GUARDED TRANSFORMATION FOR THE MODAL μ-CALCULUS

FLORIAN BRUSE, OLIVER FRIEDMANN, AND MARTIN LANGE

Abstract. Guarded normal form requires occurrences of fixpoint variables in a μ-calculus-formula to occur under the scope of a modal operator. The literature contains guarded transformations that effectively bring a μ-calculus-formula into guarded normal form. We show that the known guarded transformations can cause an exponential blowup in formula size, contrary to existing claims of polynomial behaviour. We also show that any polynomial guarded transformation for μ-calculus-formulas in the more relaxed vectorial form gives rise to a polynomial solution algorithm for parity games, the existence of which is an open problem.

1. Introduction

The modal μ-calculus $\mathcal{L}_\mu$, as introduced by Kozen [15], is a fundamental modal fixpoint logic which subsumes many other temporal [6, 5] and dynamic logics [15, 8]. The paper at hand is concerned with *guarded normal form* for the μ-calculus, or simply *guarded form*. A formula is guarded if every occurrence of any fixpoint variable is under the scope of a modal operator inside its defining fixpoint formula. For instance, the formula $\nu Y.\Diamond\mu X.p \vee \Diamond X$ from above is guarded, whereas $\nu Y.\mu X.(p \wedge Y) \vee \Diamond X$ is not.

Intuitively, guarded form ensures that in the evaluation of a formula in a transition system by fixpoint iteration, one proceeds along at least one transition between two iterations of the same variable. Guarded form is also very useful in procedures that check for satisfiability or validity of a set of formulas and handle fixpoint formulas by unfolding: Guardedness synchronises the unfolding of all formulas in a set. Many constructions require formulas to be explicitly normalised in guarded form or assume that w.l.o.g., formulas can be brought into guarded form with polynomial overhead [14, 12, 16, 20, 18, 13]. Others can cope with unguarded formulas, but then their constructions require the solving of non-trivial decision problems [7]. Only few deal explicitly with unguarded formulas [9], but they require special tricks in order to handle unguardedness.

It has been known for quite a while that every $\mathcal{L}_\mu$-formula can effectively be transformed into an equivalent guarded formula. The first guarded transformation routine – described by Banieqbal and Barringer, as well as Walukiewicz – explicitly rewrites Boolean subformulas into disjunctive or conjunctive normal form [2, 20]. Clearly, this increases the sizes of formulas exponentially in the worst case. Such a blowup may not be considered harmful for results concerning the expressive power of $\mathcal{L}_\mu$, but it clearly makes a difference for complexity-theoretic results. Kupferman et al. [16, Thm. 2.1] notice that the transformation into Boolean normal forms

Date: May 3, 2013.

is unnecessary and present an optimised variant of this guarded transformation procedure. They claim that it only involves a linear blowup, but this is wrong. The problem lies in the very last statement of their proof: "... by definition $\varphi'(\lambda y.\varphi'(y)) \in \mathrm{Cl}(\lambda y.\varphi'(y))$ ..." While this is true, it is not true that if $\lambda y.\varphi'(y)$ is a subformula of φ – and hence in the Fischer-Ladner closure of φ – then also $\varphi[\varphi'(\lambda y.\varphi'(y))/\lambda y.\varphi'(y)]$ is in the closure of φ. Unfolding from the inside out – a principle that forms the core of the guarded transformation – produces exactly this kind of situation. Repeated application of this principle will, in the worst case, result in an exponential growth in the number of distinct subformulas.

Later, another guarded transformation procedure was given by Mateescu [18, Sect. 2.2]. It turns out that it is practically the same algorithm as that given by Kupferman et al. earlier. However, Mateescu estimates it to create an exponential blowup in formula size when used naïvely. On the other hand, he claims that "... each fixpoint subformula ... will be duplicated only once ... and reused ... leading to ... $|t(\varphi)| \leq |\varphi|^2$." Again, this is a false observation because the replacement of certain variables by constants can duplicate subformulas. Thus, the formula sharing trick that Mateescu proposes as a solution, which is just a way of saying that the size measured in terms of number of subformulas shall only be quadratic, does not work and the blow-up is not just quadratic.

Neumann and Seidl study guarded transformation in the more general context of hierarchical equation systems [19] with monotone operators. The modal operators $\Box$ and $\Diamond$ are monotone, so hierarchical equation systems are a generalisation of the *vectorial form* that is sometimes used to put multiple nestings of least or greatest fixpoints of the same kind into one block [1]. The semantics of $\mathcal{L}_\mu$ with vectorial form is simply given via simultaneous fixpoint definitions rather than parametric ones. The Bekić Lemma shows that this does not gain additional expressive power [3] but it may gain exponential succinctness because the only known transformations of $\mathcal{L}_\mu$ with vectorial form into $\mathcal{L}_\mu$ without incur an exponential blow-up in formula size.

Neumann and Seidl even give a guarded transformation algorithm for equation systems that result from a $\mathcal{L}_\mu$-formula and claim that it is polynomial. This claim is correct, as far as we can tell. However, the resulting equation system does not have the nice structure that equation systems corresponding to $\mathcal{L}_\mu$-formulas have. Thus, their guarded transformation for $\mathcal{L}_\mu$-formulas is polynomial, yet it does not produce equivalent guarded $\mathcal{L}_\mu$-formulas but only equivalent guarded equation systems. Translating these back into a guarded $\mathcal{L}_\mu$-formula incurs an exponential blowup, given current knowledge.

The structure of the paper is as follows. After introducing $\mathcal{L}_\mu$, vectorial form and guardedness formally in Sect. 2, we briefly describe and analyse the guarded transformation procedures by Kupferman et al. and Mateescu, respectively that of Neumann and Seidl in Sect. 3. We show that it can produce formulas of at least exponential size (measured as the number of different subformulas). In Sect. 4 we show that guarded transformation for $\mathcal{L}_\mu$-formulas in vectorial form is hard: we show that it is not easier than solving parity games. As mentioned above, we also show that unfolding vectorial form into a non-vectorial formula has the same lower complexity bound. This means that any polynomial algorithm for one of these problems would yield a polynomial algorithm for solving parity games, and this would settle a major and long-standing open problem. Finally, in Sect. 5 we

discuss the consequences of this work for previously acclaimed results about $\mathcal{L}_\mu$ that can be found in the literature and we briefly sketch the relation of our results to the problem of eliminating ε-transitions in alternating parity tree automata.

2. The Modal μ-Calculus

Transition Systems. A *labeled transition system* (LTS) over a set of *action names* Σ and a set of *atomic propositions* $\mathcal{P}$ is a tuple $\mathcal{T} = (S, \to, \ell)$ where S is a set of *states*, $\to \subseteq S \times \Sigma \times S$ defines a set of *transitions* between states, labeled with action names, and $\ell : S \to 2^{\mathcal{P}}$ labels each state with the set of atomic propositions that are true in this state.

Syntax. Let Σ and $\mathcal{P}$ be as above and let $\mathcal{V}$ be a set of variables. Formulas of the modal μ-calculus $\mathcal{L}_\mu$ in positive normal form are those that can be derived from φ in

$$\varphi \ ::= \ q \mid \overline{q} \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi,$$

where $X \in \mathcal{V}$, $q \in \mathcal{P}$, and $a \in \Sigma$.

The operators μ and ν act as *binders* for the variables in a formula. A *free occurrence* of a variable X is therefore one that does not occur under the scope of such a binder. A *closed* formula is one that does not have any free variables. We write σ for either μ or ν.

Let $\mathrm{Sub}(\varphi)$ denote the set of *subformulas* of φ. Define the *size* of a formula φ as the number of its distinct subformulas, i.e. $|\varphi| := |\mathrm{Sub}(\varphi)|$. We assume all $\mathcal{L}_\mu$-formulas to be *well-named* in the sense that each variable is bound at most once. Hence, for every $\mathcal{L}_\mu$-formula there is a partial function $\mathrm{fp}_\varphi : \mathcal{V} \to \mathrm{Sub}(\varphi)$ which maps a variable X that is bound in φ by some operator $\sigma X.\psi$ to its *defining fixpoint formula* ψ.

As usual, we use the abbreviations $\mathtt{tt} = q \vee \overline{q}$ and $\mathtt{ff} = q \wedge \overline{q}$ for an arbitrary q. Given a fixpoint binder σ, we write $\hat{\sigma} = \mathtt{ff}$ if $\sigma = \mu$ and $\hat{\sigma} = \mathtt{tt}$ if $\sigma = \nu$.

We write $\varphi[\psi/X]$ to denote the formula that results from φ by replacing every free occurrence of the variable X in it with the formula ψ.

The *modal depth* md is the maximal nesting depth of modal operators in a formula, formally defined as follows.

$$\begin{aligned} \mathrm{md}(q) \ = \ \mathrm{md}(\overline{q}) \ &= \ \mathrm{md}(X) \ := \ 0 \\ \mathrm{md}(\varphi \vee \psi) \ &= \ \mathrm{md}(\varphi \wedge \psi) \ := \ \max\{\mathrm{md}(\varphi), \mathrm{md}(\psi)\} \\ \mathrm{md}(\langle a \rangle \varphi) \ &= \ \mathrm{md}([a]\varphi) \ := \ 1 + \mathrm{md}(\varphi) \\ \mathrm{md}(\mu X.\varphi) \ &= \ \mathrm{md}(\nu X.\varphi) \ := \ \mathrm{md}(\varphi) \end{aligned}$$

An important fragment of $\mathcal{L}_\mu$ that we consider later is the *propositional μ-calculus* $\mathcal{B}_\mu$. It consists of all $\varphi \in \mathcal{L}_\mu$ such that $\mathrm{md}(\varphi) = 0$. Hence, $\mathcal{B}_\mu$-formulas do not contain any subformulas of the form $\langle a \rangle \psi$ or $[a]\psi$. A formula is called *purely propositional* if it belongs to $\mathcal{B}_\mu$ and does not contain any fixpoint operators.

Semantics. Formulas of $\mathcal{L}_\mu$ are interpreted in states of an LTS $\mathcal{T} = (S, \to, \ell)$. Let $\rho \colon \mathcal{V} \to 2^{\mathcal{S}}$ be an environment used to interpret free variables. We write $\rho[X \mapsto T]$ to denote the environment which maps X to T and behaves like ρ on all other arguments. The semantics of $\mathcal{L}_\mu$ is given as a function $[\![\cdot]\!]$ mapping a formula to

the set of states where it holds w.r.t. the environment.

$$\begin{aligned}
[\![q]\!]^{\mathcal{T}}_{\rho} &= \{s \in S \mid q \in \ell(s)\} \\
[\![\overline{q}]\!]^{\mathcal{T}}_{\rho} &= \{s \in S \mid q \notin \ell(s)\} \\
[\![X]\!]^{\mathcal{T}}_{\rho} &= \rho(X) \\
[\![\varphi \vee \psi]\!]^{\mathcal{T}}_{\rho} &= [\![\varphi]\!]^{\mathcal{T}}_{\rho} \cup [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\varphi \wedge \psi]\!]^{\mathcal{T}}_{\rho} &= [\![\varphi]\!]^{\mathcal{T}}_{\rho} \cap [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\langle a \rangle \varphi]\!]^{\mathcal{T}}_{\rho} &= \{s \in S \mid \exists t \in [\![\varphi]\!]^{\mathcal{T}}_{\rho} \text{ with } s \xrightarrow{a} t\} \\
[\![[a] \varphi]\!]^{\mathcal{T}}_{\rho} &= \{s \in S \mid \forall t \in S : \text{ if } s \xrightarrow{a} t \text{ then } t \in [\![\varphi]\!]^{\mathcal{T}}_{\rho}\} \\
[\![\mu X.\varphi]\!]^{\mathcal{T}}_{\rho} &= \bigcap \{T \subseteq S \mid [\![\varphi]\!]^{\mathcal{T}}_{\rho[X \mapsto T]} \subseteq T\} \\
[\![\nu X.\varphi]\!]^{\mathcal{T}}_{\rho} &= \bigcup \{T \subseteq S \mid T \subseteq [\![\varphi]\!]^{\mathcal{T}}_{\rho[X \mapsto T]}\}
\end{aligned}$$

Two formulas φ and ψ are equivalent, written $\varphi \equiv \psi$, iff for all LTS $\mathcal{T}$ and all environments ρ we have $[\![\varphi]\!]^{\mathcal{T}}_{\rho} = [\![\psi]\!]^{\mathcal{T}}_{\rho}$. We may also write $\mathcal{T}, s \models_{\rho} \varphi$ instead of $s \in [\![\varphi]\!]^{\mathcal{T}}_{\rho}$.

Vectorial Form. Sometimes it is convenient to define several fixpoints simultaneously. Vectorial form is an extension of the syntax of $\mathcal{L}_{\mu}$ which allows one to do so. Let $X_1, \ldots, X_m$ be variables and let $\psi_1, \ldots, \psi_m$ be formulas, possibly with free occurrences of the X_i. For both $\sigma \in \{\mu, \nu\}$,

$$\Phi = \sigma \left\{ \begin{array}{ll} X_1. & \psi_1 \\ & \vdots \\ X_m. & \psi_m \end{array} \right\}$$

is a formula in *m-vectorial form*, and the X_i are considered bound in Φ. We say that a formula is in *vectorial form* if it is in m-vectorial form for some m. Hence, formulas in 1-vectorial form are ordinary formulas as introduced above. The curly brackets are used to indicate that the m defining fixpoint equations are to be seen as a set; there is no implicit order among them with the exception of a variable marked as *entry variable*. By convention, the entry variable is the variable that occurs first, e.g. X_1 in the example above. We consider multiple instances of the same vectorial form, but with different entry variable, to be the same subformula.

Such formulas are used in order to abbreviate formulas with multiple nestings of fixpoint subformulas of the same kind. We give an inductive translation from formulas in vectorial form into ordinary formulas. Let

$$\Phi = \sigma \left\{ \begin{array}{ll} X_1. & \psi_1 \\ & \vdots \\ X_m. & \psi_m \end{array} \right\}$$

be a formula in m-vectorial form as above. Let $\mathcal{X}$ denote the set $\{X_1, \ldots, X_m\}$. By convention, X_1 is the entry variable. For each ψ_i and for each $\emptyset \neq \mathcal{Y} \subseteq \mathcal{X}$ define a formula $\psi_i^{\mathcal{Y}}$ inductively via $\psi_i^{\{i\}} = \sigma X_i.\psi_i$ and for $i \in \mathcal{Y}$ via $\psi_i^{\mathcal{Y}} = \sigma X_i.\psi_i[\psi_j^{\mathcal{Y} \setminus \{i\}} / X_j : j \in \mathcal{Y} \setminus \{i\}]$. The formula $\sigma X_1.\psi_1^{\mathcal{X} \setminus \{1\}}$ provides the intended meaning for Φ.

The size of this formula can be estimated as follows: The size of $\sigma X_i.\psi_i$ is $|\psi_i|+1$. The size of $\sigma X_i.\psi_i^{\mathcal{Y}}$ can be bounded from above by $|\varphi_i| + 1 + \Sigma_{j\in\mathcal{Y}\setminus\{i\}}|\varphi_j^{\mathcal{Y}\setminus\{i\}}|$. If $c = \max_i |\varphi_i|$ then $|\sigma X_1.\psi_1^{\mathcal{X}\setminus\{1\}}| \in \mathcal{O}(c \cdot m!)$. While this procedure has very bad complexity, to our knowledge, no better procedure is known.

Example 2.1. Consider the following 3-vectorial formula where $\Box\psi$ is used to abbreviate $\bigwedge_{a\in\Sigma}[a]\psi$.

$$\mu\left\{\begin{array}{ll} X. & \Box\mathtt{ff} \vee \langle a\rangle Y \vee Z \\ Y. & \langle b\rangle(Y \vee X) \\ Z. & \langle a\rangle X \vee \langle c\rangle Z \end{array}\right\}$$

It expresses "there is a maximal path labelled with a word from $(ab^+ + c^*a)^*$". It abbreviates the following formula in non-vectorial form.

$$\mu X.\Box\mathtt{ff} \vee \langle a\rangle(\mu Y.\langle b\rangle(Y \vee X)) \vee \mu Z.\langle a\rangle X \vee \langle c\rangle Z$$

Guardedness and Weak Guardedness. A formula φ is *guarded w.r.t. a variable* X if every occurrence of X that is bound by some $\sigma X.\psi$ is in the scope of a modal operator $\langle a\rangle$ or $[a]$ within ψ. A formula φ is *guarded* iff φ is guarded w.r.t. every bound variable.

We say that the variable X is *weakly guarded* in φ if X is guarded or if it occurs under the scope of another fixpoint quantifer in its defining fixpoint subformula $\sigma X.\psi$. Take for instance $\mu X.q \vee (\mu Y.(q \wedge X) \vee (\overline{q} \wedge Y) \vee \langle a\rangle Y)$. Then Y has both a guarded and an unguarded occurrence, whereas the only occurrence of X is not guarded but it is weakly guarded. Note that weak guardedness is indeed weaker than guardedness, hence, not being weakly guarded entails not being guarded.

For a formula in vectorial form, an unguarded cycle is a sequence of variables $X_1, \ldots, X_n$ such that X_{i+1} occurs unguarded on the right hand side of X_i for all $i < n$ and such that X_1 occurs unguarded on the right hand side of X_n. A formula in vectorial form is guarded if it does not contain any unguarded cycles. An occurrence of a variable X is weakly guarded or if it occurs on the right side of a variable $Y \neq X$. The reader is invited to check that a formula in vectorial form is guarded if and only if the associated non-vectorial formula is guarded.

Example 2.2. The formula

$$\mu X.\Box\mathtt{ff} \vee \langle a\rangle(\mu Y.\langle b\rangle(Y \vee X)) \vee \mu Z.\langle a\rangle X \vee \langle c\rangle Z$$

from Example 2.1 expresses "there is a maximal path labeled with a word from $(ab^+ + c^*a)^*$" and is guarded. However, consider the following formula which expresses the slightly different property "there is a maximal path labeled with a word from $(ab^+ + c^*)^*$".

$$\mu X.\Box\mathtt{ff} \vee \langle a\rangle(\mu Y.\langle b\rangle(Y \vee X)) \vee \mu Z.X \vee \langle c\rangle Z$$

It is not guarded; in particular, there is an occurrence of X—the latter one—which is not guarded in its defining fixpoint formula, which happens to be the entire formula in this case.

A *guarded transformation* for $\mathcal{L}_\mu$ is a function $\tau : \mathcal{L}_\mu \to \mathcal{L}_\mu$ such that $\tau(\varphi)$ is guarded and $\tau(\varphi) \equiv \varphi$ for every $\varphi \in \mathcal{L}_\mu$.

3. Guarded Transformation As Before Is Exponential

Guarded Transformation Without Vectorial Form. The guarded transformation procedures for non-vectorial formulas by Kupferman et al. and Mateescu rely on two principles. The first principle is the well-known fixpoint unfolding.

Proposition 3.1. *For every $\sigma X.\varphi \in \mathcal{L}_\mu$ we have $\sigma X.\varphi \equiv \varphi[\sigma X.\varphi/X]$.*

The second principle states how occurrences that are not weakly guarded can be eliminated.

Proposition 3.2 ([16, 18])**.** *Let $\sigma X.\varphi \in \mathcal{L}_\mu$ and let $\sigma X.\varphi'$ result from $\sigma X.\varphi$ by replacing with $\hat{\sigma}$ every occurrence of X that is not weakly guarded. Then $\sigma X.\varphi \equiv \sigma X.\varphi'$.*

These two principles can be combined to a simple guarded transformation procedure. Starting with the innermost fixpoint bindings, one replaces all occurrences of the corresponding variables that are not weakly guarded by $\mathtt{tt}$ or $\mathtt{ff}$ using Proposition 3.2. Note that for the innermost fixpoint subformulas, the concepts of being weakly guarded and being guarded coincide. Thus, the innermost fixpoint subformulas are guarded after this step.

For outer fixpoint formulas this only ensures that all remaining occurrences are weakly guarded. However, by the induction hypothesis, all inner ones are already guarded, and unfolding them using Prop. 3.1 puts all weakly guarded but unguarded occurrences of the outer variable under a $\langle a\rangle$- or $[a]$-modality. Hence, only occurrences that are either guarded or not weakly guarded survive, and the latter can be eliminated using Proposition 3.2 again.

Let τ_0 denote the guarded transformation which works as described above. Kupferman et al. claim that the worst-case blowup in formula size produced by τ_0 is linear, Mateescu claims that it is quadratic. We will show that it is indeed exponential. Consider the family of formulas

$$\Phi_n \;:=\; \mu X_1 \ldots \mu X_n.(X_1 \vee \cdots \vee X_n) \vee \langle a\rangle(X_1 \vee \cdots \vee X_n).$$

Theorem 3.3. *We have $|\Phi_n| = 3n + 1$ and $|\tau_0(\Phi_n)| = \Omega(2^n)$.*

Proof. The first claim about the linear growth of Φ_n is easily verified. We prove that $\tau_0(\Phi_n)$ contains a subformula of modal depth at least 2^{n-1}, which entails exponential size of $\tau_0(\Phi_n)$.

Let $\varphi = (X_1 \vee \cdots \vee X_n)$. Mateesecu's guarded transformation transforms a (strict) subformula of the form $\sigma X.\psi$, into $f_X(t'(\psi))[\sigma X.f_X(t'(\psi))/X]$, where t' is the guarded transformation for subformulas and f_X replaces unguarded occurrences of X by $\hat{\sigma}$. Moreover, $t'(\varphi \vee \langle a\rangle\varphi) = \varphi \vee \langle a\rangle\varphi$. For $2 \leq i \leq n$, define $\varphi_i = t'(\mu X_i \cdots X_n.(\varphi \vee \langle a\rangle\varphi)$. Then $\varphi_n = f_{X_n}(\varphi \vee \langle a\rangle\varphi)[\mu X_n.f_{X_n}(\varphi \vee \langle a\rangle\varphi)/X_n]$, and generally, $\varphi_i = f_{X_i}(\varphi_{i+1})[\mu X_i.f_{X_i}(\varphi_{i+1})/X_i]$. We show that φ_i contains φ at modal depth $2^{(n+1-i)}$. Clearly φ_n contains φ at modal depth $2 = 2^1$. Since $\varphi_i = f_{X_i}(\varphi_{i+1})[\mu X_i.f_{X_i}(\varphi_{i+1})/X_i]$, we have that, if φ_{i+1} contains φ at modal depth 2^{n-i}, then φ_i contains φ at double the modal depth, or $2 \cdot 2^{n-i} = 2^{n+1-i}$. Hence, $t'(\mu X_2 \cdots \mu X_n.(\varphi \vee \langle a\rangle\varphi)) = \varphi_2$ contains a formula of modal depth 2^{n-1}. Finally, Mateescu defines $\tau_0(\sigma X.\psi) = \sigma X.f_X(t'(\psi))$, whence $\tau_0(\Phi_n) = \mu X_1.f_X(\varphi_2)$, and the proof is finished. □

Guarded Transformation With Vectorial Form. Neumann and Seidl present guarded transformation in the context of *hierarchical equation systems* (HES) over distributive lattices with monotone operators [19]. Such equation systems can be seen as a generalisation of *vectorial form*, and the powerset lattice with operators associated to $\langle a\rangle$ and $[a]$ is distributive. Hence, the HES obtained from $\mathcal{L}_\mu$-formulas form a subclass of the class of all HES. Moreover, the notion of guardedness has a suitable generalisation for HES. Furthermore, the class of HES obtained from $\mathcal{L}_\mu$-formulas allows a guarded transformation that runs in polynomial time. Unfortunately, the resulting guarded HES are no longer in the class that corresponds directly to $\mathcal{L}_\mu$-formula, whence the procedure by Neumann and Seidl does not constitute a polynomial guarded transformation.

This has, roughly speaking, the following reason. Consider the variable dependency graph of a non-vectorial formula, defined as follows: The variables form the nodes, and there is an edge from variable X to variable Y if Y depends on$X$.[1] For $\mathcal{L}_\mu$-formulas, this graph is a tree with back edges, i.e. there is a set of edges that, if removed, leaves a tree. Moreover, the edges in this set go from a node to one of its predecessors in the tree. Forward edges correspond to occurrences of $\sigma Y.\psi_Y$ in the scope of $\sigma' X.\psi_X$, that is, without a third fixpoint binder in between. Back edges correspond to occurrences of X in ψ_Y such that $\sigma Y.\psi_Y$ is a subformula of $\sigma' X.\psi_X$. On the other hand, the variable dependency graph can be quite complicated already for $\mathcal{L}_\mu$-formulas in vectorial form: In a formula in m-vectorial form with variables $X_1, \ldots, X_m$, the dependency graph, restricted to these variables, can be the complete graph with m vertices. In general HES, this behaviour is not restricted to a single vector alone. The guarded transformation procedure of Neumann and Seidl destroys the nice structure of the dependency graph. An example is the following family of formulas:

$$\mu X_1. \ldots \mu X_n. X_1 \vee \cdots X_n \vee \langle a\rangle\Big(\bigvee_{j=1}^{m} \mu Y_j.\langle a\rangle\big(Y_j \vee \bigvee_{i=1}^{n} X_i\big)\Big)$$

The associated equation system looks roughly like this:

$$\begin{aligned} X_1 &= X_2 & Y_1 &= \langle a\rangle(Y_1 \vee X_1 \vee \cdots \vee X_n) \\ &\vdots & &\vdots \\ X_{n-1} &= X_n & Y_m &= \langle a\rangle(Y_m \vee X_1 \vee \cdots \vee X_n) \\ X_n &= X_1 \vee \cdots \vee X_n \vee \langle a\rangle(Y_1 \vee \cdots \vee Y_m) \end{aligned}$$

After the guarded transformation, the HES looks like this:

$$\begin{aligned} X_1 &= \langle a\rangle(Y_1 \vee \cdots \vee Y_m) & Y_1 &= \langle a\rangle(Y_1 \vee X_1 \vee \cdots \vee X_n) \\ &\vdots & &\vdots \\ X_n &= \langle a\rangle(Y_1 \vee \cdots \vee Y_m) & Y_m &= \langle a\rangle(Y_m \vee X_1 \vee \cdots \vee X_n) \end{aligned}$$

This HES has a variable dependency graph that is not a tree with back edges. To our knowledge, there is no known way to transform such a HES into a $\mathcal{L}_\mu$-formula without exponential blowup.

[1] NB: This is the converse of the way Neumann and Seidl define variable dependency.

4. Guarded Transformation Is Not That Easy

In this section we show that guarded transformation for HES is at least as hard—modulo polynomials—as parity game solving. The problem of whether or not the latter is possible in polynomial time has been open for a long while. We also show that unfolding a formula in vectorial form into an equivalent non-vectorial formula is at least as hard as parity game solving. The core of the proofs is a product construction similar to that in Kupferman et al. [16].

Theorem 4.1 (Product Construction)**.** *For every LTS $\mathcal{T}$ with state s_0 and every closed $\varphi \in \mathcal{L}_\mu$ there is an LTS $\mathcal{T}'$ with a state v_0 and vectorial $\varphi' \in \mathcal{B}_\mu$ such that*

- $\mathcal{T}, s_0 \models \varphi$ *iff* $\mathcal{T}', v_0 \models \varphi'$,
- $|\mathcal{T}| = \mathcal{O}(|\varphi| \cdot |\mathcal{T}|)$, *and*
- $|\varphi'| = \mathcal{O}(|\varphi| \cdot |\mathcal{T}|)^2$.

Proof. Let $\mathcal{T} = (S, \to, \ell)$ and let φ be defined over propositions $\mathcal{P}$ and variables $\mathcal{V}$. W.l.o.g. we assume that $S = \{1, \dots, m\}$ for some $m \in \mathbb{N}$. Define new sets of propositions $\mathcal{P}' := \mathcal{P} \times S$ and variables $\mathcal{V}' := \mathcal{V} \times S$. We write X_s and q_s instead of (X, s) and (q, s).

Let $\mathcal{T}' = (\{v_0\}, \emptyset, \ell')$ consist of a single state with the following labeling: $q_s \in \ell'(v_0)$ iff $q \in \ell(s)$.

Next we give an inductively defined transformation $\mathrm{tr}\colon S \times \mathcal{L}_\mu \to \mathcal{B}_\mu$ which turns an $\mathcal{L}_\mu$ formula over $\mathcal{V}$ and $\mathcal{P}$ into a $\mathcal{B}_\mu$ formula over $\mathcal{V}'$ and $\mathcal{P}'$.

$$
\begin{aligned}
\mathrm{tr}_s(q) &= q_s \\
\mathrm{tr}_s(\overline{q}) &= \overline{q_s} \\
\mathrm{tr}_s(X) &= X_s \\
\mathrm{tr}_s(\psi_1 \vee \psi_2) &= \mathrm{tr}_s(\psi_1) \vee \mathrm{tr}_s(\psi_2) \\
\mathrm{tr}_s(\psi_1 \wedge \psi_2) &= \mathrm{tr}_s(\psi_1) \wedge \mathrm{tr}_s(\psi_2) \\
\mathrm{tr}_s(\langle a \rangle \psi) &= \bigvee \{ \mathrm{tr}_t(\psi) \mid t \in S \text{ with } s \xrightarrow{a} t \} \\
\mathrm{tr}_s([a]\psi) &= \bigwedge \{ \mathrm{tr}_t(\psi) \mid t \in S \text{ with } s \xrightarrow{a} t \}
\end{aligned}
$$

$$
\mathrm{tr}_s \begin{pmatrix} X^1 & . & \psi^1 \\ & \vdots & \\ X^n & . & \psi^n \end{pmatrix} = \sigma \left\{ \begin{array}{lcl}
X^1_s & . & \mathrm{tr}_s(\psi^1) \\
X^1_1 & . & \mathrm{tr}_1(\psi^1) \\
 & \vdots & \\
X^1_{s-1} & . & \mathrm{tr}_{s-1}(\psi^1) \\
X^1_{s+1} & . & \mathrm{tr}_{s+1}(\psi^1) \\
 & \vdots & \\
X^1_m & . & \mathrm{tr}_m(\psi^1) \\
X^2_1 & . & \mathrm{tr}_1(\psi^2) \\
 & \vdots & \\
X^2_m & . & \mathrm{tr}_m(\psi^2) \\
 & \vdots & \\
X^n_1 & . & \mathrm{tr}_1(\psi^n) \\
 & \vdots & \\
X^n_m & . & \mathrm{tr}_m(\psi^n)
\end{array} \right\}
$$

Now let φ' be defined as $\mathrm{tr}_{s_0}(\varphi)$. It should be clear that φ' is indeed a formula of $\mathcal{B}_\mu$, and that its size is bounded by $(|\varphi| \cdot |\mathcal{T}|)^2$. Correctness of the reduction is not hard to prove either. We present an invariant which can easily be used to show by induction that $\mathcal{T}, s \models \varphi$ iff $v_0 \models \varphi'$ for any $s \in S$. The following stronger statement holds: let $\rho : \mathcal{V} \to 2^S$ be an environment for an $\mathcal{L}_\mu$ formula interpreted over $\mathcal{T}$. Define its associated $\mathcal{T}'$-environment $\rho' \colon \mathcal{V}' \times S \to 2^{\{v_0\}}$ as

$$\rho'(X_s) \;=\; \begin{cases} \{v_0\} & \text{if } s \in \rho(X), \\ \emptyset & \text{otherwise.} \end{cases}$$

Then, for all $s \in S$ and all ψ, we have $\mathcal{T}, s \models_\rho \psi$ iff $\mathcal{T}', v_0 \models_{\rho'} \mathrm{tr}_s(\psi)$. □

Theorem 4.2. *Guarded Transformation for $\mathcal{L}_\mu$-formulas or $\mathcal{B}_\mu$-formulas in vectorial form is at least as difficult as solving parity games.*

Proof. We show that any polynomial guarded transformation for $\mathcal{B}_\mu$-formulas in vectorial form yields a polynomial solution algorithm for parity games. The statement for $\mathcal{L}_\mu$-formulas follows from that because every guarded transformation for $\mathcal{L}_\mu$-formulas is also one for $\mathcal{B}_\mu$-formulas.

Assume that there is a guarded transformation procedure τ for vectorial $\mathcal{B}_\mu$-formulas. Given a parity game, we can treat it as an LTS with one accessibility relation and labellings for ownership and priority of states. Solving the parity game means deciding whether the first player wins from the initial vertex, and this is equivalent to model-checking Walukiewicz' formula [21] for the corresponding priority. This formula is of size linear in the number of priorities, hence it is polynomial in the size of the parity game. Via the product construction from Theorem 4.1, we obtain a vectorial $\mathcal{B}_\mu$-formula φ that is also of size polynomial in the size of the parity game, and a one-state transition system $\mathcal{T}'$ such that $\mathcal{T} \models \varphi$ if and only if the first player wins the parity game. Consider $\tau(\varphi)$. Because τ runs in polynomial time, the size of $\tau(\varphi)$ is still polynomial in the size of the parity game. Moreover, since the truth value of the $\mathcal{B}_\mu$-formula φ only depends on the state v_0, this must be true for $\tau(\varphi)$ as well. In effect, all modal operators introduced by τ are vacuous and can be replaced by $\mathtt{tt}$ in case of Boxes, and by $\mathtt{ff}$ in case of Diamonds. The resulting vectorial formula is again strictly boolean, but also guarded. Hence, it can not contain any occurrences of fixpoint variables at all, since any such occurrence would be unguarded. Therefore, all fixpoint quantifiers can be removed. The resulting formula is purely propositional and can be solved in polynomial time by a simple bottom-up algorithm. This yields the desired polynomial solution for the parity game. □

Theorem 4.3. *Transforming a vectorial $\mathcal{L}_\mu$-formula to a non-vectorial $\mathcal{L}_\mu$-formula is at least as difficult as solving parity games. This also holds for $\mathcal{B}_\mu$.*

Proof. We show that any polynomial procedure that transforms a vectorial $\mathcal{B}_\mu$-formula into a vectorial $\mathcal{B}_\mu$-formula yields a polynomial solution algorithm for parity games. As in the proof for Theorem 4.2, given a parity game we consider the vectorial $\mathcal{B}_\mu$-formula created as product of the LTS from of the parity game and the corresponding Walukiewicz formula. By assumption, we can transform it into a non-vectorial $\mathcal{B}_\mu$-formula in polynomial time. However, $\mathcal{B}_\mu$-formulas can be evaluated in linear time [19] whence, in effect, we can solve the parity game in polynomial time. □

In fact, this theorem even holds for HES in the sense of Neumann and Seidl, since any formula in vectorial form can be considered a HES over the powerset lattice of the underlying set of a transition system, and with operators $\langle a \rangle$ and $[a]$.

5. Conclusion

In Section 3 we saw that the known guarded transformations produce an exponential blowup in the worst case. In Section 4 we saw that a guarded transformation for vectorial formulas automatically entails a polynomial solution algorithm for parity games. We also saw that polynomial translation from vectorial formulas to non-vectorial formulas yields the same. The existence of a polynomial guarded transformation for *non-vectorial* formulas is still open, and it is possible that such a transformation exists without yielding a polynomial solution algorithm for parity games.

Consequences and Corrections. Several constructions and procedures that deal with the modal μ-calculus directly or indirectly use guarded transformation. Often they use the (possibly) false claim that guarded transformation can be done at a linear or quadratic blow-up. We examine consequences of the fact that according to current knowledge, guarded transformation is exponential.

As a first step, it is interesting to check whether any of the results from Kupferman/Vardi/Wolper's seminal paper on automata for branching-time temporal logics [16] crucially rely on a polynomial guarded transformation. Fortunately, this is not the case. The product automaton construction [16, Prop. 3.2] also works if the input automaton is not ε-free, and the resulting product automaton has no ε-transitions. This allows the subsequent Thm. 3.1 to be applied, which needs ε-free automata. Hence, all results of [16] that rely on a polynomial guarded transformation remain true.

In [11], the reliance on a polynomial guarded transformation causes problems. It is not immediately obvious that the complexity results for satisfiability of existential and universal $\mathcal{L}_\mu$ and alternation-free $\mathcal{L}_\mu$ (claimed to be NP-complete) as well as derived results should hold for unguarded formulas. A preceding transformation into guarded form will exhaust the complexity limitations, so further investigation on this work is necessary.

Mateescu claims that model checking for $\mathcal{L}_\mu$ on acyclic structures can be done in polynomial time. He observes that on acyclic structures, least and greatest fixpoints coincide. Thus, the alternation hierarchy collapses to its alternation-free fragment on such structures. It is known that this fragment can be model-checked in linear time [4]. However, least and greatest fixpoints only coincide for guarded formulas: Clearly $\mu X.X$ and $\nu X.X$ are not equivalent, not even on acyclic structures. The collapse result is still true but with current technology at hand, we need to assume an exponential blow-up in formula size. Thus, model checking guarded formulas on acyclic structures can be done in polynomial time, arbitrary formulas still require exponential time.

Automata and ε-transitions. A problem closely related to guarded transformations is the elimination of ε-transitions from alternating automata. It is standard practice to construct alternating parity tree automata from guarded $\mathcal{L}_\mu$-formulas, or weak alternating tree automata from guarded alternation-free formulas [7, 22, 10]. The resulting automata are of size linear in the size of the formula. The situation

is different for unguarded (alternation-free) $\mathcal{L}_\mu$-formulas because of the absence of a polynomial guarded transformation. Currently, we need to assume a blow-up that is exponential in the size of the formula when translating arbitrary, and therefore possibly unguarded, $\mathcal{L}_\mu$-formulas into alternating parity tree automata. If ε-transitions are allowed in alternating parity tree automata then it is possible to translate arbitrary $\mathcal{L}_\mu$-formulas into such automata at a linear blow-up only; the known constructions can be modified accordingly.

From Theorem 4.1 follows also that a polynomial procedure that eliminates ε-transitions in parity tree automata yields a polynomial algorithm for parity games: It is not difficult to encode a $\mathcal{B}_\mu$-formula in vectorial form into an automaton that only has ε-transitions. Eliminating these means solving the formula, a problem we know to be at least as hard as solving parity games by the theorem. The existing procedures to eliminate ε-transitions from parity tree automata have an exponential blowup. An example is a procedure of Wilke's [23] that also incurs a quadratic blowup in the number of states.[2] Finally, translating parity tree automata back into $\mathcal{L}_\mu$-formulas must be considered (super)exponential in the number of states as well, since any polynomial translation could be used to unravel formulas in vectorial form into non-vectorial formulas.

The above means that the size of an alternating automaton cannot be measured in terms of number of states. Instead, such a notion of size has to include the size of the transition relation, which can easily be exponentially larger. An example of the confusion that the wrong measure can cause is Kupferman/Vardi's work on alternating automata. They show that nonemptiness of weak alternating automata can be checked in linear time when size is measured including the transition relation [16]. This result is then used in a context where the size is measured in the number of states [17]. We do believe that this claim, and subsequent results, are to be questioned.

Further Research. The existence of a polynomial guarded transformation is still open, both for vectorial as well as non-vectorial formulas. It is also not known whether the existence of a polynomial guarded transformation procedure for non-vectorial formulas would entail parity games being solvable in polynomial time. Moreover, it is not known whether it is possible to unfold vectorial formulas into non-vectorial formulas with only polynomial blowup. Since the last two problems do entail a polynomial algorithm for parity games, we believe that, in order to find a polynomial algorithm for parity games, focusing on games directly might be more fruitful.

Finally, the converse questions are also open: Does a polynomial solution algorithm for parity games entail a polynomial guarded transformation procedure? Does it entail a polynomial procedure to unravel vectorial formulas into non-vectorial formulas? Of course, the ability to solve parity games in polynomial time allows polynomial model-checking for vectorial formulas, but the problems of guarded transformation, respectively of unravelling vectorial formulas, might be of independent interest.

[2]The paper does not state the quadratic blowup. Closer inspection shows that the result is flawed, but can be repaired with quadratic blowup.

References

[1] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

[2] B. Banieqbal and H. Barringer. Temporal logic with fixed points. In *Proc. Coll. on Temporal Logic in Specification*, volume 398 of *LNCS*, pages 62–73. Springer, 1989.

[3] H. Bekić. *Programming Languages and Their Definition, Selected Papers*, volume 177 of *LNCS*. Springer, 1984.

[4] Rance Cleaveland and Bernhard Steffen. A linear-time model-checking algorithm for the alternation-free modal mu-calculus. *Formal Methods in System Design*, 2(2):121–147, 1993.

[5] M. Dam. CTL* and ECTL* as fragments of the modal μ-calculus. *TCS*, 126(1):77–96, 1994.

[6] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 16, pages 996–1072. Elsevier and MIT Press, New York, USA, 1990.

[7] E. A. Emerson and C. S. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd Symp. on Foundations of Computer Science*, pages 368–377, San Juan, Puerto Rico, 1991. IEEE.

[8] E. A. Emerson and C. L. Lei. Efficient model checking in fragments of the propositional μ–calculus. In *Symposion on Logic in Computer Science*, pages 267–278, Washington, D.C., USA, 1986. IEEE.

[9] O. Friedmann and M. Lange. The modal μ-calculus caught off guard. In *20th Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'11*, volume 6793 of *LNCS*, pages 149–163. Springer, 2011.

[10] J. Gutierrez, F. Klaedtke, and M. Lange. The μ-calculus alternation hierarchy collapses over structures with restricted connectivity. In *Proc. 3rd Int. Symp. on Games, Automata, Logics and Formal Verification, GandALF'12*, volume 96 of *Elect. Proc. in Theor. Comp. Sc.*, pages 113–126, 2012.

[11] Thomas A. Henzinger, Orna Kupferman, and Rupak Majumdar. On the universal and existential fragments of the *mu*-calculus. *Theor. Comput. Sci.*, 354(2):173–186, 2006.

[12] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus with respect to monadic second order logic. In *Proc. 7th Conf. on Concurrency Theory, CONCUR'96*, volume 1119 of *LNCS*, pages 263–277. Springer, 1996.

[13] N. Jungteerapanich. A tableau system for the modal μ-calculus. In *Proc. 18th Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'09*, volume 5607 of *LNCS*, pages 220–234. Springer, 2009.

[14] R. Kaivola. Axiomatising linear time mu-calculus. In *Proc. 6th Int. Conf. on Concurrency Theory*, volume 962 of *LNCS*, pages 423–437. Springer, 1995.

[15] D. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.

[16] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[17] Orna Kupferman and Moshe Y. Vardi. Weak alternating automata and tree automata emptiness. In *STOC*, pages 224–233, 1998.

[18] R. Mateescu. Local model-checking of modal mu-calculus on acyclic labeled transition systems. In *Proc. 8th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'02*, volume 2280 of *LNCS*, pages 281–295. Springer, 2002.

[19] H. Seidl and A. Neumann. On guarding nested fixpoints. In *Proc. 8th Ann. Conf. on Computer Science Logic, CSL'99*, volume 1683 of *LNCS*, pages 484–498. Springer, 1999.

[20] I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Inf. and Comput.*, 157(1–2):142–182, 2000.

[21] Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theoret. Comput. Sci.*, 275(1-2):311–346, 2002.

[22] T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Belgian Math. Soc.*, 8(2):359–391, 2001.

[23] Thomas Wilke. Ctl$^+$ is exponentially more succinct than ctl. In *FSTTCS*, pages 110–121, 1999.

Florian Bruse, School of Elect. Eng. and Comp. Sc., University of Kassel, Germany
E-mail address: florian.bruse@uni-kassel.de

Oliver Friedmann, Dept. of Comp. Sc., University of Munich, Germany
E-mail address: oliver.friedmann@ifi.lmu.de

Martin Lange, School of Elect. Eng. and Comp. Sc., University of Kassel, Germany
E-mail address: martin.lange@uni-kassel.de