The Łoś-Tarski and Lyndon Theorems for the μ-logic

Giovanna D'Agostino

July 2, 1997

1 Abstract

We prove the Lyndon Theorem and the Łoś-Tarski Theorem for the modal μ-calculus, using automata which run on process graphs and are equivalent in expressive power to the μ-calculus.

2 Introduction

In first-order logic the Łoś-Tarski Theorem and Lyndon Theorem characterize syntactically fragments of first-order logic having certain semantic properties. More precisely, the Łoś-Tarski Theorem says that the first-order sentences that are preserved under submodels are the universal ones, while the Lyndon Theorem characterizes, as the ones that are positive in P, the sentences that are preserved when the interpretation of the unary predicate P is enlarged.

Similar characterizations can be given in modal logic. Here the role of the universal formulae is played by those sentences which are constructed without the use of the $\diamond$ operator.

In this paper, a Lyndon and Łoś-Tarski Theorem is given for an extension of modal logic: the Kozen modal μ-calculus ([4]), which consists of basic modal logic plus least and greatest fixed points operators. The μ-calculus is a powerful fragment of monadic second order logic, well suited to describe properties of processes, when these are represented as classes of labeled transition systems. Indeed, two bisimilar systems represent the same process, and it has been recently proved that the μ-calculus consists exactly of those monadic second-order sentences which are invariant under bisimulation (see [3]; we refer to this theorem as *the MSO-Invariance Theorem*). Hence, at least where processes are concerned, the μ-calculus has the same strength as that of monadic second-order logic. In spite of its expressivity, the modal μ-calculus enjoys some desirable properties such as decidability, a complete calculus and the finite model property.

To give some examples of the possible use of these theorems, consider first the class of (inversely) well-founded [1] process graphs. This class is described by the second-order formula $\forall P(r \in P \rightarrow \exists y(y \in P \wedge \forall w(w \in P \rightarrow \neg yRw)))$, where r is a constant denoting the root and R is the accessibility relation of the process graph. Since the well-founded class is closed under bisimulation, the MSO-Invariance Theorem of [3] tells us that there is a μ-sentence describing the class. But the well-founded class is also closed under submodels, hence the Łoś-Tarski Theorem for the μ-logic would assure us that no diamond is needed to describe the class. Indeed, the well-founded processes are described by $\mu X.\Box X$. Note that this example shows why the straightforward generalization to monadic second-order logic of the first-order Łoś-Tarski Theorem does not hold. Universal quantifiers are not enough in MSO-logic to describe the well-founded class: the existential first-order quantifier in the second-order description of the class is essential because a Π^1_1 formula with a universal first-order matrix would be equivalent to a first-order formula, but we know that the class of well-founded process graphs is not elementary.

Consider now the Lyndon Theorem. In the μ-calculus the property of positivity in a variable X is used to build formulae: we are only allowed to use the fixed-point quantifier $\mu X.\phi(X)$ if the formula $\phi(X)$ is positive in X, because this assures that the operator $\phi(X)$ is monotone and hence that its least fixed point exists. But once we have all μ-formulae, what about the fixed point of a monotone formula $\phi(X)$ which is not positive in X? The Lyndon Theorem for the μ-logic would say that we can recover all these fixed points in the μ-calculus, because any monotone formula $\phi(X)$ is equivalent to a formula which is positive in X.

Both the Lyndon Theorem and the Łoś-Tarski Theorem are proved using μ-automata, defined in [2, 3]. These automata run on models of the μ-calculus and correspond exactly to μ-sentences: to any μ-sentence there exists an automaton that recognizes the same models and vice versa.

The paper is organized as follows. First, a preliminary section in which the μ-calculus and μ-automata are introduced. The second section deals with the Lyndon Theorem, which is proved using interpolation for the μ-calculus ([1]). Finally, in the third section we prove the Łoś-Tarski Theorem for the μ-calculus.

3 Preliminaries

In what follows we consider μ-formulae as being constructed from a set of propositional constants $\{P_1, \ldots\}$, their negations, and propositional variables $\{X_1, \ldots\}$ using conjunctions, disjunctions, the polymodal operators $\Box_a$ and $\Diamond_a$ for a in a set of atomic actions $\{a_1, \ldots\}$, and the fixed point operators μ, ν. The μ-sentences are the μ-formulae without free propositional variables (the variable X is not free in $\mu X.\phi(X)$ or $\nu X.\phi(X)$). We refer to the set of atomic actions and propositional constants that occur in a formula ϕ as the language of ϕ and denote it by $\mathcal{L}(\phi)$.

[1] That is: every subset of the domain containing the root has an R-maximal element

Given a finite subset Σ_p of propositional variables and a finite subset Σ_r of atomic actions, a **process graph** for the language Σ_p, Σ_r is a first-order model $\mathcal{M}$ of the first-order language with a unary predicate P for any $P \in \Sigma_p$, a binary predicate R_a for any $a \in \Sigma_r$, and a constant symbol r denoting the root. We denote by M the domain of $\mathcal{M}$, by $P^{\mathcal{M}}$, $R_a^{\mathcal{M}}$, $r^{\mathcal{M}}$, the interpretation of the predicates and of the constant in $\mathcal{M}$, and if $s \in M$ we let $R_a(s) = \{t \in M : sR_a^{\mathcal{M}}t\}$. In the following, we often refer to a process graph simply as a **model**. If $P \in \Sigma_p$, $\mathcal{M}$ is a process graph, and A is a subset of the domain of $\mathcal{M}$, we denote by $\mathcal{M}[P := A]$ the model which is like $\mathcal{M}$ except that P is now interpreted by A. If $s \in M$ we denote by $(\mathcal{M}, s)$ the process graph which is like $\mathcal{M}$ but with root equal to s.

Given a model $\mathcal{M}$ and a valuation $V : \{X_1, \ldots\} \to \wp(M)$, a μ-formula ϕ is interpreted as the subset $[\![\phi]\!]_V^{\mathcal{M}}$ of the domain M defined as follows:

$$
\begin{array}{lcl}
[\![P]\!]_V^{\mathcal{M}} & := & P^{\mathcal{M}} \\
[\![X]\!]_V^{\mathcal{M}} & := & V(X) \\
[\![\neg\phi]\!]_V^{\mathcal{M}} & := & M - [\![\phi]\!]_V^{\mathcal{M}} \\
[\![\phi \vee \psi]\!]_V^{\mathcal{M}} & := & [\![\phi]\!]_V^{\mathcal{M}} \cup [\![\psi]\!]_V^{\mathcal{M}} \\
[\![\phi \wedge \psi]\!]_V^{\mathcal{M}} & := & [\![\phi]\!]_V^{\mathcal{M}} \cap [\![\psi]\!]_V^{\mathcal{M}} \\
[\![\Diamond_a\phi]\!]_V^{\mathcal{M}} & := & \{s \in \mathcal{M} \mid [\![\phi]\!]_V^{\mathcal{M}} \cap \{t : sR_at\} \neq \emptyset\} \\
[\![\Box_a\phi]\!]_V^{\mathcal{M}} & := & \{s \in \mathcal{M} \mid \{t : sR_at\} \subseteq [\![\phi]\!]_V^{\mathcal{M}}\} \\
[\![\mu X.\phi]\!]_V^{\mathcal{M}} & := & \bigcap\{S \subseteq M \mid [\![\phi]\!]_{V[X:=S]}^{\mathcal{M}} \subseteq S\} \\
[\![\nu X.\phi]\!]_V^{\mathcal{M}} & := & \bigcup\{S \subseteq M \mid S \subseteq [\![\phi]\!]_{V[X:=S]}^{\mathcal{M}}\}
\end{array}
$$

where $V[X := S]$ is similar to the valuation V except that X is assigned to S. Note that $[\![\mu X.\phi]\!]_V^{\mathcal{M}}$ is the least fixed point of the monotone operator $S \mapsto [\![\phi]\!]_{V[X:=S]}^{\mathcal{M}}$, and $[\![\nu X.\phi]\!]_V^{\mathcal{M}}$ its greatest fixed point. Note also that if $X_1, \ldots, X_n$ are the free variables that occur in ϕ, the valuation becomes irrelevant on other variables: given sets $A_1, \ldots, A_n$ we may unambiguously write $[\![\phi]\!]_{[X_1:=A_1,\ldots,X_n:=A_n]}^{\mathcal{M}}$. We will often denote $s \in [\![\phi]\!]_V^{\mathcal{M}}$ by $(\mathcal{M}, V, s) \models \phi$; in case of a sentence ϕ and when the model is well understood, we will also write $s \models \phi$. Finally, $\mathcal{M} \models \phi$ stands for $r^{\mathcal{M}} \models \phi$.

In the following we will only consider **connected** process graphs, in which it is possible to reach any element of the domain from the root via a finite number of steps using the relations $\{R_a^{\mathcal{M}} : a \in \Sigma_r\}$. Since μ-sentences are invariant under bisimulation and any process graph is bisimilar to a connected one, we can make this restriction without loss of generality.

To prove the Lyndon Theorem and the Łoś-Tarski Theorem we use μ-automata, defined in [2], in the version of [1]:

Definition 3.1 *A* **μ-automaton** *([2]) $\mathcal{A}$ is a tuple $(Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ such that:*

1. *Q is a finite set of states;*
2. *$q_0 \in Q$ is the initial state;*
3. *Σ_p is a finite subset of the set of propositional constants;*

4. Σ_r *is a finite subset of the set of atomic actions;*

5. $\delta : Q \times \wp(\Sigma_p) \to \wp\wp(\Sigma_r \times Q)$*;*

6. $\Omega : Q \to \mathbb{N}$. □

We refer to (Σ_p, Σ_r) as the language of the automaton $\mathcal{A}$. Given a process graph $\mathcal{M}$ in the language of $\mathcal{A}$, a **μ-game** of $\mathcal{A}$ on $\mathcal{M}$ is played between two players, the Duplicator and the Spoiler, as follows:

1. The starting position is $(r^{\mathcal{M}}, q_0)$;
2. If we are in a position (s, q) then the Duplicator has to make a move. Define $L(s)$ as $\{P \in \Sigma_p \mid s \models P\}$. A legal move for the Duplicator consists of a **marking** $m : \Sigma_r \times Q \to \wp(\{t : \exists a(sR_a t)\})$ such that $m(a, q') \subseteq \{t : sR_a t\}$ for every $(a, q') \in \Sigma_r \times Q$, and such that there exists a $D \in \delta(q, L(s))$ with:
   (a) For all $a \in \Sigma_r$ and all t such that $sR_a t$ there is an $(a, q') \in D$ with $t \in m(a, q')$.
   (b) $(a, q') \in D$ if and only if $m(a, q') \neq \emptyset$.
3. If the Duplicator has just made a move, namely a marking m, the Spoiler picks a $t \in m(a, q)$ (for some $a \in \Sigma_r, q \in Q$). The new position becomes (t, q).

Either player wins the game if the other player cannot make a move. An infinite game $(r^{\mathcal{M}} = s_0, q_0), m_0, (s_1, q_1), m_1, (s_2, q_2), m_2 \ldots$ is won by the Duplicator if:

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

is even. We say that $\mathcal{M}$ is **accepted** by $\mathcal{A}$ iff the Duplicator has a winning strategy on $\mathcal{M}$.

Janin and Walukiewicz ([2]) prove that μ-automata have the same expressive power than μ-sentences: for every μ-sentence ϕ there is an automaton $\mathcal{A}_\phi$ in the language $\mathcal{L}(\phi)$ such that $\mathcal{A}_\phi$ accepts a process graph $\mathcal{M}$ iff $\mathcal{M} \models \phi$. The converse also holds: for every μ-automaton $\mathcal{A}$ there is a μ-sentence $\phi_{\mathcal{A}}$ in the same language such that $\mathcal{A}$ accepts a model $\mathcal{M}$ iff $\mathcal{M} \models \phi_{\mathcal{A}}$.

A formula ϕ and an automaton $\mathcal{A}$ are said to be equivalent if for all $\mathcal{M}$, the automaton $\mathcal{A}$ accepts $\mathcal{M}$ iff $\mathcal{M} \models \phi$.

To help us recognize which process graphs are accepted by a given automaton $\mathcal{A}$, we will use the notion of $\mathcal{A}$-labeling, introduced in [1].

Given a process graph $\mathcal{M}$ and a μ-automaton $\mathcal{A}$ an **$\mathcal{A}$-labeling** of $\mathcal{M}$ is a total function $l : \mathcal{M} \to Q$ such that:

1. $l(r^{\mathcal{M}}) = q_0$;

2. If $l(s) = q$ then:

$$D(s) := \{(a, q') \in \Sigma_r \times Q \mid \exists t \; (sR_a t \wedge l(t) = q'\} \in \delta(q, L(s))$$

3. For any infinite path $r^{\mathcal{M}} \overset{a_1}{\to} s_1 \overset{a_2}{\to} \dots$ in $\mathcal{M}$:

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

is even.

In [1] it is proved:

Lemma 3.2 *A process graph $\mathcal{M}$ is bisimilar to a system which has an $\mathcal{A}$-labeling if and only if $\mathcal{M}$ is accepted by $\mathcal{A}$.*

Following Janin Thesis, we now briefly describe how to find a μ-sentence equivalent to a given automaton $\mathcal{A}$.

Definition 3.3 *An automaton $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ is said to be in* **normal** *form if there exists a partial ordering $\leq_{\mathcal{A}}$ of the set of states Q such that:*

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

When an automaton is in normal form, the μ-sentence which is equivalent to it can be constructed inductively on the partial ordering $\leq_{\mathcal{A}}$ of the set Q:

Definition 3.4 *If $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ is in normal form and $q \in Q$, the μ-formula α_q is defined as follows:*

$$\alpha_q = (\sigma_q X_q) \bigvee_{\Sigma \subseteq \Sigma_p} \bigvee_{D \in \delta(q, \Sigma)} \alpha_\Sigma \wedge \bigwedge_{a \in \Sigma_r} \bigwedge_{(a, q_1) \in D} \Diamond_a \beta_{q, q_1} \wedge \Box_a (\bigvee_{(a, q_1) \in D} \beta_{q, q_1})$$

where

1. *if $\Sigma \subseteq \Sigma_p$ then α_Σ is the formula $\bigwedge_{P \in \Sigma} P \wedge \bigwedge_{P \in \Sigma_p \setminus \Sigma} \neg P$*
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 source (in this case $q_1 \leq_A q$), and $\beta_{q, q_1} = \alpha_{q_1}$ otherwise (in this case $q \leq_A q_1$ and α_{q_1} is already defined by induction).*

For an automaton $\mathcal{A}$ in normal form with initial state q_0 Janin proved that the sentence $\phi_{\mathcal{A}} = \alpha_{q_0}$ is equivalent to $\mathcal{A}$ ([2]). Moreover, for any automaton $\mathcal{A}$, there exists an automaton $\mathcal{A}_T$ in normal form which recognizes the same class of process graphs, so that the sentence $\phi_{\mathcal{A}_T}$ is equivalent to $\mathcal{A}$.

A state q of an automaton $\mathcal{A}$ is called a **true state** if $\Omega(q) = 0$ and $\delta(q, \Sigma) = \{\{(a,q) : a \in X\} : X \subseteq \Sigma_r\}$, for all $\Sigma \subseteq \Sigma_p$. Notice that if q is a true state of the automaton $\mathcal{A}$, then the automaton $(\mathcal{A}, q)$ accepts any process graph: in a position (s, q) the Duplicator can choose the marking

$$m(a, q') \quad = \quad \begin{cases} R_a(s) & \text{if } q' = q \\ \emptyset & \text{otherwise} \end{cases}$$

This marking is correct with regards to $\{(a,q) : R_a(s) \neq \emptyset\}$, which belongs to $\delta(q, \Sigma)$, and since $\Omega(q) = 0$ this gives a winning strategy for the Duplicator.

Hence if $\mathcal{A}$ is in normal form and q_0 is a true state, the sentence α_{q_0} defined in 3.4 must be equivalent to $\top$. This is also true for a true state q which is not the initial one:

Lemma 3.5 *If $\mathcal{A}$ is in normal form and q is a true state then α_q is a sentence and it is equivalent to $\top$.*

Proof.
Since q is a true state we have $\delta(q, \Sigma) = \{\{(a,q) : a \in X\} : X \subseteq \Sigma_r\}$, and q is a back state of the automaton $\mathcal{A}$, with source q. By definition we have

$$\alpha_q = \nu X_q \bigvee_{\Sigma \subseteq \Sigma_p} \bigvee_{X \subseteq \Sigma_r} \alpha_\Sigma \wedge \bigwedge_{a \in X} \Diamond_a X_q \wedge \Box_a(X_q) \wedge \bigwedge_{a \notin X} \Box_a(\bot) = \nu X_q \phi(X_q).$$

Since α_q is the greatest fixed point of $\phi(X_q)$, to prove that α_q is equivalent to $\top$ it is enough to show that $\phi(\top) = \bigvee_{\Sigma \subseteq \Sigma_p} \bigvee_{X \subseteq \Sigma_r} \alpha_\Sigma \wedge \bigwedge_{a \in X} \Diamond_a \top \wedge \bigwedge_{a \notin X} \Box_a(\bot)$ is equivalent to $\top$, that is, it is true in any model. Consider any model $(\mathcal{M}, s)$, and let $X = \{a \in \Sigma_r : R_a(s) \neq \emptyset\}$. Then if $\Sigma = L(s)$, we have $(\mathcal{M}, s) \models \alpha_\Sigma \wedge \bigwedge_{a \in X} \Diamond_a \top \wedge \bigwedge_{a \notin X} \Box_a(\bot)$, which is a disjoint of $\phi(\top)$. $\Box$

4 The Lyndon Theorem

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

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

Positive sentences are monotone. The Lyndon Theorem establishes the converse: any monotone μ-sentence is equivalent to a positive one.

This section is devoted to the proof of this theorem. In the following definition we will isolate a property of an automaton $\mathcal{A}$ in normal form which will be proved in lemma 4.4 and lemma 4.7 to correspond to the monotonicity of the sentence $\phi_{\mathcal{A}}$.

Definition 4.3 *An automaton* $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ *is called* **monotone** *in the propositional constant* P *if for all* $q \in Q$*, and* $\Sigma \subseteq \Sigma_p$*, it is true that* $\delta(q, \Sigma) \subseteq \delta(q, \Sigma \cup \{P\})$.

Lemma 4.4 *If the automaton* $\mathcal{A}$ *is in normal form and monotone in* P*, then the* μ*-sentence* $\phi_{\mathcal{A}}$ *is positive in* P.

Proof.
Let $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ be in normal form and monotone in P, and let $\alpha_\Sigma, \beta_{q,q_1}, \alpha_q$ be the μ-formulas defined as in 3.4. For $\Sigma \subseteq \Sigma_r$, $q \in Q$, and $D \in \delta(q, \Sigma)$, define $\beta_{q,D} = \bigwedge_{a\in\Sigma_r} \bigwedge_{(a,q_1)\in D} \Diamond_a \beta_{q,q_1} \wedge \Box_a (\bigvee_{(a,q_1)\in D} \beta_{q,q_1})$. Then $\beta_{q,D}$ is positive in P whenever all subformulae β_{q,q_1} are so and

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

By the monotonicity of $\mathcal{A}$ we have that

$$\delta(q, \Sigma \cup \{P\}) = \delta(q, \Sigma) \cup [\delta(q, \Sigma \cup \{P\}) \setminus \delta(q, \Sigma)],$$

and using this equality we can prove that α_q is equivalent ($\equiv$) to a formula positive in P:

$$\bigvee_{\Sigma\subseteq\Sigma_p} \bigvee_{D\in\delta(q,\Sigma)} (\alpha_\Sigma \wedge \beta_{q,D}) \equiv$$

$$\bigvee_{\Sigma\subseteq\Sigma_p\setminus\{P\}} \bigvee_{D\in\delta(q,\Sigma)} (\alpha_\Sigma \wedge \beta_{q,D}) \vee \bigvee_{D\in\delta(q,\Sigma\cup\{P\})} (\alpha_{\Sigma\cup\{P\}} \wedge \beta_{q,D}) \equiv$$

$$\bigvee_{\Sigma\subseteq\Sigma_p\setminus\{P\}} \bigvee_{D\in\delta(q,\Sigma)} (\alpha_\Sigma \wedge \beta_{q,D}) \vee \bigvee_{D\in\delta(q,\Sigma)} (\alpha_{\Sigma\cup\{P\}} \wedge \beta_{q,D}) \vee$$

$$\bigvee_{D\in\delta(q,\Sigma\cup\{P\})\setminus\delta(q,\Sigma)} (\alpha_{\Sigma\cup\{P\}} \wedge \beta_{q,D}) \equiv$$

$$\bigvee_{\Sigma\subseteq\Sigma_p\setminus\{P\}} \bigvee_{D\in\delta(q,\Sigma)} (\alpha_\Sigma \wedge \beta_{q,D}) \vee (\alpha_{\Sigma\cup\{P\}} \wedge \beta_{q,D}) \vee$$

$$\bigvee_{D\in\delta(q,\Sigma\cup\{P\})\setminus\delta(q,\Sigma)} (\alpha_{\Sigma\cup\{P\}} \wedge \beta_{q,D}).$$

For $\Sigma \subseteq \Sigma_p \setminus \{P\}$, define α_Σ^{-P} as

$$\alpha_\Sigma^{-P} = \bigwedge_{R \in \Sigma} R \wedge \bigwedge_{R \in \Sigma_p \setminus \Sigma, R \neq P} \neg R.$$

Then α_Σ^{-P} is positive in P, $\alpha_\Sigma = \alpha_\Sigma^{-P} \wedge \neg P$, $\alpha_{\Sigma \cup \{P\}} = \alpha_\Sigma^{-P} \wedge P$, and we can proceed in the chain of equivalences as follows:

$$\equiv \bigvee_{\Sigma \subseteq \Sigma_p \setminus \{P\}} \bigvee_{D \in \delta(q,\Sigma)} (\alpha_\Sigma^{-P} \wedge \neg P \wedge \beta_{q,D}) \vee (\alpha_\Sigma^{-P} \wedge P \wedge \beta_{q,D}) \vee$$

$$\bigvee_{D \in \delta(q, \Sigma \cup \{P\}) \setminus \delta(q,\Sigma)} (\alpha_{\Sigma \cup \{P\}} \wedge \beta_{q,D}) \equiv$$

$$\bigvee_{\Sigma \subseteq \Sigma_p \setminus \{P\}} \bigvee_{D \in \delta(q,\Sigma)} (\alpha_\Sigma^{-P} \wedge \beta_{q,D}) \vee \bigvee_{D \in \delta(q, \Sigma \cup \{P\}) \setminus \delta(q,\Sigma)} (\alpha_{\Sigma \cup \{P\}} \wedge \beta_{q,D}),$$

which is positive in P whenever all formulae $\beta_{q,D}$ are so. Hence we can prove that all α_q are positive in P by induction on the partial ordering $\leq_\mathcal{A}$ of Q, starting from the leaves: given $q \in Q$ 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 $q <_A q_1$, and by induction these are positive.

$\Box$

We will prove the converse of lemma 4.4 and then the Lyndon Theorem by using the (uniform) interpolation Theorem for the μ-logic (see [1]).

The following lemma is a special case of the Lyndon Theorem and will be used to prove the full theorem (Theorem 4.8). To state the lemma, we need the concept of uniform interpolant: given a sentence ψ, its uniform interpolant with respect to a subset $\mathcal{L}$ of its language is a sentence θ in the language $\mathcal{L}$ such that

1. $\models \psi \to \theta$;
2. For any sentence ϕ, if $\mathcal{L}(\phi) \cap \mathcal{L}(\psi) \subseteq \mathcal{L}$ and $\models \psi \to \phi$ then $\models \theta \to \phi$.

In [1] it is proved that any μ-sentence has uniform interpolants with respect to any subset of its language. In particular, if R is a propositional constant we have:

Theorem 4.5 *([1]) If the automaton $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ corresponds to the μ-sentence ψ, then the automaton $\hat{\mathcal{A}} = (Q, q_0, \Sigma_p \setminus \{R\}, \Sigma_r, \hat{\delta}, \Omega)$, where $\hat{\delta}(q, \Sigma) = \delta(q, \Sigma) \cup \delta(q, \Sigma \cup \{R\})$, corresponds to a uniform interpolant of ψ with respect to $\mathcal{L}(\psi) \setminus \{R\}$.*

In the following, we denote by $\Box^*(\neg R \vee P)$ the μ-sentence $\nu X.((\neg R \vee P) \wedge \bigvee_{a \in \Sigma_r} \Box_a X)$ which is true in $\mathcal{M}$ iff for all $s \in M$, it holds $s \models \neg R \vee P$ (remember our hypothesis on the connectedness of $\mathcal{M}$). We also denote by $\phi(P/R)$ the formula which is obtained from ϕ by substituting P with R.

Lemma 4.6 *For any μ-sentence ϕ, if R is a constant not appearing in ϕ, both the sentence $\psi = \Box^*(\neg R \vee P) \wedge \phi(P/R)$ and its uniform interpolant with respect to $\mathcal{L}(\psi) \setminus \{R\}$ have an equivalent automaton which is in normal form and monotone in P.*

Proof.
We first prove that there exists an automaton monotone in P corresponding to the sentence $\psi = \Box^*(\neg R \vee P) \wedge \phi(P/R)$. If Σ_p is the set of propositional constants appearing in ϕ, then since P does not appear in $\phi(P/R)$ there exists an automaton $\mathcal{A} = (Q, q_0, (\Sigma_p \setminus \{P\}) \cup \{R\}, \Sigma_r, \delta, \Omega)$ in normal form which is equivalent to $\phi(P/R)$. Consider the following automaton $\mathcal{A}^* = (Q, q_0, \Sigma_p \cup \{R\}, \Sigma_r, \delta^*, \Omega)$, where δ^* is defined as follows: if $R \notin \Sigma$ or $P \in \Sigma$ then $\delta^*(q, \Sigma) = \delta(q, \Sigma \setminus \{P\})$, and $\delta^*(q, \Sigma) = \emptyset$, otherwise. It is clear that $\mathcal{A}^*$ is monotone in P and in normal form. We now prove that it is equivalent to ψ, that is, for any process graph $\mathcal{M}$, $\mathcal{M}$ is accepted by $\mathcal{A}^*$ if and only if $\mathcal{M} \models \psi$. Given a process graph $\mathcal{M}$ for the language of ψ, we denote by $\mathcal{M}^{-P}$ the process graph in the language without P which is obtained from $\mathcal{M}$ by ignoring the interpretation of P.

If $\mathcal{M} \models \psi$ then for any $s \in M$ we have either $R \notin L(s)$ or $P \in L(s)$, and for any $q \in Q$ it holds $\delta^*(q, L(s)) = \delta(q, L(s) \setminus \{P\})$. Moreover, $\mathcal{M}^{-P} \models \phi(P/R)$, and a winning strategy for the Duplicator on the game of $\mathcal{A}$ on $\mathcal{M}^{-P}$ is also a winning strategy for the Duplicator on the game of $\mathcal{A}^*$ on $\mathcal{M}$.

Vice versa, if $\mathcal{M}$ is accepted by $\mathcal{A}^*$ then $\mathcal{M} \models \Box^*(\neg R \vee P)$, otherwise there would be an $s \in M$ with $\delta^*(q, L(s)) = \emptyset$ for any $q \in Q$, and the Spoiler could force a run to arrive in s, where the Duplicator would not be able to proceed the game. To prove that $\mathcal{M} \models \phi(P/R)$, just note that a winning strategy of the Duplicator on the game of $\mathcal{A}^*$ on $\mathcal{M}$ is also a winning strategy for the Duplicator on the game of $\mathcal{A}$ on $\mathcal{M}^{-P}$. Hence $\mathcal{M}^{-P} \models \phi(P/R)$ and since P does not appear in $\phi(P/R)$ we conclude that $\mathcal{M} \models \phi(P/R)$.

To prove the same for the uniform interpolant of ψ, notice that the operator $\hat{}$ on automata defined in theorem 4.5 preserves normal forms and monotonicity; hence to finish the proof we only need to apply this operator to the automaton $\mathcal{A}^*$.

$\Box$

Lemma 4.7 *If ϕ is monotone in P, then there exists an automaton in normal form which is equivalent to ϕ and monotone in P.*

Proof.

If ϕ is monotone in P consider the sentence $\psi = \Box^*(\neg R \vee P) \wedge \phi(P/R)$, where R is a new constant. We have $\models \psi \to \phi$: if not, there would be a process graph $\mathcal{S}$ such that $\mathcal{S} \models \psi \wedge \neg\phi$, and since we may suppose $\mathcal{S}$ connected, we have that $R^{\mathcal{S}} \subseteq P^{\mathcal{S}}$; by the monotonicity of ϕ from $\mathcal{S} \models \phi(P/R)$ it follows $\mathcal{S} \models \phi$, a contradiction. Hence $\models \psi \to \phi$ and any interpolant θ of this implication will be equivalent to ϕ: we have $\models \theta \to \phi$ and $\models \psi \to \theta$ by the properties of the interpolant; from $\models \psi \to \theta$ it follows $\models (\psi \to \theta)(R/P)$, that is, $\models \phi \to \theta$. If we could find an interpolant of $\models \psi \to \phi$ with an equivalent automaton which is in normal form and monotone in P, then we would have proved the lemma. But a uniform interpolant of ψ with respect to $\mathcal{L}(\psi) \setminus \{R\}$ is in particular an interpolant for the implication $\models \psi \to \phi$, hence the lemma follows from lemma 4.6.

$\square$

Theorem 4.8 *A monotone μ-sentence is equivalent to a positive one.*

Proof.
If ϕ is a monotone sentence, by lemma 4.7 we can find an equivalent automaton which is in normal form and monotone in P. By lemma 4.4 such an automaton is equivalent to a positive sentence, which in turn is equivalent to ϕ. $\square$

5 Łoś-Tarski Theorem

In this section we find a syntactic characterization of μ-sentences that are preserved under submodels. To this end, we extend the definition of the universal modal formulae, which are constructed without using the diamond operators $\Diamond_a$.

Definition 5.1 *The* **universal** *μ-formulae are those constructed from propositional constants, their negations, and variables using conjunctions, disjunctions, the $\Box_a$-operators and the fixed point operators μ, ν.*

Definition 5.2 *A μ-sentence ϕ is* **preserved under submodels** *if for all process graphs $\mathcal{M} \subseteq \mathcal{N}$ and $s \in M$, if $(\mathcal{N}, s) \models \phi$ then $(\mathcal{M}, s) \models \phi$.*

Lemma 5.3 *A universal μ-sentence is preserved under submodels.*

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

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

The desired result follows by applying this to a sentence ϕ. We only work out the cases of the least fixed point $\phi(X_1, \ldots, X_n) = \mu Y.\psi(X_1, \ldots, X_n, Y)$ and of the greatest fixed point $\phi(X_1, \ldots, X_n) = \nu Y.\psi(X_1, \ldots, X_n, Y)$.

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

  For a process graph $\mathcal{S}$, $s \in S$, and $S_1, \ldots, S_n \subseteq S$ we have by the properties of the fixed point operator μ that $s \in [\![\phi]\!]^{\mathcal{S}}_{[X_1:=S_1,\ldots,X_n:=S_n]}$ iff whenever $[\![\psi]\!]^{\mathcal{S}}_{[X_1:=S_1,\ldots,X_n:=S_n,Y:=Z]} \subseteq Z$ for $Z \subseteq S$ then $s \in Z$.

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

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

  hence

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

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

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

  For a process graph $\mathcal{S}$, $s \in S$, and $S_1, \ldots, S_n \subseteq S$ we have by the properties of the fixed-point operator ν that $s \in [\![\phi]\!]^{\mathcal{S}}_{[X_1:=S_1,\ldots,X_n:=S_n]}$ iff there exists $Z \subseteq S$ such that $Z \subseteq [\![\psi]\!]^{\mathcal{S}}_{[X_1:=S_1,\ldots,X_n:=S_n,Y:=Z]}$ and $s \in Z$.

  Suppose that $m \in [\![\phi]\!]^{\mathcal{N}}_{[X_1:=B_1,\ldots,X_n:=B_n]} \cap M$, and let $Z \subseteq N$ be such that $Z \subseteq [\![\psi]\!]^{\mathcal{N}}_{[X_1:=B_1,\ldots,X_n:=B_n,Y:=Z]}$, and $m \in Z$. Let $Z' = Z \cap M$. We have $Z' \subseteq [\![\psi]\!]^{\mathcal{N}}_{[X_1:=B_1,\ldots,X_n:=B_n,Y:=Z]} \cap M$, and by induction $Z' \subseteq [\![\psi]\!]^{\mathcal{M}}_{[X_1:=B_1\cap M,\ldots,X_n:=B_n\cap M,Y:=Z']}$. Since $m \in Z'$ we are done.

$\square$

In order to prove the converse of lemma 5.5, we find a property of μ-automata that corresponds to universal μ-sentences.

Definition 5.4 *An automaton $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ is* **universal** *if for all $q \in Q$, $\Sigma \subseteq \Sigma_p$, $D, D' \subseteq \Sigma_r \times Q$ such that $D \in \delta(q, \Sigma)$ and $D' \subseteq D$ it is true that $D' \in \delta(q, \Sigma)$.*

Lemma 5.5 *Let $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ be a μ-automaton in normal form. If $\mathcal{A}$ is universal, then the μ-sentence $\phi_{\mathcal{A}}$ corresponding to the automaton $\mathcal{A}$ is equivalent to a universal μ-sentence.*

Proof.
Suppose that the normal automaton $\mathcal{A}$ is universal. By induction on the partial ordering $\leq_{\mathcal{A}}$ of Q, we prove that α_q is a universal formula:

- if q is a leaf of $(Q, \leq_{\mathcal{A}})$ then for any $\Sigma \subseteq \Sigma_p$, $D \in \delta(q, \Sigma)$, $a \in \Sigma_r$, and $(a, q_1) \in D$ the state q_1 is a back-state, and the sentence β_{q,q_1} is equal to X_{q_1}. Since A is universal, we can prove that the formula

  $$\beta = \bigvee_{D\in\delta(q,\Sigma)} \alpha_\Sigma \wedge \bigwedge_{a\in\Sigma_r} \bigwedge_{(a,q_1)\in D} \Diamond_a X_{q_1} \wedge \Box_a (\bigvee_{(a,q_1)\in D} X_{q_1})$$

is equivalent to the universal formula

$$\gamma = \bigvee_{D \in \delta(q,\Sigma)} \alpha_\Sigma \wedge \bigwedge_{a \in \Sigma_r} \Box_a (\bigvee_{(a,q_1) \in D} X_{q_1}).$$

It is clear that β implies γ. For the converse, suppose that a disjunct $\alpha_\Sigma \wedge \bigwedge_{a\in\Sigma_r} \Box_a(\bigvee_{(a,q_1)\in D} X_{q_1})$ of γ holds in a state s of a process graph $\mathcal{M}$ under a valuation V of the free variables in γ. Consider the set D' defined by $D' = \{(a, q_1) \in D : \exists s'(sR_a s' \wedge (\mathcal{M}, V, s') \models X_{q_1})\}$. Then $(\mathcal{M}, V, s) \models \alpha_\Sigma \wedge \bigwedge_{a\in\Sigma_r} \left(\bigwedge_{(a,q_1)\in D'} \Diamond_a X_{q_1}\right) \wedge \Box_a(\bigvee_{(a,q_1)\in D'} X_{q_1})$, which is a disjunct of β since $\mathcal{A}$ is universal.

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

□

We prove now the converse of 5.5:

Lemma 5.6 *For any μ-sentence which is preserved under submodels there exists an equivalent universal automaton.*

Proof.
Consider a μ-sentence ϕ preserved under submodels and a μ-automaton $\mathcal{A} = (Q, q_0, \Sigma_p, \Sigma_r, \delta, \Omega)$ in normal form which corresponds to ϕ. From $\mathcal{A}$ we construct the automaton $\hat{\mathcal{A}} = (Q, q_0, \Sigma_p, \Sigma_r, \hat{\delta}, \Omega)$ by defining $\hat{\delta}(q, \Sigma) = \{D' : \exists D \in \delta(q, \Sigma), D' \subseteq D\}$. It is easy to verify that the automaton $\hat{\mathcal{A}}$ is in normal form, whenever $\mathcal{A}$ it is so. Moreover, $\hat{\mathcal{A}}$ is universal. Using the fact that the sentence ϕ is preserved under submodels, we now prove that a process graph $\mathcal{S}$ is accepted by $\mathcal{A}$ if and only if it is accepted by $\hat{\mathcal{A}}$, and this will conclude the proof.

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

For the converse we suppose w.l.o.g. that any state $q \in Q$ is nonempty, that is, there exists a process graph $\mathcal{S}_q$ which is accepted by the automaton $(\mathcal{A}, q)$ (which is the same as $\mathcal{A}$ except that the initial state is q). We also suppose w.l.o.g. that $\mathcal{S}_q$ has an $\mathcal{A}$-labeling l_q, and that $\mathcal{S}$ has an $\hat{\mathcal{A}}$-labeling $\hat{l}$. From $\mathcal{S}$ we construct a process graph $\mathcal{S}'$ in such a way that $\mathcal{S} \subseteq \mathcal{S}'$, $r^{\mathcal{S}'} := r^{\mathcal{S}}$, and $\mathcal{S}'$ has an $\mathcal{A}$-labeling. Since the automaton $\mathcal{A}$ corresponds to the sentence ϕ which is preserved under submodels, this will imply that $\mathcal{S}$ is accepted by $\mathcal{A}$.

- Construction of $\mathcal{S}'$. Since $\hat{\mathcal{A}}$ accepts $\mathcal{S}$ via the labeling $\hat{l}$, for any node s of S the set $\hat{D}(s) = \{(a, q') : \exists t(sR_a t \wedge \hat{l}(t) = q')\}$ belongs to $\hat{\delta}(\hat{l}(s), L(s))$. 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 $\mathcal{S}'$ as the disjoint union of $\mathcal{S}$ and all the $\mathcal{S}_q$, for $q \in Q$. The root of $\mathcal{S}'$ is the root of S, and if $(a,q) \in D_s \setminus \hat{D}(s)$, there is a new a-transition from s to the root $r^{\mathcal{S}_q}$ of $\mathcal{S}_q$.

We claim that $\mathcal{S}'$ is accepted by $\mathcal{A}$. Consider the function $l : S' \to Q$ defined as $l(s') = \hat{l}(s')$ if $s' \in S$, and $l(s') = l_q(s')$ if $s' \in S_q$. We prove that l is an $\mathcal{A}$-labeling on $\mathcal{S}'$, by showing that for all $s \in S'$

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

since the other conditions of a labeling are trivially satisfied.

Suppose first that $s \in S$. 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 $\delta(l(s), L(s))$ (D_s has been defined above). To prove that $D(s) \subseteq D_s$, notice that for any action a, $\{t : sR_a^{S'}t\} = \{t : sR_a^S t\} \cup \{r^{\mathcal{S}_q} : (a,q) \in D_s \setminus \hat{D}(s)\}$. If $(a,q') \in D(s)$, then there exists a $t \in S'$ such that $sR_a^{S'}t \wedge l(t) = q'$. If $sR_a^S t$, then $q' = l(t) = \hat{l}(t)$ and $(a,q') \in \hat{D}(s) \subseteq D_s$. If $t \in \{r^{\mathcal{S}_q} : (a,q) \in D_s \setminus \hat{D}(s)\}$, then $q' = l(t) = l_q(r^{\mathcal{S}_q}) = q$, with $(a,q) \in D_s \setminus \hat{D}(s)$. Hence $(a,q') \in D_s$.

Vice versa, if $(a,q) \in D_s$, suppose first that $(a,q) \in \hat{D}(s)$. In this case, since $\hat{D}(s) \subseteq D(s)$ as it is easily seen from the definitions of the two sets, we have $(a,q) \in D(s)$. If $(a,q) \in D_s \setminus \hat{D}(s)$, then $sR_a^{S'}r^{\mathcal{S}_q}$ and $l(r^{\mathcal{S}_q}) = q$, proving that $(a,q) \in D(s)$.

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

We are now ready to prove:

Theorem 5.7 *A μ-sentence which is preserved under submodels is equivalent to a universal sentence.*

Proof.
Consider a μ-sentence ϕ which is preserved under submodels and an automaton $\mathcal{A}$ corresponding to ϕ. By lemma 5.5 we may suppose that $\mathcal{A}$ is universal, and by lemma 5.6 $\mathcal{A}$ is equivalent to a universal sentence. $\square$

An analogous result for existential μ-sentences can be obtained easily by dualizing theorem 5.7: a sentence is preserved under extensions if and only if it is equivalent to an existential sentence (i.e. one that does not use the operator $\Box$). Nevertheless it is interesting to capture the class of automata which corresponds to existential sentences, but negation is not an easy operation on automata, hence the 'existential 'automata are not easily obtained from the universal ones.

Definition 5.8 *An automaton $\mathcal{A} = (Q, \Sigma_p, \Sigma_r, q_0, \delta, \Omega)$ in normal form is* **existential** *if for all $q \in Q$ there exists a true state q_T such that*

1. *for all* $\Sigma \subseteq \Sigma_p$, $D \in \delta(q, \Sigma)$, *and* $X \supseteq \{a \in \Sigma_r : \exists q' \ (a, q') \in D\}$, *the set* $D \cup \{(a, q_T) : a \in X\}$ *belongs to* $\delta(q, \Sigma)$;

2. *if* q *is not a true state then* q_T *is an immediate successor of* q *in the order* $\leq_{\mathcal{A}}$.

Notice that if q is a true state we can verify the two conditions above with $q_T = q$.

Lemma 5.9 *If the automaton* $\mathcal{A}$ *is in normal form and existential, then* $\phi_{\mathcal{A}}$ *is equivalent to an existential sentence.*

Proof.
We prove that for any $q \in Q$ the formula α_q of definition 5.10 is equivalent to an existential formula. This follows from Lemma 3.5 if q is a true state, since then α_q is eqivalent to $\top$. If q is not a true state, we show that for any q, Σ, the formula

$$B = \bigvee_{D \in \delta(q,\Sigma)} \bigwedge_{a \in \Sigma_r} \bigwedge_{(a,q_1) \in D} \Diamond_a \beta_{q,q_1} \wedge \Box_a \Big(\bigvee_{(a,q_1) \in D} \beta_{q,q_1} \Big)$$

is equivalent to the formula

$$C = \bigvee_{D \in \delta(q,\Sigma)} \bigwedge_{a \in \Sigma_r} \bigwedge_{(a,q_1) \in D} \Diamond_a \beta_{q,q_1}$$

and then the result can be proved by induction on the partial ordering $\leq_{\mathcal{A}}$.

It is obvious that B implies C. To prove the converse, suppose that (M, s) is a model for the language Σ_p, Σ_r and $\mathcal{V}$ is a valuation of the free variables in C such that $(M, s, \mathcal{V}) \models C$; then there exists $D \in \delta(q, \Sigma)$ such that $(M, s, \mathcal{V}) \models \bigwedge_{a \in \Sigma_r} \bigwedge_{(a,q_1) \in D} \Diamond_a \beta_{q,q_1}$. Let $X = \{a \in \Sigma_r : R_a(s) \neq \emptyset\}$. Then $\{a \in \Sigma_r : \exists q' \ (a, q') \in D\} \subseteq X$: if there exists q' such that $(a, q') \in D$, then $(M, s, \mathcal{V}) \models \Diamond_a(\beta_{q,q'})$, $R_a(s) \neq \emptyset$, and $a \in X$.

Denote by D_X the set $\{(a, q_T) : a \in X\}$, where q_T is a true state which satisfies the two conditions of definition 5.8. We claim that

$$(M, s, \mathcal{V}) \models \bigwedge_{a \in \Sigma_r} \bigwedge_{(a,q_1) \in D \cup D_X} \Diamond_a \beta_{q,q_1} \wedge \Box_a \Big(\bigvee_{(a,q_1) \in D \cup D_X} \beta_{q,q_1} \Big),$$

which is a disjoint of B since $\mathcal{A}$ is existential.

First we prove that $(M, s, \mathcal{V}) \models \bigwedge_{(a,q_1) \in D \cup D_X} \Diamond_a \beta_{q,q_1}$, for any $a \in \Sigma_r$. If $(a, q_1) \in D$, then $(M, s, \mathcal{V}) \models \Diamond_a \beta_{q,q_1}$ follows from $(M, s, \mathcal{V}) \models C$. If $(a, q_1) \in D_X$ then $q_1 = q_T$ and since q_T is an immediate successor of q we have $\beta_{q,q_T} = \alpha_{q_T}$, which is equivalent to $\top$ by Lemma 3.5. From $(a, q_1) \in D_X$ it follows $a \in X$, hence $R_a(s) \neq \emptyset$ and $(M, s, \mathcal{V}) \models \Diamond_a(\beta_{q,q_T})$.

Next we prove that for any $a \in \Sigma_r$,

$$(M, s, \mathcal{V}) \models \bigwedge_{a \in \Sigma_r} \Box_a \Big(\bigvee_{(a,q_1) \in D \cup D_X} \beta_{q,q_1} \Big).$$

Fix $a \in \Sigma_r$, and consider two cases:

1. $(D \cup D_X) \cap \{a\} \times Q^* = \emptyset$. In this case $a \notin X$, $R_a(s) = \emptyset$, and $(M, s, \mathcal{V}) \models \Box_a(\bot)$.

2. $(D \cup D_X) \cap \{a\} \times Q^* \neq \emptyset$. In this case $a \in X$: if $a \notin X$ then there exists q' such that $(a, q') \in D$, and since $\{a : \exists q' \ (a, q') \in D\} \subseteq X$ we get a contradiction. Since $a \in X$, $(a, q_T) \in D_X$, and the formula $\beta_{q,q_T} = \alpha_{q_T}$ is a disjoint of $\bigvee_{(a,q') \in D \cup D_X} \beta_{q,q'}$. But α_{q_T} is equivalent to $\top$ and so $\Box_a(\bigvee_{(a,q_1) \in D \cup D_X} \beta_{q,q_1})$ is true in s.

Lemma 5.10 *Let $\mathcal{A} = (Q, \Sigma_p, \Sigma_r, q_0, \delta, \Omega)$, $\mathcal{A}' = (Q', \Sigma_p, \Sigma_r, q_0, \delta', \Omega')$ be automata such that $Q \subseteq Q'$, $\Omega'(q) = \Omega(q)$, and $D' \cap (\Sigma_r \times Q) \in \delta(q, \Sigma)$, for all $q \in Q$ and $D' \in \delta'(q, \Sigma)$. Then if the model $\mathcal{M}$ has an $\mathcal{A}'$ labeling l', the restriction l of l' to Q is an $\mathcal{A}$-labeling for the submodel $\mathcal{N}$ based on the domain $\{s \in M : l'(s) \in Q\}$.*

Proof.
We just have to verify that for all $s \in N$, the set $D(s) = \{(a, q') \in \Sigma_r \times Q : \exists t \ (sR_a^{\mathcal{N}} t \wedge l(t) = q')\} \in \delta(l(s), L(s)\}$, since the other conditions for l to be a labeling are obviously verified. Since l' is an $\mathcal{A}'$ labeling, we have that the set $D'(s) = \{(a, q') \in \Sigma_r \times Q' : \exists t \ (sR_a^{\mathcal{M}} t \wedge l'(t) = q')\}$ belongs to $\delta'(l'(s), L(s)\}$. Moreover, $D(s) = D' \cap (\Sigma_r \times Q)$, hence it belongs to $\delta(l(s), L(s))$ by hypothesis. □

Lemma 5.11 *If the sentence ϕ is preserved under extensions, then there is an existential automaton $\mathcal{A}$ which corresponds to ϕ.*

Proof.
Let $\mathcal{A} = (Q, \Sigma_p, \Sigma_r, q_0, \delta, \Omega)$ be an automaton in normal form corresponding to the sentence ϕ.

From $\mathcal{A}$ we define the automaton $\mathcal{A}^* = (Q^*, \Sigma_p, \Sigma_r, q_0, \delta^*, \Omega^*)$ as follows: for any state q of $\mathcal{A}$ which is not a true state we introduce a new state q_T, and define $Q^* = Q \cup \{q_T : q \in Q \text{ is not a true state}\}$,

$$\delta^*(q, \Sigma) \quad = \quad \begin{cases} \delta(q, \Sigma) \cup \{D \cup D_X\} : X_D \subseteq X \subseteq \Sigma_r\} & \text{if } q \in Q \\ \{\{(a, q) : a \in X\} : X \subseteq \Sigma_r\} & \text{if } q \notin Q \end{cases}$$

where $D_X = \{(a, q_T) : a \in X\}$, $X_D = \{a : \exists q' \ (a, q') \in D\}$, $\Omega^*(q) = \Omega(q)$ for $q \in Q$, and $\Omega^*(q) = 0$, for $q \notin Q$.

Consider the partial ordering $\leq_{\mathcal{A}^*}$ defined as the transitive and reflexive closure of $\leq_{\mathcal{A}} \cup \{(q, q_T) : q \in Q \text{ is not a true state}\}$. With this order, the automaton $\mathcal{A}^\star$ is in normal form and existential, as it is easily verified.

We prove that a model $\mathcal{M}$ is accepted by $\mathcal{A}$ iff it is accepted by $\mathcal{A}^*$.

Suppose first that $\mathcal{M}$ is accepted by $\mathcal{A}$. Then, since $\delta(q, \Sigma) \subseteq \delta^*(q, \Sigma)$, we also have that $\mathcal{M}$ is accepted by $\mathcal{A}^*$.

Vice versa, if $\mathcal{M}$ is accepted by $\mathcal{A}^*$, we may suppose without loss of generality that $\mathcal{M}$ has a $\mathcal{A}$-labeling. Then $\mathcal{A}$ and $\mathcal{A}^*$ verify the hypothesis of Lemma 5.10

and there exists a submodel of $\mathcal{M}$ which is accepted by $\mathcal{A}$. Since the formula ϕ to which $\mathcal{A}$ is equivalent is preserved under extensions, $\mathcal{M}$ is accepted by $\mathcal{A}$. $\square$

6 Acknowledgement

The author wishes to thank Marco Hollenberg for reading and commenting an earlier version of this paper.

References

[1] G.D'Agostino, M. Hollenberg, *Uniform interpolation, automata and the modal μ-calculus*; Logic Group Preprint Series N0. 165 May 1996 , University of Utrecht.

[2] D. Janin, *Proprietes Logique du non determinisme et μ-calcul modal*; PhD thesis, January 1996, University of Bordeaux I.

[3] D. Janin and I. Walukiewicz, *Automata for the modal μ-calculus and related results*; In *Proceedings MFCS'95*, pages 552–562. Springer, 1995. `http://www.daimi.aau.dk/~igw/mfcs.ps`.

[4] D. Kozen, *Results on the propositional μ-calculus*, Theoretical Computer Science 27:234-241, 1983.