mathscr{M}} k'$. Next, consider $a \in \mathscr{L} \cap \mathscr{L}'$. If $(m, n)R_a^{\mathscr{K}} k'$, then $k' = (m', n')$ with $mR_a^{\mathscr{M}} m'$ and $k'Sm'$. Vice versa, if $mR_a^{\mathscr{M}} m'$ then since Z is a $\mathscr{L} \cap \mathscr{L}'$-bisimulation and mZn, there exists $n' \in \mathscr{N}$ such that $nR_a^{\mathscr{N}} n'$ and $m'Zn'$. Then $(m', n') \in \mathscr{K}$, $(m, n)R_a^{\mathscr{K}} (m', n')$, and $(m', n')Sm'$. ⊣

PROOF OF COROLLARY 3.4. We claim that the sentence ψ constructed from ϕ as in Theorem 3.3 enjoys the desired properties. It is clear that $\phi \models \psi$ and $\mathscr{L}(\psi) = \mathscr{L}(\phi) \setminus \{p\}$. Let θ be such that $\phi \models \theta$ and $p \notin \mathscr{L}(\theta)$. We want to prove that for any $(\mathscr{L}(\theta) \cup \mathscr{L}(\psi))$-structure $\mathscr{M}$, if $\mathscr{M} \models \psi$ then $\mathscr{M} \models \theta$.

Let $\mathscr{L}(\theta) \cup \mathscr{L}(\psi) = \mathscr{L}$, $\mathscr{L}(\phi) = \mathscr{L}'$, and let $\mathscr{M}$ be an $\mathscr{L}$-structure such that $\mathscr{M} \models \psi$. If we consider $\mathscr{M}$ as an $\mathscr{L}(\psi)$-structure, then by the property of ψ we know that there exists an $\mathscr{L}'$-structure $\mathscr{N}$ which is $\mathscr{L}(\psi)$-bisimilar to $\mathscr{M}$ and satisfies the formula ϕ. Since $\mathscr{L} \cap \mathscr{L}' = \mathscr{L}(\psi)$, we can apply Lemma 3.5 to get an $(\mathscr{L} \cup \mathscr{L}')$-structure $\mathscr{K}$ which is $\mathscr{L}$-bisimilar to $\mathscr{M}$ and $\mathscr{L}'$-bisimilar to $\mathscr{N}$. From $\mathscr{K} \sim_{\mathscr{L}'} \mathscr{N}$ it follows $\mathscr{K} \models \phi$, from $\phi \models \theta$ it follows $\mathscr{K} \models \theta$, and from $\mathscr{M} \sim_{\mathscr{L}} \mathscr{K}$ we obtain $\mathscr{M} \models \theta$. ⊣

REMARK. Note that Theorem 3.3 gives us a satisfaction condition for $\widetilde{\exists} p.\phi$: $\mathscr{M} \models \widetilde{\exists} p.\phi$ if and only if there is some structure $\mathscr{N}$ bisimilar to $\mathscr{M}$ with respect to all symbols in ϕ except p such that $\mathscr{N} \models \phi$. So we may call $\widetilde{\exists}$ a *bisimulation quantifier*. In some cases the bisimulation quantifier may be read as a classical second-order existential quantifier, for instance if ϕ is monotone in p. In fact, in the latter case, $\widetilde{\exists} p.\phi$ is equivalent to $\phi[p := \top]$.

We give a few more examples to illustrate the bisimulation quantifier.

$\widetilde{\exists} p.\langle a\rangle p \wedge \langle a\rangle \neg p$ is equivalent to the formula $\langle a\rangle \top$. So in this case the bisimulation quantifier differs from the standard second-order quantifier, as $\exists p.\langle a\rangle p \wedge \langle a\rangle \neg p$ expresses that there are at least *two* a-successors. It is not surprising that prefixing a modal formula (that is, one without fixed point operators) with a bisimulation quantifier gives us a formula equivalent to a modal formula. This follows from the proof of uniform interpolation for the minimal modal logic K (see [28]).

Let $\phi(p) = p \wedge \nu X.((p \to \langle a\rangle p) \wedge [a]X)$. In PDL-notation, this formula may be written as: $p \wedge [a^*](p \to \langle a\rangle p)$. Then $\widetilde{\exists} p.\phi(p)$ coincides with $\exists p.\phi(p)$: it expresses the property of nonwellfoundedness: it is true at points from which an infinite a-path emanates. In the modal μ-calculus this may be expressed as $\nu X.\langle a\rangle X$.

The proof that PDL does not have uniform interpolation consists of proving that the uniform interpolant of $\phi(p)$ (with respect to $\{a\}$) must be equivalent to $\nu X.\langle a\rangle X$, which cannot be expressed in PDL: it cannot even be expressed in $\mathscr{L}_{\infty\omega}$, into which PDL can be embedded. Whether PDL has ordinary Craig interpolation is still considered an open problem.

We turn our attention to atomic actions.

THEOREM 3.6. *Let ϕ be a μ-sentence with $\mathscr{L}(\phi) = \mathscr{L}$. Let a be an atomic action. Then there is a μ-sentence ψ with $\mathscr{L}(\psi) = \mathscr{L} \setminus \{a\}$ such that:*

$$\mathscr{M} \models \psi \text{ if and only if there is a structure } \mathscr{N} \text{ with } \mathscr{M} \sim_{(\mathscr{L}\setminus\{a\})} \mathscr{N} \models \phi.$$

PROOF. Let $\mathscr{A} = (Q, \Sigma, \Lambda, q_0, \delta, \Omega)$ be an automaton for ϕ, with $\mathscr{L} = \Sigma \cup \Lambda$. If ϕ is unsatisfiable, we may take $\psi := \bot$. If ϕ is satisfiable, we may assume that Q does not contain any empty states. So we may choose, for each $q \in Q$, a structure $\mathscr{M}_q$ with an $\mathscr{A}$-labeling l_q.

Now define the automaton $\mathscr{B}$ as $(Q, \Sigma, \Lambda \setminus \{a\}, q_0, \delta', \Omega)$ where:

$$\delta'(q, L) := \{D \cap ((\Lambda \setminus \{a\}) \times Q) \mid D \in \delta(q, L)\}.$$

We prove that any structure accepted by $\mathscr{A}$ is also accepted by $\mathscr{B}$. For suppose the Duplicator has a winning strategy for the $\mathscr{A}$-game on $\mathscr{M}$. We can transform this into a strategy for the $\mathscr{B}$-game. Let the Duplicator play two games at once, an $\mathscr{A}$- and a $\mathscr{B}$-game on $\mathscr{M}$. We control the Spoiler in the $\mathscr{A}$-game, but we have no control over him in the $\mathscr{B}$-game. The Duplicator plays the $\mathscr{A}$-game according to his winning strategy. After $2n$ moves (not counting the initial position as a move) both the $\mathscr{A}$-game and the $\mathscr{B}$-game are in the same position. This is clearly true if no moves have been made: then both games are at the same initial position (the initial state of $\mathscr{B}$ is the same as that of $\mathscr{A}$). If we have reached a position (s, q) in both games, let the Duplicator pick a marking m in the $\mathscr{A}$-game according to his winning strategy. This marking is correct with respect to some $D \in \delta(q, L(s))$. We then let the Duplicator pick $m' := m \upharpoonright ((\Lambda \setminus \{a\}) \times Q)$ as his next move in the $\mathscr{B}$-game. This marking is correct with respect to $D' := D \cap ((\Lambda \setminus \{a\}) \times Q) \in \delta'(q, L(s))$. Now the Spoiler must choose some $(b, q) \in D'$ and some $t \in m'(b, q)$ and play (t, q) in the $\mathscr{B}$-game. If he cannot pick such a position, the Duplicator wins the $\mathscr{B}$-game. If he *can*, we let the Spoiler pick the same position in the $\mathscr{A}$-game. This gives us a strategy for the $\mathscr{B}$-game. It is a winning strategy, because any infinite game comes across the same positions as some $\mathscr{A}$-game played according to the winning strategy of the Duplicator for the $\mathscr{A}$-game.

Now for the converse. Suppose $\mathscr{M}$ is accepted by $\mathscr{B}$. Again, we may unravel $\mathscr{M}$ to $\overline{\mathscr{M}}$ such that this structure is $(\mathscr{L} \setminus \{a\})$-bisimilar to $\mathscr{M}$ and such that it carries a $\mathscr{B}$-labeling l. We define a mapping $d : \overline{\mathscr{M}} \to \wp(\Lambda \times Q)$, as follows. By definition of labelings, for every s in the domain of $\overline{\mathscr{M}}$:

$$D(s) := \{(b, q) \in (\Lambda \setminus \{a\}) \times Q \mid \exists t \in \mathrm{SUCC}_b(s).\, l(t) = q\} \in \delta'(l(s), L(s)).$$

So there must be some $D \in \delta(l(s), L(s))$ with $D \cap ((\Lambda \setminus \{a\}) \times Q) = D(s)$. Choose such a D arbitrarily (there may be more than one) and fix $d(s) := D$.

Let $\mathscr{N}$ be the disjoint union of $\overline{\mathscr{M}}$ and $\mathscr{M}_q$ for all $q \in Q$, with the root of $\overline{\mathscr{M}}$ as its root and with some new a-transitions, namely an a-transition from s to the root of $\mathscr{M}_q$ is added for every $s \in \overline{\mathscr{M}}$, $q \in Q$ such that $(a, q) \in d(s)$. Clearly the embedding of $\overline{\mathscr{M}}$ into $\mathscr{N}$ is an $(\mathscr{L} \setminus \{a\})$-bisimulation. Furthermore, $l \cup \bigcup_{q \in Q} l_q$ is an $\mathscr{A}$-labeling of $\mathscr{N}$, so $\mathscr{N}$ is accepted by $\mathscr{A}$. ⊣

COROLLARY 3.7. *For any μ-sentence ϕ and every atomic action $a \in A$ there is a formula $\widetilde{\exists} a.\phi$ such that:*

(1) $\phi \models \widetilde{\exists} a.\phi$;

(2) $\mathscr{L}(\widetilde{\exists}a.\phi) = \mathscr{L}(\phi) \setminus \{a\}$;
(3) *If* $\phi \models \theta$ *and* $a \notin \mathscr{L}(\theta)$ *then* $\widetilde{\exists}a.\phi \models \theta$.

PROOF. Similar to the proof of Corollary 3.4, again using Lemma 3.5. ⊣

COROLLARY 3.8. *The modal μ-calculus enjoys uniform interpolation.*

PROOF. Let ϕ be a μ-sentence. Fix some $\mathscr{L}' \subseteq \mathscr{L}(\phi)$. Suppose

$$\mathscr{L}(\phi) \setminus \mathscr{L}' = \{p_1, \dots, p_n, a_1, \dots, a_m\}.$$

Then the uniform interpolant is: $\widetilde{\exists}p_1 \dots \widetilde{\exists}p_n.\widetilde{\exists}a_1 \dots \widetilde{\exists}a_m.\phi$. ⊣

Bisimulation-quantifiers may also be studied in their own right, of course. We present here one particular application: adding the bisimulation-quantifiers $\widetilde{\exists}p.\phi$ to PDL results in the μ-calculus itself.

Define BQL (Bisimulation-Quantifier Logic) as follows:

$$\begin{array}{lcll} \textbf{Propositions:} & \phi & ::= & p \mid \phi \vee \phi \mid \neg\phi \mid \langle\pi\rangle\phi \mid \widetilde{\exists}p\phi \\ \textbf{Programs:} & \pi & ::= & a \mid \phi? \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \end{array}$$

where $p \in$ PROP and $a \in A$.

BQL is simply the extension of PDL with bisimulation-quantifiers, which are interpreted as in the remark following the proof of Corollary 3.4. For the reader not familiar with PDL, we just mention that it is a formalism where diamonds not only range over atomic actions, but may be labeled with arbitrary programs, which have a regular structure.

THEOREM 3.9. *BQL is equivalent to the μ-calculus. That is: for any BQL-proposition there is an equivalent μ-sentence and vice versa.*

PROOF. Both ways, the proof consists of a translation from one formalism into the other, preserving meaning. We assume that we are working in a μ-calculus that has arbitrary negations, with the restriction that μX-operators are only applied to formulae ϕ where X only occurs positively in ϕ. This formulation yields the same formulae as the formulation given in Definition 2.1.

From BQL to the μ-calculus, the translation works as follows. Consider a BQL-proposition ϕ. First, we replace all complex diamonds by simpler ones, as follows:

$$\begin{array}{lcl} \langle\phi?\rangle\psi & \mapsto & \phi \wedge \psi \\ \langle\pi_1;\pi_2\rangle\phi & \mapsto & \langle\pi_1\rangle\langle\pi_2\rangle\phi \\ \langle\pi_1 \cup \pi_2\rangle\phi & \mapsto & \langle\pi_1\rangle\phi \vee \langle\pi_2\rangle\phi \\ \langle\pi^*\rangle\phi & \mapsto & \mu X.(\phi \vee \langle\pi\rangle X). \end{array}$$

The translation of formulae of the form $\langle\pi\rangle\phi$ is such that if a variable X is positive in $\langle\pi\rangle\phi$ then X must also be positive in its translation. This ensures that μX-operators are only applied to formulae with just positive occurrences of ϕ.

This gives us a μ-sentence possibly still containing bisimulation-quantifiers. The idea is to use Theorem 3.3 to replace these with actual μ-calculus formulae, from the inside out. Theorem 3.3 only deals with sentences, whereas here we apply it to formulae possibly containing free variables. This is not a real problem, as Theorem 3.3 can be applied if we simply regard free variables as proposition constants. A more serious problem, in principle, is where $\widetilde{\exists}p.\phi(p, X)$, where X occurs only positively in this formula is replaced by an equivalent formula, but with a negative occurrence of X. However, the procedure for removing complex diamonds ensures that no free μ-variable ever occurs within the scope of a bisimulation-quantifier, so there is no real problem.

This proves that any BQL-formula is equivalent to a μ-calculus sentence. Next, we prove the converse. This is achieved by means of a direct translation. The only interesting case is where we wish to translate a μ-formula of the form: $\mu X.\phi(X)$, where $\phi(X)$ has been shown to be equivalent to the BQL-formula $\phi^\circ(X)$. We will show that $\mu X.\phi(X)$ is then equivalent to the formula $\neg\tilde{\exists}p.(\Box^*(\phi^\circ(p) \to p) \wedge \neg p)$, where $\Box^*$ is defined as $[(a_1 \cup \cdots \cup a_n)^*]$ and $a_1, \ldots, a_n$ are the atomic actions in $\phi(X)$.

First, suppose $\mathscr{M}, s \models \neg\tilde{\exists}p.(\Box^*(\phi^\circ(p) \to p) \wedge \neg p)$. Let P be the least fixed point of $\phi(X)$ in $\mathscr{M}$. Change the interpretation of p in $\mathscr{M}$ to P. This gives us a structure $\mathscr{N}$, such that $\mathscr{M}, s \sim_{\mathscr{L}\setminus\{p\}} \mathscr{N}, s$. Thus $\mathscr{N}, s \not\models \Box^*(\phi^\circ(p) \to p) \wedge \neg p$. Clearly s satisfies the first conjunct in $\mathscr{N}$ (because P is also the least fixed point in $\mathscr{N}$) so the second conjunct must be false, i.e., $s \in P$, so $\mathscr{M}, s \models \mu X.\phi(X)$.

Now, for the other direction, suppose $\mathscr{M}, s \models \tilde{\exists}p.(\Box^*(\phi^\circ(p) \to p) \wedge \neg p)$, say $\mathscr{N}, t \models \Box^*(\phi^\circ(p) \to p) \wedge \neg p$, with $\mathscr{M}, s \sim_{\mathscr{L}\setminus\{p\}} \mathscr{N}, t$. Let $\mathscr{K}$ be the substructure of $\mathscr{N}$ generated via actions $a_1, \ldots, a_n$ from t. Then in $\mathscr{K}$, $p^{\mathscr{K}}$ is a pre-fixed point of $\phi(X)$ and $t \notin p^{\mathscr{K}}$. Thus $\mathscr{K}, t \not\models \mu X.\phi(X)$, hence $\mathscr{M}, s \not\models \mu X.\phi(X)$. ⊣

§4. **The Lyndon Theorem.** This section is devoted to the proof of Theorem 4.9, the Lyndon Theorem for the μ-calculus, which states that positivity corresponds modulo logical equivalence to monotonicity. We begin by explaining these notions.

DEFINITION 4.1. Let p be a proposition constant. A μ-sentence is *positive* in p if it does not contain literals of the form $\neg p$.

Alternatively, if we allow arbitrary negations, a formula ϕ is positive in p if and only if all occurrences of p in ϕ occur under an even number of negations. The alternative definition of the μ-calculus actually uses this notion to restrict applications of the μ-operator: $\mu X.\phi$ is only grammatical if ϕ is positive in X.

DEFINITION 4.2. A μ-sentence ϕ is *monotone* in the proposition constant p if for all $\mathscr{M}$ and all subsets A of the domain, if $p^{\mathscr{M}} \subseteq A$ and $\mathscr{M} \models \phi$ then $\mathscr{M}[p := A] \models \phi$.

Positive sentences are monotone. The Lyndon Theorem establishes the converse, that is, any monotone μ-sentence is equivalent to a positive one. To prove the Lyndon Theorem, we briefly describe, following Janin's thesis ([11]), how to find a μ-sentence corresponding to a given automaton $\mathscr{A}$.

DEFINITION 4.3. An automaton $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ is said to be in a *tree-like* form if there exists a partial ordering $\leq_{\mathscr{A}}$ of the set of states Q such that:

- q_0 is the minimum with regard to $\leq_{\mathscr{A}}$;
- if $q_2 \leq_{\mathscr{A}} q_1$ and $q_3 \leq_{\mathscr{A}} q_1$ then q_2, q_3 are comparable;
- if $(a, q_2) \in D \in \delta(q_1, S)$ then either q_2 is an immediate successor of q_1 (that is $q_1 <_{\mathscr{A}} q_2$ and there is no $q \in Q$ such that $q_1 <_{\mathscr{A}} q <_{\mathscr{A}} q_2$), or $q_2 \leq_{\mathscr{A}} q_1$ (in this case q_2 is called a back-state and q_1 its root)
- if q_2 is a back-state and q_1 a root of q_2, then for all q, if $q_2 \leq_{\mathscr{A}} q \leq_{\mathscr{A}} q_1$ then $\Omega(q_2) \leq \Omega(q)$.

When an automaton is in tree-like form, the μ-sentence which is equivalent to it can be constructed inductively on the tree-like structure of the set Q:

DEFINITION 4.4. If $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ is in tree-like form and $q \in Q$, the μ-formula α_q is defined as follows:

$$(\sigma_q X_q) \bigvee_{S \subseteq \Sigma} \bigvee_{D \in \delta(q.S)} \left(\alpha_S \wedge \bigwedge_{a \in \Lambda} \left(\left(\bigwedge_{(a.q_1) \in D} \langle a \rangle \beta_{q.q_1} \right) \wedge [a] \left(\bigvee_{(a.q_1) \in D} \beta_{q.q_1} \right) \right) \right),$$

where

(1) if $S \subseteq \Sigma$ then α_S is the formula $\bigwedge_{p \in S} p \wedge \bigwedge_{p \in \Sigma \setminus S} \neg p$
(2) $\sigma_q = \nu$ if $\Omega(q)$ is even, $\sigma_q = \mu$ if $\Omega(q)$ is odd;
(3) $\beta_{q.q_1} = X_{q_1}$ if q_1 is a back state and q is its root (in this case $q_1 \leq_A q$), and $\beta_{q.q_1} = \alpha_{q_1}$ otherwise (in this case $q <_A q_1$ and α_{q_1} is already defined by induction).

For an automaton $\mathscr{B}$ in tree-like form with initial state q_0 Janin proves that the sentence $\phi_{\mathscr{B}} = \alpha_{q_0}$ is equivalent to $\mathscr{B}$ ([11]). Moreover, for any automaton $\mathscr{A}$, there exists an automaton $\mathscr{A}_T$ in tree-like form which recognizes the same class of structures, so that the sentence $\phi_{\mathscr{A}_T}$ is equivalent to $\mathscr{A}$.

To get familiar with this construction we find the sentence α_{q_0} corresponding to our first example of an automaton:

$\mathscr{A} = (\{q_0, q_1\}, \{p\}, \{a\}, q_0, \delta, \Omega)$ where:

$$\delta(q, L) := \begin{cases} \{\{(a, q_0), (a, q_1)\}\} & \text{if } q = q_0 \text{ and } L = \emptyset; \\ \{\emptyset, \{(a, q_1)\}\} & \text{else,} \end{cases}$$

and the parity function is defined as $\Omega(q_0) = 1$, $\Omega(q_1) = 0$. We know that this automaton accepts exactly the structures in which $\mu X.(p \vee \langle a \rangle X)$ is true, and we now want to obtain this sentence by computing α_{q_0}. Notice that the automaton $\mathscr{A}$ is in tree-like form with respect to the partial order $\leq_A = \{(q_0, q_0), (q_0, q_1), (q_1, q_1)\}$.

We first calculate $\alpha_{q_1} = \nu_{q_1} X_{q_1}.(\phi_1 \vee \phi_2)$, where

$$\phi_1 = \bigvee_{D \in \delta(q_1.\emptyset)} \left(\neg p \wedge \left(\bigwedge_{(a.q) \in D} \langle a \rangle \beta_{q_1.q} \wedge [a] \left(\bigvee_{(a.q) \in D} \beta_{q_1.q} \right) \right) \right),$$

and

$$\phi_2 = \bigvee_{D \in \delta(q_1.\{p\})} \left(p \wedge \left(\bigwedge_{(a.q) \in D} \langle a \rangle \beta_{q_1.q} \wedge [a] \left(\bigvee_{(a.q) \in D} \beta_{q_1.q} \right) \right) \right).$$

Since $\delta(q_1, \emptyset) = \delta(q_1, \{p\})$, p and $\neg p$ eliminate each other in α_{q_1}, and α_{q_1} is equivalent to

$$\nu_{q_1} X_{q_1}. \bigvee_{D \in \delta(q_1.\{p\})} \left(\bigwedge_{(a.q_1) \in D} \langle a \rangle \beta_{q.q_1} \wedge [a] \left(\bigvee_{(a.q_1) \in D} \beta_{q.q_1} \right) \right).$$

From $\delta(q_1, \{p\}) = \{\emptyset, \{(a, q_1)\}\}$ it is then easy to see that α_{q_1} is equivalent to $\nu_{q_1} X_{q_1}.([a]\bot \vee (\langle a \rangle X_{q_1} \wedge [a] X_{q_1}))$, which is in turn equivalent to $\top$.

We can now calculate $\alpha_{q_0} = \mu_{q_0} X_{q_0}.(\psi_1 \vee \psi_2)$, where

$$\psi_1 = (\neg p \wedge (\langle a\rangle\alpha_{q_1} \wedge \langle a\rangle X_{q_0} \wedge [a](\alpha_{q_1} \vee X_{q_0}))),$$

and

$$\psi_2 = (p \wedge ([a]\bot \vee (\langle a\rangle\alpha_{q_1} \wedge [a](\alpha_{q_1})))).$$

Since α_{q_1} is equivalent to $\top$, we easily obtain that α_{q_0} is equivalent to $\mu X_{q_0}(p \vee \langle a\rangle X_{q_0})$.

In the following definition we will isolate a property of an automaton $\mathscr{A}$ in tree-like form which will be proved in Lemmas 4.6 and 4.8 to correspond to the monotonicity of the sentence $\phi_{\mathscr{A}}$.

DEFINITION 4.5. An automaton $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ is called *monotone* in the proposition constant p if for all $q \in Q$, and $S \subseteq \Sigma$, it is true that

$$\delta(q, S) \subseteq \delta(q, S \cup \{p\}).$$

LEMMA 4.6. *If the automaton $\mathscr{A}$ is in tree-like form and monotone in p, then the μ-sentence $\phi_{\mathscr{A}}$ is positive in p.*

PROOF. Let $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ be in tree-like form and monotone in p, and let $\alpha_S, \beta_{q,q_1}, \alpha_q$ be the μ-formulae defined as in 4.4. For $S \subseteq \Sigma$, $q \in Q$, and $D \in \delta(q, S)$, define $\beta_{q,D} = \bigwedge_{a\in\Lambda}(\bigwedge_{(a,q_1)\in D}\langle a\rangle\beta_{q,q_1}) \wedge [a](\bigvee_{(a,q_1)\in D}\beta_{q,q_1})$. Then

$$\alpha_q = \sigma_q X_q \bigvee_{S\subseteq\Sigma}\ \bigvee_{D\in\delta(q,S)} (\alpha_S \wedge \beta_{q,D}).$$

Notice that the literal $\neg p$ appears in α_q, at least in the subformulae α_S if $p \notin S$. We first use the monotonicity of the automaton to show how to remove these occurrences of $\neg p$.

First of all, the formula α_q may be written as

$$\sigma_q X_q. \bigvee_{S\subseteq\Sigma\setminus\{p\}} \left(\bigvee_{D\in\delta(q,S)} (\alpha_S \wedge \beta_{q,D}) \vee \bigvee_{D\in\delta(q,S\cup\{p\})} (\alpha_{S\cup\{p\}} \wedge \beta_{q,D}) \right).$$

For $S \subseteq \Sigma \setminus \{p\}$ and $q \in Q$, denote by α_S^p the formula $\alpha_{S\cup\{p\}}$, and consider the sets $X_q^S = \delta(q, S \cup \{p\})$, $Y_q^S = \delta(q, S)$, and $Z_q^S = X_q^S \setminus Y_q^S$. By the monotonicity of $\mathscr{A}$ we have that $X_q^S = Y_q^S \cup Z_q^S$, and α_q is equivalent to:

$$\sigma_q X_q. \bigvee_{S\subseteq\Sigma\setminus\{p\}} \left(\bigvee_{D\in Y_q^S} (\alpha_S \wedge \beta_{q,D}) \vee \bigvee_{D\in Y_q^S} (\alpha_S^p \wedge \beta_{q,D}) \vee \bigvee_{D\in Z_q^S} (\alpha_S^p \wedge \beta_{q,D}) \right)$$

$$\equiv \sigma_q X_q. \bigvee_{S\subseteq\Sigma\setminus\{p\}} \left[\left(\bigvee_{D\in Y_q^S} (\alpha_S \wedge \beta_{q,D}) \vee (\alpha_S^p \wedge \beta_{q,D}) \right) \vee \bigvee_{D\in Z_q^S} (\alpha_S^p \wedge \beta_{q,D}) \right].$$

If α_S^{-p} is the formula defined as:

$$\alpha_S^{-p} = \bigwedge_{r\in S} r \wedge \bigwedge_{r\in\Sigma\setminus S, r\neq p} \neg r,$$

then for $S \subseteq \Sigma \setminus \{p\}$: α_S^{-p} is positive in p, $\alpha_S = \alpha_S^{-p} \wedge \neg p$, $\alpha_S^p = \alpha_S^{-p} \wedge p$, and the formula α_q is equivalent to

$$\sigma_q X_q. \bigvee_{S \subseteq \Sigma \setminus \{p\}} \left[\left(\bigvee_{D \in Y_q^S} (\alpha_S^{-p} \wedge \neg p \wedge \beta_{q.D}) \vee (\alpha_S^{-p} \wedge p \wedge \beta_{q.D}) \right) \vee \bigvee_{D \in Z_q^S} (\alpha_S^p \wedge \beta_{q.D}) \right],$$

which is in turn equivalent to

$$\sigma_q X_q. \bigvee_{S \subseteq \Sigma \setminus \{p\}} \left[\left(\bigvee_{D \in Y_q^S} (\alpha_S^{-p} \wedge \beta_{q.D}) \right) \vee \left(\bigvee_{D \in Z_q^S} (\alpha_S^p \wedge \beta_{q.D}) \right) \right].$$

In this way all occurrences of $\neg p$ in subformulae of type α_Σ with $p \notin \Sigma$ have been removed.

We now prove by induction on the tree-like structure of $\mathscr{A}$ that all α_q are equivalent to positive formulae. By performing the transformation above, it suffices to prove that all $\beta_{q,D}$ are positive in p. But the formula $\beta_{q,D}$ is equal to $\bigwedge_{a \in \Sigma_r} (\bigwedge_{(a,q_1) \in D} \langle a \rangle \beta_{q,q_1}) \wedge [a](\bigvee_{(a,q_1) \in D} \beta_{q,q_1})$, and all possible subformulae β_{q,q_1} of $\beta_{q.D}$ are either variables (only variables if q is a leaf), or formulae of type α_{q_1}, for $q <_A q_1$. By induction these are positive. ⊣

We will prove the converse of Lemma 4.6 and then the Lyndon Theorem by using the (uniform) interpolation theorem for the μ-calculus (Corollary 3.8).

In the following, we denote by $\Box^*(s \rightarrow p)$ the μ-sentence $\nu X.((s \rightarrow p) \wedge \bigvee_{a \in \Lambda} [a] X)$ which is true in $\mathscr{M}$ if and only if for all $m \in \mathscr{M}$ reachable from the root via atomic transitions labeled with elements of Λ, it holds that $\mathscr{M}, m \models s \rightarrow p$.

The following lemma is a special case of the Lyndon Theorem and will be used to prove the full theorem (Theorem 4.9).

LEMMA 4.7. *Let $\phi(p)$ be a μ-sentence, whose atomic actions occur among Λ. If s is a proposition constant not appearing in $\phi(p)$ then both the sentence $\psi = \Box^*(s \rightarrow p) \wedge \phi(s)$ and its uniform interpolant with respect to $\mathscr{L}(\psi) \setminus \{s\}$ have an equivalent automaton which is in tree-like form and monotone in p.*

PROOF. We first prove that there exists an automaton monotone in p corresponding to the sentence $\psi = \Box^*(s \rightarrow p) \wedge \phi(s)$. If Σ is the set of proposition constants appearing in $\phi(s)$, then there exists an automaton $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ in tree-like form which is equivalent to $\phi(s)$. Consider the following automaton $\mathscr{A}^* = (Q, q_0, \Sigma \cup \{p\}, \Lambda, \delta^*, \Omega)$, where $\delta^*(q, S)$, for $q \in Q$ and $S \subseteq \Sigma \cup \{p\}$, is defined as follows: if $s \notin S$ or $p \in S$ then $\delta^*(q, S) = \delta(q, S \setminus \{p\})$, and $\delta^*(q, S) = \emptyset$ otherwise. It is clear that $\mathscr{A}^*$ is monotone in p and in tree-like form. We now prove that it is equivalent to ψ, that is, for any structure $\mathscr{M}$, $\mathscr{M}$ is accepted by $\mathscr{A}^*$ if and only if $\mathscr{M} \models \psi$. Without loss of generality, we may assume that $\mathscr{M}$ is generated from its root by means of the atomic actions in Λ.

If $\mathscr{M} \models \psi$ then for any $m \in \mathscr{M}$ we have either $s \notin L_{\mathscr{A}^*}(m)$ or $p \in L_{\mathscr{A}^*}(m)$, hence for any $q \in Q$ it holds that $\delta^*(q, L_{\mathscr{A}^*}(m)) = \delta(q, L_{\mathscr{A}}(m))$. So a winning strategy for the Duplicator on the game of $\mathscr{A}$ on $\mathscr{M}$ is also a winning strategy for the Duplicator on the game of $\mathscr{A}^*$ on $\mathscr{M}$. As the Duplicator has such a strategy ($\mathscr{M} \models \phi(s)$), we are done.

Vice versa, if $\mathscr{M}$ is accepted by $\mathscr{A}^*$ then $\mathscr{M} \models \Box^*(s \to p)$, otherwise there would be an $m \in \mathscr{M}$ with $\delta^*(q, L_{\mathscr{A}^*}(m)) = \emptyset$ for all $q \in Q$, and the Spoiler could force a game to arrive in m, where the Duplicator would not be able to proceed. To prove that $\mathscr{M} \models \phi(s)$, just note that a winning strategy of the Duplicator on the game of $\mathscr{A}^*$ on $\mathscr{M}$ is also a winning strategy for the Duplicator on the game of $\mathscr{A}$ on $\mathscr{M}$. Hence $\mathscr{M} \models \phi(s)$.

To prove the same for the uniform interpolant of ψ, notice that the operator defined in Theorem 3.3, that transforms an automata for ψ in an automata for the uniform interpolant, preserves tree-like forms and monotonicity; hence to finish the proof we only need to apply this operator on the automaton $\mathscr{A}^*$. ⊣

LEMMA 4.8. *If $\phi(p)$ is monotone in p, then there exists an automaton in tree-like form which is equivalent to $\phi(p)$ and monotone in p.*

PROOF. Let Λ be the set of atomic actions occurring in the formula $\phi(p)$. Consider the sentence $\psi = \Box^*(s \to p) \wedge \phi(s)$, where s is a new proposition constant. We have $\models \psi \to \phi(p)$: if not, there would be a structure $\mathscr{M}$ such that $\mathscr{M} \models \psi \wedge \neg\phi(p)$. We may assume that $\mathscr{M}$ is generated from its root by means of transitions labeled with actions in Λ. Then $s^{\mathscr{M}} \subseteq p^{\mathscr{M}}$, so by the monotonicity of $\phi(p)$ from $\mathscr{M} \models \phi(s)$ it follows that $\mathscr{M} \models \phi(p)$, a contradiction. Hence $\models \psi \to \phi(p)$ and any interpolant θ of this implication will be equivalent to $\phi(p)$: we have $\models \theta \to \phi(p)$ and $\models \psi \to \theta$ by the property of the interpolant; from $\models \psi \to \theta$ it follows $\models (\psi \to \theta)[s := p]$, that is, $\models \phi(p) \to \theta$. If we could find an interpolant of $\models \psi \to \phi(p)$ with an equivalent automaton which is in tree-like form and monotone in p, then we would have proved the lemma. But a uniform interpolant of ψ with respect to $\mathscr{L}(\psi) \setminus \{s\}$ is in particular an interpolant for the implication $\models \psi \to \phi(p)$, hence the lemma follows from Lemma 4.7. ⊣

THEOREM 4.9 (Lyndon Theorem). *A monotone μ-sentence is equivalent to a positive one.*

PROOF. If ϕ is a monotone sentence, by Lemma 4.8 we can find an equivalent automaton which is in tree-like form and monotone in p. By Lemma 4.6 such an automaton is equivalent to a positive sentence, which in turn is equivalent to ϕ. ⊣

The theorem can be easily generalized to monotonicity in more than one argument: if $\phi(p_1, \dots, p_n)$ is monotone in each of $p_1, \dots, p_n$ then there is an equivalent formula which has only positive occurrences of these proposition constants.

§5. The Łoś-Tarski Theorem.

In this section we find a syntactic characterization of μ-sentences that are preserved under substructures. To this end, we define *universal modal formulae*, which are constructed without using the diamond operators $\langle a \rangle$.

DEFINITION 5.1. The set of *universal formulae* of the μ-calculus is defined as follows:

$$\phi ::= p \mid \neg p \mid X \mid \phi \vee \phi \mid \phi \wedge \phi \mid [a]\phi \mid \mu X.\phi \mid \nu X.\phi$$

where $p \in \text{PROP}$, $X \in \text{VAR}$ and $a \in A$. ⊣

This is an extension to the modal μ-calculus of the notion of universal modal formula as defined in modal logic. The reader may consult [23, 24] for this definition. Here one can also find a Łoś-Tarski Theorem for modal logic. Things are easier in the modal case, because modal logic is a fragment of first-order logic and compactness and saturation are available.

DEFINITION 5.2. A μ-sentence ϕ is *preserved under substructures* if for all structures $\mathscr{M} \subseteq \mathscr{N}$: if $\mathscr{N} \models \phi$ then $\mathscr{M} \models \phi$.

We mean '*substructure*' in the traditional structure-theoretical sense. This is to be contrasted with generated substructures. Because the root is interpreted by a constant, the root of the substructure is simply the root of the original: $r^{\mathscr{M}} = r^{\mathscr{N}}$.

LEMMA 5.3. *A universal μ-sentence is preserved under substructures.*

PROOF. Given a universal μ-formula ϕ with free variables $X_1, \dots, X_n$, we prove by induction on its complexity that for all structures $\mathscr{M} \subseteq \mathscr{N}$ and for all subsets $B_1, \dots, B_n$ of $\mathscr{N}$ it holds:

$$[\![\phi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n]} \cap M \subseteq [\![\phi]\!]^{\mathscr{M}}_{[X_1:=B_1\cap M,\dots,X_n:=B_n\cap M]}.$$

The desired result follows by applying this to a sentence ϕ.

We only work out the cases $\phi(X_1, \dots, X_n) = \mu Y.\psi(X_1, \dots, X_n, Y)$ and $\phi(X_1, \dots, X_n) = \nu Y.\psi(X_1, \dots, X_n, Y)$.

- $\phi(X_1, \dots, X_n) = \mu Y.\psi(X_1, \dots, X_n, Y)$.

For a structure $\mathscr{S}$, $s \in \mathscr{S}$, and $S_1, \dots, S_n \subseteq \mathscr{S}$ we have by the property of the fixed point operator μ that $s \in [\![\phi]\!]^{\mathscr{S}}_{[X_1:=S_1,\dots,X_n:=S_n]}$ if and only if whenever $[\![\psi]\!]^{\mathscr{S}}_{[X_1:=S_1,\dots,X_n:=S_n,Y:=Z]} \subseteq Z$ for some $Z \subseteq S$ then $s \in Z$.

Suppose that $m \in [\![\phi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n]} \cap M$, and let $Z \subseteq M$ be such that $[\![\psi]\!]^{\mathscr{M}}_{[X_1:=B_1\cap M,\dots,X_n:=B_n\cap M,Y:=Z]} \subseteq Z$. Using induction we have

$$[\![\psi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n,Y:=Z\cup(N\setminus M)]} \cap M \subseteq [\![\psi]\!]^{\mathscr{M}}_{[X_1:=B_1\cap M,\dots,X_n:=B_n\cap M,Y:=Z]},$$

hence

$$[\![\psi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n,Y:=Z\cup(N\setminus M)]} \subseteq Z \cup (N \setminus M).$$

Since $m \in [\![\phi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n]}$ we obtain $m \in Z \cup (N \setminus M)$, and from $m \in M$ we have $m \in Z$.

- $\phi(X_1, \dots, X_n) = \nu Y.\psi(X_1, \dots, X_n, Y)$.

For a structure $\mathscr{S}$, $s \in \mathscr{S}$, and $S_1, \dots, S_n \subseteq \mathscr{S}$ we have by the property of the fixed-point operator ν that $s \in [\![\phi]\!]^{\mathscr{S}}_{[X_1:=S_1,\dots,X_n:=S_n]}$ if and only if there exists $Z \subseteq S$ such that $Z \subseteq [\![\psi]\!]^{\mathscr{S}}_{[X_1:=S_1,\dots,X_n:=S_n,Y:=Z]}$ and $s \in Z$.

Suppose that $m \in [\![\phi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n]} \cap M$, and let $Z \subseteq N$ be such that

$$Z \subseteq [\![\psi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n,Y:=Z]},$$

and $m \in Z$. Let $Z' = Z \cap M$. We have $Z' \subseteq [\![\psi]\!]^{\mathscr{N}}_{[X_1:=B_1,\dots,X_n:=B_n,Y:=Z]} \cap M$, and by the induction hypothesis: $Z' \subseteq [\![\psi]\!]^{\mathscr{M}}_{[X_1:=B_1\cap M,\dots,X_n:=B_n\cap M,Y:=Z']}$. Since $m \in Z'$ we are done. ⊣

In order to prove the converse of Lemma 5.3, we introduce a property of μ-automata that corresponds to universality of μ-sentences.

DEFINITION 5.4. An automaton is *closed under subsets* if for all $q \in Q$, $S \subseteq \Sigma$, $D, D' \subseteq \Lambda \times Q$ such that $D \in \delta(q, S)$ and $D' \subseteq D$ it holds $D' \in \delta(q, S)$.

LEMMA 5.5. *Let $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ be a μ-automaton in tree-like form. If $\mathscr{A}$ is closed under subsets, then the μ-sentence $\phi_{\mathscr{A}}$ corresponding to the automaton $\mathscr{A}$ is equivalent to a universal μ-sentence.*

PROOF. Suppose that the tree-like automaton $\mathscr{A}$ is closed under subsets. By induction on the tree-like structure of Q, we prove that α_q is a universal formula:

• If q is a leaf of $(Q, \leq_{\mathscr{A}})$ then for any $S \subseteq \Sigma$, $D \in \delta(q, S)$, $a \in \Lambda$, and $(a, q_1) \in D$ the state q_1 is a back-state, and the sentence $\beta_{q.q_1}$ is equal to X_{q_1}. Since $\mathscr{A}$ is closed under subsets, we can prove that the formula

$$\beta = \bigvee_{D \in \delta(q.S)} \alpha_S \wedge \bigwedge_{a \in \Lambda} \left(\bigwedge_{(a.q_1) \in D} \langle a \rangle X_{q_1} \wedge [a] \left(\bigvee_{(a.q_1) \in D} X_{q_1} \right) \right)$$

is equivalent to the universal formula:

$$\gamma = \bigvee_{D \in \delta(q.S)} \alpha_S \wedge \bigwedge_{a \in \Lambda} [a] \left(\bigvee_{(a.q_1) \in D} X_{q_1} \right).$$

It is clear that β implies γ. For the converse, suppose that a disjunct $\alpha_S \wedge \bigwedge_{a \in \Lambda} [a] (\bigvee_{(a.q_1) \in D} X_{q_1})$ of γ holds in a state s of a structure $\mathscr{M}$ under a valuation V of the free variables in γ. Consider the set D' defined by

$$D' = \{(a, q_1) \in D : \exists s'(sR_a s' \wedge \mathscr{M}, V, s' \models X_{q_1})\}.$$

Then $\mathscr{M}, V, s \models \alpha_S \wedge \bigwedge_{a \in \Lambda} \left(\bigwedge_{(a.q_1) \in D'} \langle a \rangle X_{q_1} \right) \wedge [a] (\bigvee_{(a.q_1) \in D'} X_{q_1})$, which is a disjunct of β since $\mathscr{A}$ is closed under subsets.

• Suppose we already proved that the formula α_{q_1} is universal, for all q_1 such that $q <_A q_1$. Then the subformulae $\beta_{q.q_1}$ of α_q are universal, and the same argument used in the leaf case can be applied. ⊣

We prove now a kind of converse to 5.5:

LEMMA 5.6. *For any μ-sentence which is preserved under substructures there exists a corresponding automaton which is closed under subsets.*

PROOF. Consider a μ-sentence ϕ preserved under substructures and a μ-automaton $\mathscr{A} = (Q, q_0, \Sigma, \Lambda, \delta, \Omega)$ in tree-like form which corresponds to ϕ. From $\mathscr{A}$ we construct the automaton $\hat{\mathscr{A}} = (Q, q_0, \Sigma, \Lambda, \hat{\delta}, \Omega)$ by defining $\hat{\delta}(q, S) = \{D' : \exists D \in \delta(q, S), D' \subseteq D\}$. It is easy to verify that the automaton $\hat{\mathscr{A}}$ is in tree-like form, whenever $\mathscr{A}$ it is so. Moreover, $\hat{\mathscr{A}}$ is closed under subsets. Using the fact that the sentence ϕ is preserved under substructures, we now prove that a structure $\mathscr{M}$ is accepted by $\mathscr{A}$ if and only if it is accepted by $\hat{\mathscr{A}}$, and this will conclude the proof.

Since $\delta(q, S) \subseteq \hat{\delta}(q, S)$, any winning strategy for the Duplicator on the game of $\mathscr{A}$ on $\mathscr{M}$ is also a winning strategy for the automaton $\hat{\mathscr{A}}$. Hence any structure accepted by $\mathscr{A}$ is accepted by $\hat{\mathscr{A}}$.

For the converse we suppose without loss of generality that any state $q \in Q$ is nonempty, that is, there exists a structure $\mathscr{M}_q$ which is accepted by the automaton

$(\mathscr{A}, q)$ (which is the same as $\mathscr{A}$ except that the initial state is q). We also suppose without loss of generality that $\mathscr{M}_q$ has an $\mathscr{A}$-labeling l_q, and that $\mathscr{M}$ has an $\hat{\mathscr{A}}$-labeling $\hat{l}$. From $\mathscr{M}$ we construct a structure $\mathscr{M}'$ in such a way that $\mathscr{M}$ is a substructure of $\mathscr{M}'$, and $\mathscr{M}'$ has an $\mathscr{A}$-labeling. Since the automaton $\mathscr{A}$ corresponds to the sentence ϕ which is preserved under substructures, this will imply that $\mathscr{M}$ is accepted by $\mathscr{A}$.

- Construction of $\mathscr{M}'$. Since $\hat{\mathscr{A}}$ accepts $\mathscr{M}$ via the labeling $\hat{l}$, for any node s of $\mathscr{M}$ the set $\hat{D}(s) = \{(a, q') : \exists t(sR_a^{\mathscr{M}} t \wedge \hat{l}(t) = q')\}$ belongs to $\hat{\delta}(\hat{l}(s), L(s))$ and, by definition of $\hat{\delta}$, there exists a set $D_s \in \delta(\hat{l}(s), L(s))$ such that $\hat{D}(s) \subseteq D_s$. Define $\mathscr{M}'$ as the disjoint union of $\mathscr{M}$ and all the $\mathscr{M}_q$, for $q \in Q$. The root of $\mathscr{M}'$ is the root of M, and there are some new a-transitions from s to the root $r^{\mathscr{M}_q}$ of $\mathscr{M}_q$, if $(a, q) \in D_s \setminus \hat{D}(s)$.
- Construction of the $\mathscr{A}$-labeling $l : \mathscr{M}' \to Q$: define $l(s') = \hat{l}(s')$ if $s' \in \mathscr{M}$, and $l(s') = l_q(s')$ if $s' \in \mathscr{M}_q$, for $q \in Q$.

We prove that l is an $\mathscr{A}$-labeling on $\mathscr{M}'$, by showing that for all $s \in \mathscr{M}'$

$$D(s) = \{(a, q') : \exists t\ (sR_a^{\mathscr{M}'} t \wedge l(t) = q')\} \in \delta(l(s), L(s)),$$

since the other conditions of a labeling are trivially satisfied.

Suppose first that $s \in \mathscr{M}$. In this case $l(s) = \hat{l}(s)$ and we claim that the set $D(s)$ is equal to D_s and hence it belongs to $\delta(l(s), L(s))$. To prove that $D(s) \subseteq D_s$, notice that $\{t : sR_a^{\mathscr{M}'} t\} = \{t : sR_a^{\mathscr{M}} t\} \cup \{r^{\mathscr{M}_q} : (a, q) \in D_s \setminus \hat{D}(s)\}$. If $(a, q') \in D(s)$, then there exists a $t \in \mathscr{M}'$ such that $sR_a^{\mathscr{M}'} t \wedge l(t) = q'$. If $sR_a^{\mathscr{M}} t$, then $q' = l(t) = \hat{l}(t)$ and $(a, q') \in \hat{D}(s) \subseteq D_s$. If $t \in \{r^{\mathscr{M}_q} : (a, q) \in D_s \setminus \hat{D}(s)\}$, then $q' = l(t) = l_q(r^{\mathscr{M}_q}) = q$, hence $(a, q') \in D_s$. Vice versa, if $(a, q) \in D_s$, then either $(a, q) \in \hat{D}(s)$ or $(a, q) \in D_s \setminus \hat{D}(s)$. In the first case, since $\hat{D}(s) \subseteq D(s)$ as it is easily seen from the definitions of the two sets, we have $(a, q) \in D(s)$. In the second case, $sR_a^{\mathscr{M}'} r^{\mathscr{M}_q}$ and $l(r^{\mathscr{M}_q}) = q$, proving that $(a, q) \in D(s)$.

Now suppose that $s \in \mathscr{M}' \setminus \mathscr{M}$. Then there is a $q \in Q$ such that $s \in \mathscr{M}_q$, $\{t : sR_a^{\mathscr{M}'} t\} = \{t : sR_a^{\mathscr{M}_q} t\}$ and $l(t) = l_q(t)$ for any $t \in \{t : sR_a^{\mathscr{M}'} t\}$. Thus $D(s)$ is equal to the set $\{(a, q') : \exists t(sR_a^{\mathscr{M}_q} t \wedge l_q(t) = q')\}$, which is in $\delta(l_q(s), L(s))$ because l_q is an $\mathscr{A}$-labeling of $\mathscr{M}_q$. Then $D(s) \in \delta(l(s), L(s))$, because $l(s) = l_q(s)$. ⊣

THEOREM 5.7 (Łoś-Tarski Theorem). *A μ-sentence which is preserved under substructures is equivalent to a universal sentence.*

PROOF. Consider a μ-sentence ϕ which is preserved under substructures and an automaton $\mathscr{A}$ corresponding to ϕ. By Lemma 5.5 we may suppose that $\mathscr{A}$ is closed under subsets, and by Lemma 5.6 $\mathscr{A}$ is equivalent to a universal sentence. ⊣

REMARK. Notice that the operator on automata taking $\mathscr{A} = (Q, \Sigma, \Lambda, q_0, \delta, \Omega)$ to $\mathscr{B} = (Q, \Sigma \setminus \{p\}, \Lambda, q_0, \delta', \Omega)$ with:

$$\delta'(q, L) := \delta(q, L) \cup \delta(q, L \cup \{p\}),$$

defined in 3.3, preserves tree-like-forms, closure under subsets, and monotonicity. Hence Lemmas 5.6, 5.5, 4.6 and 4.8 allow us to conclude that the uniform interpolant of a universal sentence may be chosen to be universal and that the uniform interpolant of a sentence positive in p may be chosen positive in p.

§6. Conclusion. We have proved three important classical theorems for the μ-calculus. Our main tool for the proofs of these theorems have been μ-automata. Our paper shows that these μ-automata are an extremely useful tool for studying the μ-calculus. For the most part, this is due to the fact that they are more easily manipulated than the μ-formulae themselves.

The questions we have asked here can also be asked for other modal logics. As regards interpolation, we know that the μ-calculus is quite unique in the landscape of temporal logics: Linear Temporal Logic ([21]), Computation Tree Logic (CTL, [2]) and CTL* ([5, 6]) do not have Craig interpolation ([17, 18]). PDL does not have uniform interpolation, while the question whether PDL has Craig interpolation is still unclear. Lyndon and Łoś-Tarski Theorems for these logics have yet to be supplied.

Finally, there are still issues concerning the bisimulation-quantifiers. For example, we have an axiomatization for the μ-calculus ([29]). Can we also give an axiomatization for BQL? The question is even open for the logic we get by adding bisimulation-quantifiers to modal logic (which is still equivalent in expressivity to modal logic).

REFERENCES

[1] C.C. Chang and H.J. Keisler, ***Model theory***, Studies in Logic and the Foundations of Mathematics, vol. 173, Elsevier Science Publishers B.V., 1977.

[2] E.M. Clarke, E.A. Emerson, and A.P. Sistla, *Automatic verification of finite-state concurrent systems using temporal logic specifications*, ***ACM Transactions on Programming Languages and Systems***, vol. 8(2) (1986), pp. 244–263.

[3] W. Craig, *Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory*, this Journal, vol. 22 (1957), pp. 269–285.

[4] H. Ebbinghaus and J. Flum, ***Finite model theory***, Springer, Berlin, 1995.

[5] E.A. Emerson and J.Y. Halpern, *Sometimes and not never revisited: on branching versus linear time*, ***Journal of the ACM***, vol. 33(1) (1986), pp. 151–178.

[6] E.A. Emerson and Chin-Laung Lei, *Modalities for model checking: branching time logic strikes back*, ***Science of Computer Programming***, vol. 8 (1987), pp. 275–306.

[7] M.J. Fisher and R.E. Ladner, *Propositional dynamic logic of regular programs*, ***Journal of Computer and System Sciences***, vol. 18(2) (1979), pp. 194–211.

[8] Y. Gurevich, *Towards logic tailored for logical complexity*, ***Computation and proof theory*** (M. M. Richter et al., editors), Lecture Notes in Mathematics, vol. 1104, Springer, 1984, pp. 175–216.

[9] L. Henkin, *An extension of the Craig-Lyndon interpolation theorem*, this Journal, vol. 28 (1963), pp. 201–216.

[10] W. Hodges, ***Model theory***, Encyclopedia of Mathematics and Its Applications, vol. 42, Cambridge University Press, 1993.

[11] D. Janin, *Propriétés logiques du nondéterminisme et μ-calcul modal*, ***Ph.D. thesis***, LaBRI–Université de Bordeaux I, 1996.

[12] D. Janin and I. Walukiewicz, *Automata for the modal μ-calculus and related results*, ***Proceedings MFCS'95***, Springer, 1995, pp. 552–562.

[13] ———, *On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic*, to appear in the Proceedings of CONCUR'96, 1996.

[14] D. Kozen, *Results on the propositional μ-calculus*, ***Theoretical Computer Science***, vol. 27 (1983), pp. 234–241.

[15] J. Łoś, *On the extending of models (I)*, ***Fundamentae Mathematicae***, vol. 42 (1955), pp. 38–54.

[16] R.C. Lyndon, *An interpolation theorem in the predicate calculus*, ***Pacific Journal of Mathematics***, vol. 9 (1959), pp. 155–164.

[17] L. MAKSIMOVA, *Absence of interpolation and of Beth's property in temporal logics with "the next" operation*, ***Siberian Mathematical Journal***, vol. 32(6) (1991), pp. 109–113.

[18] ———, *Temporal logics of "the next" do not have the Beth property*, ***Journal of applied non-classical logics***, vol. 1(1) (1991), pp. 73–76.

[19] D. NIWIŃSKI, *Fixed points vs. infinite generation*, ***Proceedings of the 3rd Annual IEEE Symposium on Logic in Computer Science***, 1988, pp. 402–409.

[20] A. PITTS, *On an interpretation of second-order quantification in first-order intuitionistic propositional logic*, this JOURNAL, vol. 57 (1992), pp. 33–52.

[21] A. PNUELI, *The temporal logic of programs*, ***Proceedings of the 18th Annual Symposium on Foundations of Computer Science***, 1977, pp. 46–57.

[22] V. PRATT, *Semantical considerations on Floyd-Hoare logic*, ***Proceedings of the 17th IEEE Symposium on Foundations of Computer Science***, 1976, pp. 109–121.

[23] M. DE RIJKE, *Extending modal logic*, ***Ph.D. thesis***, ILLC-dissertation series 1993-4, 1993.

[24] ———, *Modal model theory*, ***Annals of Pure and Applied Logic*** (1997), to appear.

[25] R.S. STREETT and E.A. EMERSON, *An automata theoretic procedure for the propositional mu-calculus*, ***Information and Computation***, vol. 81 (1989), pp. 249–264.

[26] A. TARSKI, *Contributions to the theory of models I, II*, ***Indagationes Mathematicae***, vol. 16 (1954), pp. 572–588.

[27] W. THOMAS, *Automata on infinite objects*, ***Handbook of theoretical computer science, volume B*** (J. van Leeuwen, editor), Elsevier, 1990, pp. 133–191.

[28] A. VISSER, *Bisimulations, model descriptions and propositional quantifiers*, Logic Group Preprint Series, vol. 161, Department of Philosophy, Utrecht University, 1996.

[29] I. WALUKIEWICZ, *Completeness of Kozen's axiomatization of the propositional μ-calculus*, ***Proceedings LICS'95***, 1995, pp. 14–24.

ILLC, UNIVERSITEIT VAN AMSTERDAM
PLANTAGE MUIDERGRACHT 24
1018 TV AMSTERDAM, THE NETHERLANDS

and

UDINE UNIVERSITY
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
VIALE DELLE SCIENZE 206
33100 UDINE, ITALY
E-mail: dagostin@dimi.uniud.it

UTRECHT UNIVERSITY
DEPARTMENT OF PHILOSOPHY
HEIDELBERGLAAN 8, 3584 CS UTRECHT
THE NETHERLANDS
E-mail: Marco.Hollenberg@mail.com