Comparing levels of Bisimulation Quantifiers Logic with the mu-calculus

Giovanna D'Agostino*, Giacomo Lenzi**

* Udine University, Department of Mathematics and Computer Science
Viale delle Scienze 206, 33100 Udine, Italy
dagostin@dimi.uniud.it

** LaBRI, Université de Bordeaux I- ENSERB
351 cours de la Libération,
F-33 405 Talence cedex lenzi@labri.u-bordeaux.fr

November 29, 2001

Abstract This paper deals with the logic BQL of bisimulation quantifiers, that is, Polimodal Logic extended with the existential bisimulation quantifier $\widetilde{\exists} P\phi$: this quantifier allows to look for a subset P satisfying $\phi(P)$ not only within the model, but also in any other model which is bisimilar to the given one. It is known that BQL has the same expressive power than the μ-calculus; here we refine this result by comparing the syntactical hierarchy of the μ-calculus with a naturally defined hierarchy for BQL, and prove that, although this hierarchy collapses at level three, its first two levels coincide with the first two levels of the μ-calculus hierarchy.

1 Introduction

In this paper we deal with the logic of bisimulation quantifiers BQL. This is a logic extending Propositional Dynamic Logic by means of *bisimulation quantifiers*: an existential bisimulation quantifier $\widetilde{\exists} P$ acts like a standard existential set quantifier but allows to look for P not only within the model, but in any other model which is bisimilar to the given one.

Bisimulation quantifiers were first considered in [3] and [9] as a tool for proving uniform interpolation for modal logic, and in [2] to prove uniform interpolation for the modal μ-calculus. Indeed, it can be proved that any logic which is invariant under bisimulation and closed under bisimulation quantifiers (that is: strong enough to express the bisimulation quantifiers within the logic) enjoys uniform interpolation, the uniform interpolant of a formula $\phi(P)$ being simply $\widetilde{\exists} P\phi(P)$. Now, it is easily shown that Propositional Dynamic Logic PDL is not closed under bisimulation quantifiers, and in [7] the closure of PDL under these quantifiers (that is: BQL) was proved to be equivalent in expressive power to the μ-calculus: the Kleene star of PDL and the quantifiers $\widetilde{\exists}, \widetilde{\forall}$ are enough to express the least and greatest fixed points of the μ-calculus, and the μ-calculus itself is closed under bisimulation quantifiers. BQL represents thus an alternative formalism to the μ-calculus, and it is hence a strong logic capable of expressing interesting second order properties like wellfoundedness, fairness and the like.

In this paper we look at the equivalence between BQL and the μ-calculus more closely, trying to define a hierarchy for BQL and to compare it with the fixed-point alternation depth hierarchy of the μ-calculus. We stratify in a natural way BQL and prove that the two hierarchies coincide up to level two: using alternating and nondeterministic modal automata for the μ-calculus we show that the first two levels of the μ-calculus hierarchy are closed under bisimulation quantifiers, and this will imply the equivalence between the two levels of the hierarchies. On the other hand, we see that a drastic divergence appears at higher levels, since the BQL-hierarchy collapses at level three, whereas the μ-calculus hierarchy is infinite: as a consequence, the whole μ-calculus is expressible using only three alternations of bisimulation quantifiers.

This correspondence between BQL and the μ-calculus is related to a similar correspondence between the μ-calculus and the bisimulation invariant fragment of monadic second order logic MSO. In [4] it is announced that the hierarchy induced on this fragment by the natural quantifiers alternation hierarchy of MSO also coincides up to level two with the μ-calculus hierarchy. Together with our result this implies that, although bisimulation quantifiers are in general more expressive than standard quantifiers, two alternations of bisimulation quantifiers over PDL can be simulated by two alternations of standard quantifiers. On the other hand, the inclusion of the whole μ-calculus in the third level of the bisimulation invariant fragment of MSO is still an open question, although this inclusion has been proved in [4] for the class of graphs of bounded degree and for the class of graphs of bounded tree-width. Maybe a solution for the general case can be found using BQL as an intermediate step between the bisimulation invariant fragment of the monadic second order logic MSO and the μ-calculus.

2 Bisimulation Quantifiers Logic

We start our investigation by describing the Logic of Bisimulation Quantifiers.

Definition 2.1 Bisimulation Quantifiers Logic BQL *is defined as the least set which contains a set of propositional constants* $Prop$ *and satisfy: if* $\phi, \psi \in BQL$ *and* $P \in Prop$*, then* $\phi \vee \psi$, $\neg\phi$, $\diamond\phi$, $\diamond^*\phi$, *and* $\widetilde{\exists} P\phi$ *belong to* BQL*.*

In order to give an appropriate semantics for BQL, we first recall the definition of bisimulation. In the following, we only consider structures for the language containing a unary predicate for each symbol in $Prop$, a binary predicate R, and a constant r, i.e. tuples of the form

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

Definition 2.2 *Let M, N be structures with D^M, D^N as respective domains. A relation $Z \subseteq D^M \times D^N$ is a* bisimulation *between M and N if $r^M Z r^N$, bisimilar points agree on propositional constants, and whenever wZv the following holds: if $wR^M w'$, then there exists a v' such that $vR^N v'$ and $w'Zv'$;*
if $vR^N v'$, then there exists a w' such that $wR^M w'$ and $w'Zv'$.
Two structures M, N are bisimilar *(notation: $M \sim N$) if there exists a bisimulation between them.*

Definition 2.3 (BQL semantics) *If M is a structure, the relation $M \models \phi$ is defined inductively as follows:*

$$M \models P_i \Leftrightarrow r^M \in P_i^M; \quad M \models \neg\phi \Leftrightarrow M \not\models \phi; \quad M \models \phi \vee \psi \Leftrightarrow M \models \phi \text{ or } M \models \psi;$$

$$M \models \Diamond\phi \Leftrightarrow \exists w \text{ such that } (r^M, w) \in R^M \text{ and } (M, w) \models \phi;$$

$$M \models \Diamond^*\phi \Leftrightarrow \exists w_1, \ldots, w_n \text{ with } w_1 R^M \ldots R^M w_n \text{ and } (M, w_n) \models \phi;$$

$$M \models \widetilde{\exists} P\phi \Leftrightarrow \exists N, \exists P \subseteq D^N \text{ with } N \sim_{-P} M \text{ and with } (N, P) \models \phi,$$

where (M, w) is like M excepts for the root, which is now w, the relation $\sim_{-P}$ stands for bisimulation w.r.t. the language $Prop \setminus \{P\}$ (i.e. bisimilar points do not need to agree on P), and (N, P) stands for the structure which is like N except for the propositional constant P that receives now a new interpretation (denoted again by P).

The operators $\wedge, \rightarrow, \Box, \Box^\star$ are defined as usual, and the universal bisimulation quantifier $\widetilde{\forall}$ is defined as the dual of the existential one:

$$M \models \widetilde{\forall} P\phi \Leftrightarrow \text{ for all } N, \text{ if } N \sim_{-P} M \text{ and } P \subseteq D^N \text{ then } (N, P) \models \phi.$$

The importance we give to BQL is justified by its equivalence with a well known and widely used logic with fixed point operators: the μ-calculus.

Definition 2.4 *The* μ-calculus *μ is defined as the least set which contains a set of propositional constants $Prop$, their negations $\{\neg P : P \in Prop\}$, a set of variables X, and satisfies: if $\phi, \psi \in \mu$ then $\phi \vee \psi$, $\Diamond\phi$, $\mu X.\phi$, and $\nu X.\phi$ belong to μ.*

The semantics of a μ-calculus formula is defined as above for the propositional constants and connectives, while $\mu X.\phi$ and $\nu X.\phi$ are interpreted respectively as the least and the greatest fixed point of the operator induced by the formula $\phi(X)$ (for a precise definition see [8]).

Our investigation starts from the following results:

Theorem 2.5 ([7]) *BQL and the μ-calculus have the same expressive power.*

We are here interested in a finer comparison between BQL and the μ-calculus; more precisely, we consider the fixpoint alternation-depth hierarchy of the μ-calculus, which is known to stratify the logic into non collapsing levels:

Definition 2.6 *The* fixpoint alternation-depth hierarchy *of the μ-calculus is the sequence $N_0 = M_0, N_1, M_1, \ldots$ of sets of μ-formulas defined inductively as follows. The first level $N_0 = M_0$ is defined as the set of all modal fixpoint free formulas. For each k, level N_{k+1} (level M_{k+1}, respectively) is defined as the closure of $N_k \cup M_k$ under substitution (provided no free variable becomes bounded during the process) and greatest fixed point operator (least fixed point operator, respectively).*

We now try to see whether there exists a natural hierarchy for BQL, and if so, whether we can compare it with the μ-calculus hierarchy defined above. One could think of stratifying BQL starting from PDL (that is, the fragment of BQL which does not use the fixpoint operators μ, ν) and then simply counting the alternation of set quantifiers $\widetilde{\exists} P, \widetilde{\forall} P$, as it is done e.g. in Monadic Second Order Logic MSO. Notice however that in this way we would stratify only formulas in "prenex normal form", but since not all BQL-operators commute with bisimulation quantifiers (e.g. $\Box^*(\widetilde{\exists} P\phi)$ does not imply $\widetilde{\exists} P\Box^*(\phi)$), we don't have an easy way to put a BQL-formula in prenex normal form.

Another possibility is given by the following definition:

Definition 2.7 *We stratify BQL in levels by defining the subsets $\widetilde{\Sigma}_n, \widetilde{\Pi}_n$ as follows:*

1. *$\widetilde{\Sigma}_0 = \widetilde{\Pi}_0$ are the BQL-formulas constructed from $Prop \cup \{\neg P : P \in Prop\}$ by using conjunctions, disjunctions, and the modal operators $\Box$ and $\Diamond$. In other words, both zero-levels coincide with modal logic.*
2. *$\widetilde{\Sigma}_{n+1}$ is the closure of $\widetilde{\Sigma}_n \cup \widetilde{\Pi}_n$ under the operations described in $(a), (b)$ below.*
   (a) *Positive Substitution: if $\phi(P_1, \ldots P_n), \phi_1, \ldots, \phi_n$ are in $\widetilde{\Sigma}_{n+1}$, then $\phi(\phi_1 \ldots \phi_n)$ is in $\widetilde{\Sigma}_{n+1}$, provided $P_1, \ldots, P_n$ are positive in ϕ (i.e. they occur under an even number of negations) and no occurrence of a proposition which was free in one of the ϕ_i becomes bounded in $\phi(\phi_1 \ldots \phi_n)$.*
   (b) *$\Box^*$, $\widetilde{\exists} P$: if ϕ is in $\widetilde{\Sigma}_{n+1}$, then $\Box^*\phi$, $\widetilde{\exists} P\phi$ are in $\widetilde{\Sigma}_{n+1}$.*
3. *$\widetilde{\Pi}_{n+1}$ is the closure of $\widetilde{\Sigma}_n \cup \widetilde{\Pi}_n$ under positive substitution and the operators $\Diamond^*$, $\widetilde{\forall} P$.*

Some natural questions about this definition are the following.

1. Is this hierarchy strict?
2. Can we compare BQL-levels with the μ-calculus levels?

3 Comparing the First Two Levels

The first thing we prove is that any formula of N_k (M_k) can be translated into a $\widetilde{\Sigma}_k$-formula ($\widetilde{\Pi}_k$-formula, respectively) with the same meaning.

Lemma 3.1 *There exists a translation $\phi^\bullet$ from μ over the language $Prop \cup Var$ to BQL over $Prop' = Prop \cup Var$ such that:*

(a) ϕ is equivalent to $\phi^\bullet$;

(b) the propositions having at least a non-bounded occurrence in $\phi^\bullet$ are the propositions and the free variables of ϕ;

(c) if $\phi \in N_k$ ($\phi \in M_k$) then $\phi^\bullet \in \widetilde{\Sigma}_k$ ($\phi^\bullet \in \widetilde{\Pi}_k$, respectively).

Proof.
We define the translation $\phi^\bullet$ by induction on the levels N_k, M_k of the alternation depth hierarchy. If $k = 0$ then $\phi^\bullet = \phi$. Suppose we already defined the translation from N_k to $\widetilde{\Sigma}_k$, and from M_k to $\widetilde{\Pi}_k$; we extend it to the $k+1$-levels following the inductive definitions of the levels. If $\psi \in N_{k+1}$ is equal to $\phi(\phi_1, \ldots, \phi_r)$ and we already have a translation for $\phi, \phi_1, \ldots, \phi_r \in N_{k+1}$ into formulas $\phi^\bullet, \phi_1^\bullet, \ldots, \phi_r^\bullet$ in $\widetilde{\Sigma}_{k+1}$, then $\phi^\bullet$ is defined as $\phi^\bullet(\phi_1^\bullet, \ldots, \phi_r^\bullet)$.
If $\psi \in N_{k+1}$ is equal to $\nu X.\phi$, then define

$$\psi^\bullet = \widetilde{\exists} X(X \wedge \Box^*(\neg X \vee \phi^\bullet)).$$

Let us prove that the formula $\psi^\bullet$ is equivalent to ψ.

Suppose $M \models \nu X.\phi$; then there exists a subset A of the domain of M containing the root such that $A \subseteq \phi(A)$ (by $\phi(A)$ we mean the set of points which make the formula ϕ true under the assignment A to X). By the inductive hypothesis we know that $\phi(A) = \phi^\bullet(A)$, hence $M \models \exists X(X \wedge \Box^*(\neg X \vee \phi^\bullet))$. Since any model is bisimilar to itself we obtain $M \models \widetilde{\exists} X(X \wedge \Box^*(\neg X \vee \phi^\bullet))$.

On the other hand, if $M \models \widetilde{\exists} X(X \wedge \Box^*(\neg X \vee \phi^\bullet))$, let $N \sim_{-X} M$ and $X \subseteq D^N$ be such that $(N, X) \models X \wedge \Box^*(\neg X \vee \phi^\bullet)$. Without loss of generality, we may suppose that any point in N is accessible from the root. Then there exists a set $A \subseteq N$ containing the root such that $A \subseteq \phi^\bullet(A) = \phi(A)$. It follows that $N \models \nu X.\phi(X)$, and from $N \sim_{-X} M$ we obtain $M \models \nu X.\phi(X)$.

In the same way one defines the translation on M_{k+1}, e.g.

$$(\mu X.\phi)^\bullet = \widetilde{\forall} X(\Diamond^*(\neg X \wedge \phi^\bullet) \vee X).$$

$\Box$

3.1 Alternating Modal Automata and Systems of Fixed Point Equations

Our next task is to show that the there exists an inverse translation from $\widetilde{\Sigma}_k$ to N_k and from $\widetilde{\Pi}_k$ to M_k, but only for $k = 1, 2$. This will be easily proved once we have shown that N_1 and N_2 are closed under bisimulation quantifiers. To prove this we first recall some results about parity automata which are well known for automata running over trees of bounded degrees; for the sake of completeness we present here the results in the case of arbitrary trees. We define alternating modal automata on arbitrary trees by following closely [6], but we simplify the definition as, unlike that paper, we are not interested in the comparison with Monadic Second Order Logic and we only consider structures with one binary relation (i.e. graphs) rather than several binary relations.

Let k be a natural number. An alternating modal automaton of index k is a tuple:

$$A = \langle Q, \Sigma, q_0, \delta, \Omega \rangle,$$

where:

- Q is a finite set of states;
- Σ is a finite alphabet (the powerset of a finite subset of $Prop$);
- $q_0 \in Q$ is the initial state;
- $\Omega : Q \to \{0, 1, \ldots, k-1\}$;
- δ is a function which associates to every $q \in Q$ and $\sigma \in \Sigma$ a formula $\delta(q, \sigma)$ of first order logic without equality, which is positive over the signature consisting of one binary predicate $q(x)$ for each $q \in Q$.

Acceptance is defined by a game, as usual. Let T be a Σ-valued tree (that is: a letter $T(s) \in \Sigma$ is associated to each node s of the tree, saying which propositions are true in s). Let A be an automaton. The game $G(T, A)$ is defined as follows. The initial position is the pair (r, q_0), where r is the root of the tree. If the current position is a pair (s, q) then player I moves. He chooses a "marking" binary relation m between the sons $Succ(s)$ of s and states of Q, in a way to satisfy $\delta(q, T(s))$ on the set $Succ(s)$ (by identifying $m \subseteq Succ(s) \times Q$ with the interpretation $\{q^m : q \in Q\}$ for the language $\{q(x) : q \in Q\}$, where $q^m = \{s \in Succ(s) : (s, q) \in m\}$). Now, player II chooses one pair $(s', q) \in m$ which becomes the current position.

If one player cannot move then the other wins. If the play is infinite, then as the result we have an infinite sequence

$$(s_0, q_0), m_1, (s_1, q_1), m_2, \ldots$$

Let j the greatest number appearing infinitely often in the sequence

$$\Omega(q_0), \Omega(q_1), \Omega(q_2), \ldots$$

Player I wins if j is even, otherwise player II is the winner [1].

We say that T is *accepted* by A if there is a winning strategy for player I in the game $G(T, A)$. The *tree language* defined by A is the set of all trees accepted by A, and it can be proved that this set is closed under bisimulation.

Nondeterministic modal automata are alternating modal automata in which any transition formula $\delta(q, \sigma)$ is a disjunction of formulas of the form

$$\exists x_1 \dots \exists x_k q_1(x_1) \wedge \dots \wedge q_k(x_k) \wedge \forall z (q_1(z) \vee \dots \vee q_k(z)).$$

The acceptance condition of nondeterministic modal automata can be simplified over ω-expanded trees: *an ω-expanded tree* is a tree T in which for every $s, s' \in T$ if $s' \in Succ(s)$ then there are ω distinct successors of s which are bisimilar to s'.

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

1. *$l(r^T) = q_0$;*
2. *If $l(s) = q$ then the marking $\{(t, l(t)) : t \in Succ(s)\}$ satisfies the B-transition formula $\delta_B(q, T(s))$;*
3. *For any infinite path $r^M = s_0, s_1, \dots$ in T:*

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

   is even.

One can prove that a nondeterministic automaton B accepts an ω-expanded tree iff player I has a winning strategy in which he is allowed only to play *total and disjoint* markings, that is, markings m where for any node s there is exactly one q such that $(s, q) \in m$. From this it follows (see [7], Lemma 3.4.3 and Lemma 3.4.4):

Lemma 3.3 *An ω-expanded tree is accepted by a nondeterministic modal automaton B iff it has a B-labeling.*

Notice that the restriction to ω-expanded trees is not essential, because it can be easily proved that any tree T is bisimilar to an ω-expanded one, its ω-expansion T^ω (see for example [6]).

Alternating modal automata are closed under projection, as shown for example in [10], while nondeterministic modal automata are not closed over arbitrary trees. However, we can prove that nondeterministic modal automata are closed under projection over ω-expanded trees:

[1] In [5] the parity condition is stated in term of the smallest number appearing infinitely often. The two definitions are obvioulsy equivalent, but, since our principal reference for the missing proofs will be [1], we decided to follow the conventions of this book

Lemma 3.4 *If B is a nondeterministic modal automaton and P is a propositional constant, then there exists a nondeterministic modal automaton C over the alphabet $\Sigma_C = \{\sigma \in \Sigma_B : P \notin \sigma\}$, having the same parity index as B, such that for any ω-expanded tree T it holds:*

$$C \text{ accepts } T \Leftrightarrow \exists P \subseteq T \text{ such that } B \text{ accepts } T' = (T, P),$$

where we identify the propositional constant P with its interpretation in T'.

Proof.
If $B = \langle Q, \Sigma, q_0, \delta_B, \Omega \rangle$, consider the nondeterministic modal automaton $C = \langle Q, \Sigma', q_0, \delta_C, \Omega \rangle$, where $\Sigma' = \{\sigma \in \Sigma : P \notin \sigma\}$, and for all $q \in Q$ and $\sigma \in \Sigma'$,

$$\delta_C(q, \sigma) = \delta_B(q, \sigma) \vee \delta_B(q, \sigma \cup \{P\}).$$

We first prove: if T is a Σ_C-tree and $P \subseteq T$ is such that $T' = (T, P)$ is accepted by B then T is accepted by C. Consider the ω-expansion $(T')^\omega$ of T'. $(T')^\omega$ is accepted by B, because it is bisimilar to T'; by Lemma 3.3, it has a B-labeling l. But, as one can easily check, l is also a C-labeling for the same tree in which we don't consider the interpretation of P; this tree is hence accepted by C and bisimilar to T. This implies that T is accepted by C.

Conversely, suppose T is an ω-expanded Σ_C-tree, which is accepted by C. We want to define a subset P of T in such a way that the automaton B accepts the tree (T, P). Since T is ω-expanded and accepted by C, there exists a C-labeling $l : T \to Q$. Hence, for any $s \in T$, the marking $\{(t, l(t)) : t \in Succ(s)\}$ satisfies the formula

$$\delta_C(l(s), T(s)) = \delta_B(l(s), T(s)) \vee \delta_B(l(s), T(s) \cup \{P\}).$$

Define the subset P by putting $s \in P$ iff the marking $\{(t, l(t)) : t \in Succ(s)\}$ satisfies the disjoint $\delta_B(l(s), T(s) \cup \{P\})$. It follows that l is also a B-labeling over the tree (T, P), which is hence accepted by B. $\square$

In [6] it is proved that the modal μ-calculus and alternating modal automata have the same expressive power over trees, and, moreover, that any alternating modal automaton is equivalent to a nondeterministic one. A refinement of this result, which compares levels of the μ-calculus with alternating modal automata of fixed index, will be (re)proved in the following, together with a simulation of alternating modal automata of index 1 or 2 with nondeterministic modal automata of the same index [2]. This final result is what is needed in Section 3.3 to compare the first two BQL-levels with the first two levels of the μ-calculus.

To prove all this we borrow the technique of systems of equations from [1]. We give here only a short presentation of the results needed, adapted to our needs, referring to the book for further details.

[2]these results are already well known for automata running on binary trees, but to our knowledge the general case is not written in the literature.

A *system Eq of fixed point equations* in n equations and n variables is a sequence of equations of the form

$$\begin{aligned} x_1 &\stackrel{\theta_1}{=} f_1(x_1, \ldots, x_n) \\ &\vdots \\ x_n &\stackrel{\theta_n}{=} f_n(x_1, \ldots, x_n) \end{aligned}$$

where the f_i's are μ-formulas in $Prop \cup \{x_1, \ldots, x_n\}$ which are positive in $x_1, \ldots x_n$, and θ_i is either μ or ν. Its *solution* is a n-tuple $(e_1 \ldots, e_n)$ of μ-sentences defined inductively as follows. If $n = 1$, the solution of $x =^\theta f(x)$ is the μ-sentence $e_1 = \theta x.f(x)$. If we have more than one equation, we first solve the first equation w.r.t. x_1, obtaining the μ-formula $g_1(x_2, \ldots, x_n) = \theta_1 x_1.f_1(x_1, \ldots, x_n)$; we then consider the system of $n-1$ equations and $n-1$ variables obtained from the original one by deleting the first equation and by substituting the variable x_1 with $g_1(x_2, \ldots, x_n)$ in all the remaining equations. If the solution of this latter system is $(e_2, \ldots, e_n)$, then the solution of the original system is the n-tuple

$$(e_1 = g_1(e_2, \ldots, e_n), e_2, \ldots, e_n).$$

A system of equations is called an *elementary guarded* system whenever the μ- formulas $f_1, \ldots, f_n$ are elementary guarded, that is: $f_i = \bigvee_{\sigma \in \Sigma} \hat{\sigma} \wedge f_{i,\sigma}$, where Σ is the powerset of all propositional constants appearing in the f_i's,

$$\hat{\sigma} = \bigwedge_{p \in \sigma} p \wedge \bigwedge_{p \notin \sigma} \neg p,$$

and $f_{i,\sigma}$ is a positive boolean combination of formulas $\Box(t), \Diamond(t)$, where t is $\top, \bot$, or a positive boolean combination of variables in $\{x_1, \ldots, x_n\}$.

It can be proved that any system of equations is equivalent to an elementary guarded system.

3.1.1 Systems of Equations and Levels of the μ-Calculus

Systems of equations corresponds to levels of the μ-calculus in the following sense.[3] Given a sentence ϕ in N_k, there exists an elementary guarded system of equation $Eq(\phi)$

$$\begin{aligned} x_1 &\stackrel{\theta_1}{=} f_1(x_1, \ldots, x_n) \\ &\vdots \\ x_n &\stackrel{\theta_n}{=} f_n(x_1, \ldots, x_n) \end{aligned}$$

such that

[3] This and the following section are taken essentially from [1], with the difference that in that book the results are stated for vectorial fixed point terms instead that for systems of equations. The two formalisms are proved equivalent in the first chapter of the book, and using this equivalence one can easily transfer results from one formalism to the other.

- $\theta_n = \nu$, and the sequence $\theta_n, \dots, \theta_1$ contains $k-1$ alternations of ν, μ; (in this case, the number k is called the *index* of the system);
- the sentence ϕ is equivalent to e_1, where $(e_1, \dots, e_n)$ is the solution of the system.

Conversely, if Eq is such a system, then the sentence e_1 (and all e_i's) belongs to N_k.

3.1.2 Systems of Equations and Alternating Modal Automata

Systems of fixed point equations are essentially a disguised form of automata. The correspondence is achieved as follows. First, given a guarded system of equations

$$\begin{aligned} x_1 &\stackrel{\theta_1}{=} \bigvee_{\sigma \in \Sigma} \hat{\sigma} \wedge f_{1,\sigma} \\ &\vdots \\ x_n &\stackrel{\theta_n}{=} \bigvee_{\sigma \in \Sigma} \hat{\sigma} \wedge f_{n,\sigma} \end{aligned}$$

with solution $(e_1, \dots, e_n)$, we produce an automaton $A(Eq) = \langle Q, \Sigma, q_0, \delta, \Omega \rangle$ recognizing the class of trees in which e_1 is true.

The states Q of our automaton $A(Eq)$ are the variables $\{x_1, \dots, x_n\}$ of the system, while the formulas f_i are used to define the transition formulas of the automaton as follows. By translating modal logic into first order logic in the standard way, for each formula $f_{i,\sigma}$ of the system above we have a f.o. formula $d_{i,\sigma}$ over the first order signature having a unary predicate for any variable of the system (i.e. state of the automaton), such that for any Σ-valued tree T and for any node v of T, v satisfies $f_{i,\sigma}$ if and only if the successors of v satisfy $d_{i,\sigma}$ (e.g. $f_{i,\sigma} = \Box(x_1 \vee x_2)$, then $d_{i,\sigma} = \forall x(x_1(x) \vee x_2(x)))$. We define the transition formula of the automaton $A(Eq)$ by $\delta(x_i, \sigma) = d_{i,\sigma}$.

The priority function Ω of $A(Eq)$ is defined as follows. First divide the set of variables $\{x_n, \dots, x_1\}$ into the minimum number of adjacent sets $X_1, \dots, X_k$ with the same θ, i.e.: $x_n \in X_1$, if $x_j \in X_h$ and $j \geq i$ then

$$x_i \in X_h \Leftrightarrow \theta_r = \theta_j, \forall r \in \{i, \dots, j\},$$

and $x_{i-1} \in X_{h+1}$ if $x_i \in X_h$ and $\theta_{i-1} \neq \theta_i$. If $\theta_n = \nu$, let $\Omega(x) = h - 1$ for $x \in X_h$, while if $\theta_n = \mu$, let $\Omega(x) = h$. The initial state of the automaton will be $q_0 = x_1$.

It is possible to prove that in this way we obtain an alternating modal automaton with index k which is equivalent to the solution e_1 of the system Eq (for a proof of the equivalence, we refer to [1]).

Conversely, let $A = \langle Q, \Sigma, q_0, \delta, \Omega \rangle$ be an automaton. We will build a system of equations $Eq(A)$, with index equal to the index of the automaton, such that one of the solutions e_i of the system (depending on the initial state of A) is equivalent to the automaton A. The set of variables of the system will be indexed by the states of automaton, while the transition function δ and the priority function Ω are used to obtain a system of fixpoint equations as follows.

First notice that any first order formula positive over the signature $\{q(x) : q \in Q\}$, is equivalent to a positive boolean combination of formulas of type

$$A = q_1(x) \wedge \ldots \wedge q_h(x), \quad B = q_1(x) \vee \ldots \vee q_r(x),$$

$$C = \exists x(q_1(x) \wedge \ldots \wedge q_s(x)), \quad D = \forall x(q_1(x) \vee \ldots \vee q_t(x)).$$

This can be easily proved by induction on the structure of the positive first order formulas. The case of atomic formulas, conjunctions, and disjunctions is obvious. If F is a positive boolean combination of formulas of type A, B, C, D, then, by writing F as a disjunctions of conjunctions of these formulas, we see that $\exists zF$ can be put in this form by distributing $\exists z$, and similarly for $\forall zF$.

Given $q \in Q$ and $\sigma \in \Sigma$, the transition $\delta(q, \sigma)$ is a positive *sentence* over the above signature, and is thus equivalent to a positive boolean combination of existential and universal sentences of type C, D above.

Once the transition functions $delta(q, \sigma)$ are put in this form, for any $q \in Q$ and $\sigma \in \Sigma$ it is possible to find a modal sentence $f_{q,\sigma}$ with the property that for any Σ-valued tree T and for any node v of T, v satisfies $f_{q,\sigma}$ if and only if the successors of v satisfy $d_{q,\sigma}$. Finally, fix an order $\{q_1, \ldots, q_n\}$ of the set of states of the automaton A such that $\Omega(q_i) \geq \Omega(q_j)$ if $i < j$, and let $\theta_i = \mu$, if $\Omega(q_i)$ is odd, $\theta_i = \nu$, if $\Omega(q_i)$ is even. The system of equation $Eq(A)$ is then defined as

$$\begin{array}{rcl} x_1 & \stackrel{\theta_1}{=} & \bigvee_{\sigma \in \Sigma} \hat{\sigma} \wedge f_{q_1,\sigma} \\ & \vdots & \\ x_n & \stackrel{\theta_n}{=} & \bigvee_{\sigma \in \Sigma} \hat{\sigma} \wedge f_{q_n,\sigma}. \end{array}$$

If the variable corresponding to the initial state q_0 of the automaton A is x_i, it is possible to prove that the solution e_i of the system $Eq(A)$ is equivalent to A (again, we refer to [1] for a proof).

From the correspondence between μ-formulas and systems and between systems and automata, we get:

Theorem 3.5 *Levels N_k of the μ-calculus hierarchy correspond in expressive power to alternating modal automata of index k.*

3.2 Intersection Freeness and Nondeterministic Modal Automata

The goal of this section is to prove the following lemma, which will allow us in Section 3.3 to compare levels of the μ-calculus hierarchy with levels of BQL.

Lemma 3.6 *Any μ-calculus formula is equivalent to a nondeterministic automaton B. If the formula is in N_1 or N_2, then B can be chosen to have index 1 or 2, respectively.*

To prove this result we shall use a general result of [1] about intersection freeness, and apply this result to the modal language.

3.2.1 Intersection Free Fixed Point Terms

In [1] a general definition of fixed point term based on a signature F of functional symbols is given, together with a definition of a semantics over complete lattices (but, for our purpose, we can restrict the interpretations to powerset interpretations). For fixed point terms over a signature F one can define a syntactical notion of hierarchy and prove the equivalence between terms in the k-level and the solutions of system of equations of index k. Indeed, the results outlined in Section 3.1.1 are just an application of this correspondence to the modal language, because one can consider any μ-calculus formula as a fixed point term over the signature $Prop \cup \{\neg P : P \in Prop\} \cup \{\Box, \Diamond\} \cup \{\wedge, \vee\}$, where any symbol in $Prop \cup \{\neg P : P \in Prop\}$ is a functional symbol of arity 0, $\Box$ and $\Diamond$ are symbols of arity 1, and $\wedge, \vee$ are symbols of arity 2, with their standard interpretation over powersets.

The general result we need from [1] (in a simplified version, which is enough for our purposes) concerns signatures F containing the binary symbols $\wedge, \vee$, to be interpreted in the standard way. Over these signatures one can try to eliminate the intersection $\wedge$ in the following sense.

An *intersection-free elementary guarded term* over a signature F containing the binary symbols $\wedge, \vee$ and the constants symbols $\top, \bot$ is a disjunction of terms of the form $f(t_1, \ldots, t_n)$, where $f \in F \setminus \{\wedge, \vee\}$ and $t_1, \ldots, t_n$ are $\top, \bot$ or disjunctions of variables.

Theorem 3.7 Under certain conditions *on the interpretation of functions (to be presented shortly) each fixed point term is equivalent to the solution e_1 of a system of equations*

$$\begin{aligned} x_1 &\stackrel{\theta_1}{=} f_1(x_1, \ldots, x_n) \\ &\vdots \\ x_n &\stackrel{\theta_n}{=} f_n(x_1, \ldots, x_n), \end{aligned}$$

with the additional property that the terms f_i are intersection-free elementary guarded. Moreover, if the term is in the first or second level of the hierarchy, then the corresponding system has index 1 or 2, respectively.

The conditions needed to obtain this result can be explained as follows: for any pair f, g of functional symbols in $F \setminus \{\wedge, \vee\}$, one need to prove the existence of an intersection-free elementary guarded term $t_{f,g}$ that *brings the intersection inside*, i.e. such that the term $f(x_1, \ldots, x_n) \wedge g(y_1, \ldots, y_m)$ is equivalent to $t_{f,g}(r_1, \ldots, r_s)$, where the r_i are conjunctions of some x_i's and some y_j's. This is (a simplification of) what is called a Γ-rule in Chapter 9 of [1].

3.2.2 A Γ-rule for the μ-calculus

Our next task is to find a Γ rule for the μ-calculus, so that, by applying Theorem 3.7 to this special case, the system of equations corresponding to a μ-term in N_1 or N_2 will be simplified. But no Γ-rule exists for the original μ-calculus signature $\mu = \mathit{Prop} \cup \{\neg P : P \in \mathit{Prop}\} \cup \{\Box, \Diamond\} \cup \{\wedge, \vee\}$, and to proceed we need to change the signature of the modal language. More precisely, we consider for each natural n and $\sigma \in \Sigma$ a n-ary functional symbol $\{-, \ldots, -\}^\sigma_n$ to be interpreted as: $\{\}^\sigma_0 = \hat{\sigma} \wedge \Box(\bot)$ and, if $n > 0$,

$$\{x_0, \ldots, x_{n-1}\}^\sigma_n = \hat{\sigma} \wedge \Diamond(x_0) \wedge \ldots \wedge \Diamond(x_{n-1}) \wedge \Box(x_0 \vee \ldots \vee x_{n-1}),$$

where, as usual,

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

We then prove that a Γ-rule exists for the μ-calculus seen as a calculus of fixed point terms over the signature

$$\hat{\mu} = \{\top, \bot, \wedge, \vee\} \cup \{\{-, \ldots, -\}^\sigma_n : n \in N, \sigma \in \Sigma\}.$$

This signature is clearly equivalent to the original one, since we have:

$$p = \bigvee_{p \in \sigma} \{\}^\sigma_0 \vee \{\top\}^\sigma_1, \qquad \neg p = \bigvee_{p \notin \sigma} \{\}^\sigma_0 \vee \{\top\}^\sigma_1,$$

$$\Diamond(x) = \bigvee_{\sigma \in \Sigma} \{x, \top\}^\sigma_2, \qquad \Box(x) = \bigvee_{\sigma \in \Sigma} \{\}^\sigma_0 \vee \{x\}^\sigma_1.$$

Lemma 3.8 *If $x_0, \ldots, x_{n-1}, y_0, \ldots y_{m-1}$ are variables and $\sigma \neq \tau$ then*

$$\{x_0, \ldots, x_{n-1}\}^\sigma_n \wedge \{y_0, \ldots, y_{m-1}\}^\tau_m = \bot,$$

while

$$\{x_0, \ldots, x_{n-1}\}^\sigma_n \wedge \{y_0, \ldots, y_{m-1}\}^\sigma_m = \bigvee_R \{x_i \wedge y_j : (i, j) \in R\}^\sigma_{|R|}.$$

where the disjunction is taken over all binary relations $R \subseteq \{0, \ldots, n-1\} \times \{0, \ldots m-1\}$ which are surjective and total. Hence, there exists a Γ-rule for the signature $\hat{\mu}$.

Proof.
By inspecting the meanings of the formulas. $\Box$

We are now ready to prove:

Lemma 3.9 *Any μ-calculus formula ϕ is equivalent to a nondeterministic automaton B. If the formula is in N_1 or N_2, then B can be chosen to have index 1 or 2, respectively.*

Proof.
Consider any μ-calculus formula as a fixed point term over the signature $\hat{\mu}$; since we proved in Lemma 3.8 that a Γ-rule exists for $\hat{\mu}$, Theorem 3.7 shows that any μ-calculus formula ϕ is equivalent to a system of equations $Eq(\phi)$ in which the equations have the form:

$$x_i \stackrel{\theta_i}{=} f_i(x_1, \ldots, x_n),$$

for intersection free elementary guarded terms f_i over the signature $\hat{\mu}$. This means that each f_i is a disjunction of terms of the form $\{D_1, \ldots, D_n\}_n^\sigma$, where each D_i is a disjunction of variables, $\perp$ or $\top$. But we can further simplify the equations: since $\{x_0 \vee x_1, x_2, ..., x_n\}_n^\sigma$ is equivalent to $\{x_0, x_2, ..., x_n\}_n^\sigma \vee \{x_1, x_2, ..., x_n\}_n^\sigma \vee \{x_0, x_1, x_2, ..., x_n\}_{n+1}^\sigma$, we can eliminate the disjunctions over variables as well. It follows that any f_i can be rewritten as a disjunction of formulas $\{w_1, \ldots, w_s\}_s^\sigma$, where each w_j is a variable, $\top$ or $\perp$.

If now we transform the resulting system of equations in an alternating modal automaton as outlined in Section 3.1.2, it is easily verified that the resulting automaton, which we know to be equivalent to ϕ, is a *nondeterministic* automaton.

Moreover, if ϕ is a formula in N_1 or N_2, then the corresponding nondeterministic automaton has index 1 or 2. □

3.3 Closure under Bisimulation Quantifiers

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

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

Here we prove that the same holds if we restrict to the first two levels N_1, N_2 of the μ-calculus hierarchy: N_1, N_2 are closed under existential bisimulation quantifiers. To show this, we first prove that bisimulation quantifiers applied to bisimulation invariant formulas behave like MSO-quantifiers on the class of ω-expanded trees. We first remark:

Lemma 3.10 *If the trees T, T' are ω-expanded and bisimilar, then they satisfy the same MSO-sentences.*

Proof.
This holds because on ω-expanded trees any MSO-sentence is equivalent to a μ-calculus sentence ([6]), and bisimilar structures satisfy the same μ-sentences. □

Lemma 3.11 *If T is an ω-expanded tree and ϕ is a μ or BQL-sentence, then*

$$\exists P \subseteq T \textit{ with } (T, P) \models \phi \Leftrightarrow \exists N \textit{ and } P \subseteq N \textit{ with } N \sim_{-P} T \textit{ and } (N, P) \models \phi.$$

In other words, the existential quantifier $\exists P$ behaves like the existential bisimulation quantifier $\widetilde{\exists} P$ on ω-expanded models.

Proof.
The implication from left to rightis trivial and does not require that T is ω-expanded (just choose $N = T$). Conversely, suppose that there exists an N with $N \sim_{-P} T$ and $P \subseteq N$ such that $(N, P) \models \phi$. If N^ω is the ω-expansion of N we still have: there exists a $P \subseteq N^\omega$ with $(N^\omega, P) \models \phi$. This can be expressed as a second order property (in the language $Prop \setminus \{P\}$) of the structure N^ω, which is bisimilar to T w.r.t. this language. Hence, from Lemma 3.11 it holds: there exists a $P \subseteq T$ with $(T, P) \models \phi$. $\square$

Theorem 3.12 *N_1 and N_2 are closed under existential bisimulation quantifiers on arbitrary models.*

Proof.
Fix $k \in \{1, 2\}$ and $\phi \in N_k$; by Lemma 3.6 we know that there exists a non deterministic automaton B with index k, which is equivalent to ϕ. By Lemma 3.4 there exists an alternating modal automaton C with the same parity index as B which recognizes an ω-expanded tree T iff there is a subset $P \subseteq T$ such that (T, P) is recognized by B. By Theorem 3.5, the automaton C is equivalent to an N_k-formula ψ, and using lemma 3.11 it is easy to see that ψ is equivalent to $\widetilde{\exists} P\phi$ on arbitrary models:

$$M \models \psi \Leftrightarrow M^\omega \models \psi \Leftrightarrow \exists P \subseteq M^\omega, (M^\omega, P) \models \phi \Leftrightarrow M^\omega \models \widetilde{\exists} P\phi \Leftrightarrow M \models \widetilde{\exists} P\phi,$$

where M^ω is the ω-expansion of M, the first equivalence holds because ψ is bisimulation invariant, the second because ψ is equivalent to "$\exists P\phi$" on ω-expanded models, the third by Lemma 3.11, and the last equivalence because $\widetilde{\exists} P\phi$ is bisimulation invariant. $\square$

Lemma 3.13 *There exists a translation ϕ° from BQL to μ such that:*

(a) ϕ is equivalent to ϕ°;

(b) for $k = 1, 2$, if $\phi \in \widetilde{\Sigma}_k$ ($\phi \in \widetilde{\Pi}_k$) then $\phi^\circ \in N_k$ ($\phi^\circ \in M_k$, respectively).

Proof.
To define the translation we consider BQL-formulas as built from propositional constants in $Prop$ and their negation $\{\neg P : P \in Prop\}$ using the connectives $\wedge, \vee$, the operators $\Diamond, \Box, \Diamond^\star, \Box^\star$, and the quantifiers $\widetilde{\exists}, \widetilde{\forall}$. Then we define:

$$P^\circ = P; \quad (\neg P)^\circ = \neg P; \quad (\phi_1 \vee \phi_2)^\circ = \phi_1^\circ \vee \phi_2^\circ; \quad (\phi_1 \wedge \phi_2)^\circ = \phi_1^\circ \wedge \phi_2^\circ;$$

$$(\Diamond^* \phi)^\circ = \mu X(\phi^\circ \vee \Diamond(X)); \qquad (\Box^* \phi)^\circ = \nu X(\phi^\circ \wedge \Box(X));$$

$$(\widetilde{\exists} P\phi)^\circ = \widetilde{\exists} P\phi^\circ; \qquad (\widetilde{\forall} P\phi)^\circ = \widetilde{\forall} P\phi^\circ.$$

Notice that the definition makes sense because we know from [2] that the μ-calculus is closed under bisimulation quantifiers: hence $\widetilde{\exists} P\phi^\circ, \widetilde{\forall} P\phi^\circ$ are μ-formulas. It is then easily verified that ϕ° is equivalent to ϕ. Moreover, we proved in 3.12 that N_1, N_2 are closed under existential bisimulation quantifiers, and from this it follows that the restriction of the translation $\circ$ to $N_1 \cup N_2$ maps N_i to $\widetilde{\Sigma}_i$, for $i = 1, 2$.

Theorem 3.14 *N_i and $\widetilde{\Sigma}_i$ have the same expressive power for $i \in \{1, 2\}$.*

Proof.
This follows from Lemma 3.1 and Lemma 3.13.

4 Collapse of the BQL-hierarchy

In the preceding section we proved that N_1 and N_2 are closed under bisimulation quantifiers, and this allowed us to prove that the first two levels $\widetilde{\Sigma}_1, \widetilde{\Sigma}_2$ of BQL coincide (w.r.t. their expressive power) with the first two levels N_1, N_2 of the μ-calculus hierarchy, respectively. In the following lemma we prove that this is not true after the second level, because the whole μ-calculus can be expressed in the third level of the BQL-hierarchy.

Lemma 4.1 *The whole μ-calculus can be expressed in $\widetilde{\Sigma}_3$.*

Proof.
By Lemma 3.6 for any μ-calculus formula ϕ we can construct a nondeterministic automaton A which is equivalent to ϕ. We now prove that there exists a $\widetilde{\Sigma}_3$-formula expressing the existence of a winning strategy for player I in any game of A over an expanded tree. Any such a formula is clearly equivalent to ϕ over the class of all trees.

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

1. $Q_0, \ldots, Q_n$ are disjoint, and the root of the tree belongs to Q_0;
2. if $s \in Q_i$ and $T(s) = \sigma$, then the sons of s equipped with the marking $m = \{(t, q_j) : t \in Q_j\}$ satisfy $\delta(q_i, \sigma)$;
3. if $s_0, s_1, \ldots$ is an infinite path starting from the root with $s_j \in Q_{i_j}$, then the greatest number appearing infinitely often in the sequence

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

   is even.

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

It follows that A accepts T iff $T \models \exists Q_1 \ldots \exists Q_n(\phi_1 \wedge \phi_2 \wedge \phi_3)$, where ϕ_1, ϕ_2, ϕ_3 are formulas expressing the above points. In the following we shall prove that ϕ_1, ϕ_2, ϕ_3 can be chosen to be in $\widetilde{\Sigma}_3$. Suppose this is done. Then, since over ω-expansions existential quantifiers over BQL-formulas are equivalent to bisimulation existential quantifiers (see Lemma 3.11), we obtain that

$$\widetilde{\exists} Q_1, \ldots, \widetilde{\exists} Q_n(\phi_1 \wedge \phi_2 \wedge \phi_3)$$

is a $\widetilde{\Sigma}_3$-formula which is equivalent to the starting ϕ.

To finish the proof, it only remains to show that ϕ_1, ϕ_1, ϕ_3 can be chosen to be in $\widetilde{\Sigma}_3$. This is obvious for ϕ_1. As for ϕ_2, following the proof of Lemma 3.6 it is easy to see that the transition formulas $\delta(q_i, \sigma)$ can be expressed from a node s with a modal formula $f_{q_i,\sigma}$ in the predicate Q_i's; thus we can define ϕ_2 as the PDL-formula

$$\Box^*(\bigwedge_{i,\sigma}(Q_i \wedge \hat{\sigma} \rightarrow f_{q_i,\sigma})),$$

which is easily seen to be in $\widetilde{\Sigma}_3$.

To express ϕ_3 with a formula in $\widetilde{\Sigma}_3$ we proceeed as follows. First of all, notice that we can express the fact that there exists an infinite chain starting from a node w, in which P appears infinitely often and Q appears in any point, by a μ-formula $\psi(P, Q)$ of the second level N_2:

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

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

$$\{\neg Q_i : \Omega(q_i) > \Omega(q_k)\}$$

for Q in the formula $\psi(P, Q)$; we obtain a formula $\psi_k \in N_2$ which is true in w iff from w starts a chain in which infinite points are in Q_k, and no point is in any of the Q_i, for $\Omega(q_i) > \Omega(q_k)$. Since we proved that N_2 has the same expressive power than $\widetilde{\Sigma}_2$, we may suppose that ψ_k is in $\widetilde{\Sigma}_2$. But then the $\widetilde{\Pi}_3$-formula

$$\psi = \bigvee_{\Omega(q_k) odd} \Diamond^*(\psi_k)$$

is true in the root of a tree iff there exist a k with $\Omega(q_k)$ odd and an infinite chain $s_0, s_1, \ldots$ starting from the root in which the greatest number appearing infinitely often in the sequence $\Omega(q_{i_0}), \Omega(q_{i_1}), \ldots$, where $s_j \in Q_{i_j}$ is $\Omega(q_k)$. Finally, notice that point (3) above is expressed by $\neg\psi$, which is in $\widetilde{\Sigma}_3$.

□

Corollary 4.2 *The BQL-hierarchy collapses at level three.*

Proof.
From [2] we know that BQL has the same expressive power than the μ-calculus, and from Lemma 4.1 we know that the μ-calculus is (as expressive as) $\widetilde{\Sigma}_3$.

Corollary 4.3 *The level N_k of the μ-calculus hierarchy is not closed under bisimulation quantifiers for $k > 2$.*

Proof.
Otherwise, following the proof of Lemma 3.12, we could define a translation from $\widetilde{\Sigma}_k$ to N_k, N_k would have the same expressive power of the whole μ-calculus, and the μ-calculus hierarchy would collapse, which is not the case.

5 Conclusion and Further Works

In this paper we studied hierarchy problems of the logic BQL. Much more has to be done regarding this logic, the first problem being to find a sound and complete calculus for it. More generally, starting from a logic L which is invariant under bisimulation, what can be said about the closure $\tilde{L}$ of L under bisimulation quantifiers? Is $\tilde{L}$ decidable if L is so? If C is a sound and complete calculus for L, can we find an extension $\tilde{C}$ which is sound and complete for $\tilde{L}$? Is this extension *uniform*, that is, not depending on the particular C? The reason for looking to such an extension is that, as we already said in the introduction, any logic $\tilde{L}$ will automatically enjoy uniform interpolation, a desirable property in the applications.

References

[1] A. Arnold and D. Niwinski. *Rudiments of μ–calculus,* North–Holland, Studies in Logic and the Foundations of Mathematics, 2001.

[2] G. D'Agostino and M.Hollenberg Logical questions concerning the μ-calculus. *Journal of Symbolic Logic*, Vol. 65, N.1, pages 310–332, March 2000.

[3] S. Ghilardi and M. Zawadowski, A sheaf representation and duality for finitely presented Heyting algebras *Journal of Symbolic Logic* 60:911-939,1995.

[4] D. Janin and G. Lenzi. Relating levels of the μ-calculus hierarchy and levels of the monadic hierarchy. *Proceedings of the conference LICS 2001*, IEEE Computer Society, 347–356.

[5] D. Janin and I. Walukiewicz. Automata for the modal μ-Calculus and related results In *MFCS'95*, volume 969 of LNCS, 552–562, 1995.

[6] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic. *Proceedings of CONCUR '96*, volume 1119 of LNCS, 263–277, 1996.

[7] M.Hollenberg *Logic and Bisimulation.* PhD thesis, University of Utrecht, Vol XXIV, Zeno Insitute of Philosophy, 1998.

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

[9] A. Visser. Bisimulations, model descriptions and propositional quantifiers. *Logic Group Preprint Series 161*, Dept. of Philosophy, Utrecht University, 1996.

[10] I. Walukiewicz. Monadic Second Order Logic on tree-like structures. In *Symp. on Theor. Aspects of Computer Science,* LNCS 1046, 1996.