Size measures and alphabetic equivalence in the μ-calculus

Clemens Kupke
clemens.kupke@strath.ac.uk
University of Strathclyde
Glasgow, Scotland

Johannes Marti
johannes.marti@gmail.com
ILLC, University of Amsterdam
Amsterdam, The Netherlands

Yde Venema
y.venema@uva.nl
ILLC, University of Amsterdam
Amsterdam, The Netherlands

Abstract

Algorithms for solving computational problems related to the modal μ-calculus generally do not take the formulas themselves as input, but operate on some kind of representation of formulas. This representation is usually based on a graph structure that one may associate with a μ-calculus formula. Recent work by Kupke, Marti & Venema showed that the operation of *renaming* bound variables may incur an exponential blow-up of the size of such a graph representation. Their example revealed the undesirable situation that standard constructions, on which algorithms for model checking and satisfiability depend, are sensitive to the specific choice of bound variables used in a formula.

Our work discusses how the notion of alphabetic equivalence interacts with the construction of graph representations of μ-calculus formulas, and with the induced size measures of formulas. We introduce the condition of α-*invariance* on such constructions, requiring that alphabetically equivalent formulas are given the same (or isomorphic) graph representations.

Our main results are the following. First we show that if two μ-calculus formulas are α-equivalent, then their respective Fischer-Ladner closures have the same cardinality, up to α-equivalence. We then continue with the definition of an α-invariant construction which represents an arbitrary μ-calculus formula by a graph that has exactly the size of the quotient of the closure of the formula, up to α-equivalence. This definition, which is itself based on a renaming of variables, solves the above-mentioned problem discovered by Kupke et al.

CCS Concepts • **Theory of computation → Modal and temporal logics**; *Logic and verification*;

LICS '22, August 2–5, 2022, Haifa, Israel

ACM ISBN 978-1-4503-9351-5/22/08…$15.00
https://doi.org/10.1145/3531130.3533339

Keywords modal mu-calculus, complexity, alphabetic equivalence, model checking

ACM Reference Format:
Clemens Kupke, Johannes Marti, and Yde Venema. 2022. Size measures and alphabetic equivalence in the μ-calculus. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Haifa, Israel*. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3531130.3533339

1 Introduction

1.1 The modal μ-calculus

The modal μ-calculus [2, 3, 8, 11] is an extension of propositional modal logic by means of least- and greatest fixpoint operators, which enable the expression of recursive statements. Introduced by Kozen [14] in its current form, it has emerged in theoretical computer science as one of the key logical formalism used for specifying properties of *ongoing* processes [20]. Seen from a logical perspective, the formalism inherits many pleasant metalogical properties from basic modal logic, including uniform interpolation and other interesting model-theoretic properties [6, 10, 13], a natural complete axiomatisation [14, 21] and a complete cut-free proof system [1].

In line with the importance of the μ-calculus as a specification language, various computational aspects of the formalism have been investigated. The two problems at the center of these investigations concern *satisfiability* (given a μ-calculus formula ξ, is it satisfiable in some transition system?) and *model checking* (given a transition system $\mathbb{S}$, a state s in $\mathbb{S}$ and a μ-calculus formula ξ, is ξ true at s in $\mathbb{S}$?) The satisfiability problem was rather quickly shown to be decidable [15], while some years later Emerson & Jutla [9] gave an exponential time algorithm for satisfiability checking. Determining the complexity of the model checking problem, however, has turned out to be challenging. There is an obvious algorithm that runs in time $(k \cdot n)^d$, where k, n and d are respectively the size of the transition system, the size of the formula, and the alternation depth of the formula, i.e., the maximum number of alternating least and greatest fixpoint operators in the formula. While fairly recently a quasi-polynomial algorithm was found by Calude et alii [5], it is a long standing open question whether an algorithm exists that is entirely polynomial in the size of the formula.

1.2 Graph representations of μ-calculus formulas

Generally, the algorithms that are used to solve problems related to the modal μ-calculus do not take the formulas themselves as input, but operate on some kind of *representation* of formulas. As we will briefly discuss now in various contexts, this representation is usually based on a *graph* structure that one may associate with a μ-calculus formula, or can be viewed as such.

Parity Games The model checking problem for the μ-calculus directly corresponds to the problem of determining the winner of an (initialised) parity game. In fact, most work on the complexity of the model checking problem is done in this setting — this applies for instance to the just mentioned quasi-polynomial complexity results. Parity games are infinite two player games that are played over a graph, where the vertices are the game positions and the edges are the admissible moves relating these positions. When considering the model checking problem of a formula ξ in a transition system $\mathbb{S}$ as a parity game, the underlying graph of the game is defined as some kind of product of a graph representation of ξ with the graph of the transition system $\mathbb{S}$. When studying the model checking problem via its translation to parity games one naturally thinks of formulas as graphs.

Automata Many of the key theoretical results about the μ-calculus are proved by automata-theoretic methods. This includes for example the aforementioned exponential time satisfiability checking algorithm [9] by Emerson & Jutla, but also the expressive completeness theorem by Janin & Walukiewicz showing that the μ-calculus is the bisimulation invariant fragment of monadic second order logic [13], or the uniform interpolation result by D'Agostino & Hollenberg [6]. Concretely, the automata that one mostly associates with the modal μ-calculus are the μ-automata of Janin & Walukiewicz [12] and the alternating tree automata of Wilke [22]. Underlying the automata-theoretical approach is a construction that turns a μ-calculus formula ξ into an automaton $\mathbb{A}_\xi$ which accepts precisely those pointed transition systems where ξ is true. As argued in [16], it is in fact quite natural to view the transition structure of an automaton as a graph, and so one may indeed think of $\mathbb{A}_\xi$ as a graph-based representation of the formula ξ.

Equation Systems Formulas of the modal μ-calculus can also be represented by *systems of equations* [2, 8, 19]. In fact, as an alternative to the approach using parity games, the model checking problem can be represented as a so-called *boolean equation system* [17], which arises as some kind of interweaving of an equation system representing the formula with the model it is evaluated on. As with alternating tree automata, it is not too difficult to see the equation systems as being graph-based — here one may simply consider the union of the subformula graphs of the formulas appearing in one of the equations of the system.

Summarising, graph representations of formulas are of central importance in the theory of the modal μ-calculus. For concreteness, in this paper we will work with the *parity formulas* of [16] as uniform, generic graph-based representations of μ-calculus formulas. We will recall the definition of parity formulas in section 2.

Before we continue our discussion, let us note here that there are at least three natural ways to associate a graph with a μ-calculus formula ξ: its *syntax tree*, its *subformula dag*, which is based on the collection $Sfor(\xi)$ of subformulas of ξ, and its *closure graph*, which takes as its carrier its *(Fischer-Ladner) closure*, the set $Clos(\xi)$. Of these three, the subformula dag and the closure graph feature most prominently in algorithms and constructions.

1.3 The size of formula representations

Given the importance of graph representations of formulas in the theory of the μ-calculus, it is somewhat surprising that, while the literature is crystal clear on the algorithms that operate on these representations, the relation between a formula and its concrete representation is far less understood.

Bruse, Friedmann & Lange [4], who studied the complexity of a certain operation on μ-calculus formulas called guarded transformation, displayed a sequence of formulas of which the number of subformulas grows exponentially, whereas the size of the closure of the formulas grows only quadratically. While the closure size of a formula was known to never exceed the number of its subformulas [14], these size measures were generally assumed to be roughly the same. The observation in [4] revealed that in fact, the closure graph of a formula can be exponentially more succinct than its subformula dag. Consequently, for optimal complexity results on the μ-calculus it is generally advisable to work with closure graphs, and accordingly we will focus on this approach here.

Kupke, Marti & Venema [16] discussed the commonly made assumption that μ-calculus formulas may assumed to be clean.[1] This assumption is generally considered to be harmless, because formulas can be "cleaned up" by simply renaming bound variables. The authors, however, provided an example where such a renaming incurs an exponential blow-up of the size of its closure graph. This revealed that standard constructions for the μ-calculus, on which algorithms for model checking and satisfiability depend, are sensitive to the specific choice of bound variables that are used in a formula.

The gaps in our knowledge that were pointed out in these publications, cause problems in formulating and proving optimal (or even correct) complexity results for the modal μ-calculus. As a continuation of the work in [4] and [16], to remedy these shortcomings, our aim here is to further clarify

[1] A μ-calculus formula is *clean* (or *well-named*), if the sets of its free and bound variables are disjoint, and with every bound variable one may associate a unique subformula where this variable is bound.

the impact of *variable binding*, and more specifically, *alphabetic equivalence* on the graph representations of formulas.

1.4 Variable binding and alphabetic equivalence

A key feature of the syntax of the modal μ-calculus is that it involves *variable binding*. Every fixpoint operator binds the recursion variable in the subformula it governs. As a consequence, when working with formulas of the μ-calculus directly, one has to keep track of bound and free variables, which involves some nontrivial bookkeeping. Perhaps the appeal of game and automata theoretic approaches to the theory of the μ-calculus can be partially explained by the fact that graph representations provide an elegant variable-free alternative to standard formulas.

Here we will focus on the role of alphabetic equivalence, or briefly: α-equivalence, in the construction of graph representations of formulas. Roughly, two formulas are *α-equivalent* if they can be obtained from one another by suitable renamings of bound variables – a precise definition will be given further on. Generally, logicians tend to identify α-equivalent formulas, or at the very least, they consider the differences between α-equivalent formulas to be irrelevant. We certainly do not want to argue against this principle; on the contrary, our point is that it should be adhered to more consistently. In fact, to the best of our knowledge, there is no construction of a graph representation of a μ-calculus formula in which the principle has been taken into full account.

In particular, *none* of the currently available constructions that represent a formula ξ on the basis of its closure graph identify α-equivalent formulas in the set $Clos(\xi)$, and the same observation applies to the algorithms that work with the subformula dag. This is particularly odd since, as discussed above, such constructions generally feature a (usually implicit) preprocessing step that replaces the input formula with a clean alphabetic variant. In other words: these constructions do follow the principle of α-invariance on the side of the input, but fail to take it into account on the output side.

As discussed already, Kupke, Marti & Venema [16] were the first to point out the effects of renaming bound variables on the size of graph representations. Their main contribution is a construction that associates with an *arbitrary* (i.e., not necessarily clean) μ-calculus formula ξ a graph representation that is succinct in being based on the closure graph of ξ, while at the same time preserving the alternation depth of ξ. This construction, however, has the disadvantage that two distinct but α-equivalent formulas may receive different representations, possibly of exponentially differing sizes.

1.5 A succinct α-invariant representation

Our concrete goal here is to come up with a graph representation of μ-calculus formulas that is *α-invariant* in the sense that α-equivalent formulas obtain the same representation, and α-invariant formulas are identified throughout the construction. In contrast to the construction via *clean* alphabetic variants, which may result in an unnecessary exponential size blow-up, our construction will be much more succinct.

To formulate this more precisely we need some technical detail. Assume that, on the basis of the observations of Bruse et alii [4], in order to find an optimally succinct graph representation of a μ-calculus formula ξ, we take its Fischer-Ladner closure $Clos(\xi)$ as a starting point. Our aim will be to use the principle of α-invariance to *improve* on the construction by Kupke et alii [16], which would suggest to consider the α-equivalence classes of $Clos(\xi)$. Our first and promising observation is that while the respective closure sets of α-equivalent formulas need not have the same number of elements they do have the same number of α-cells:

$$\xi_0 =_\alpha \xi_1 \text{ implies } |Clos(\xi_0)/{=_\alpha}| = |Clos(\xi_1)/{=_\alpha}|. \quad (1)$$

Here and in the sequel we will write $=_\alpha$ to denote α-equivalence, and refer to the equivalence classes of this relation as *α-cells*. This raises the question whether perhaps we can base a graph representation of a formula ξ on the set $Clos(\xi)/{=_\alpha}$ consisting of the $=_\alpha$-cells in its closure, or perhaps on a related set of the same size.

The second and main contribution of this work is that we answer this question affirmatively.

Theorem 1.1. *There is a construction transforming an arbitrary μ-calculus formula ξ into an equivalent parity formula $\mathbb{P}_\xi$ such that*
1) $|\mathbb{P}_\xi| = |Clos(\xi)/{=_\alpha}|$;
2) the index of $\mathbb{P}_\xi$ is bounded by the alternation depth of ξ;
3) $\xi_0 =_\alpha \xi_1$ implies $\mathbb{P}_{\xi_0} = \mathbb{P}_{\xi_1}$.

Consequently, this approach induces the following *size measure* for μ-calculus formulas:

$$|\xi| := |Clos(\xi)/{=_\alpha}|.$$

This size measure is fully α-invariant in the sense that α-equivalent formulas obtain the same size, and in computing this size, α-equivalent formulas are only counted once. It is also optimal in the sense that it is the sharpest size measure (among the ones known from the literature) which can be used to correctly formulate the aforementioned complexity results for model checking and satisfiability.

The key idea underlying our proof of Theorem 1.1 is to use the observation (1) to our advantage.[2] That is, we will define an operation $\widehat{\cdot} : \mu\mathsf{ML} \to \mu\mathsf{ML}$ that is a renaming in the sense that

$$\xi =_\alpha \widehat{\xi} \quad (2)$$

for all ξ, and $\widehat{\cdot}$ picks a *fixed* member of the α-cell of its input formula:

$$\xi_0 =_\alpha \xi_1 \text{ implies } \widehat{\xi_0} = \widehat{\xi_1}. \quad (3)$$

[2] An alternative approach would be to use a different way to represent α-cells, for instance using de Bruijn indices. We will say more on this in Section 5.

The key feature of this renaming operation is that

$$\alpha\text{-equivalence is the identity relation on } Clos(\widehat{\xi}), \quad (4)$$

from which we immediately conclude that $|Clos(\widehat{\xi})/{=_\alpha}| = |Clos(\widehat{\xi})|$. Observe then that it follows from (1), (2) and (4) that

$$|Clos(\xi)/{=_\alpha}| = |Clos(\widehat{\xi})|.$$

In other words, $\widehat{\cdot}$ is a renaming operation that picks, for any μ-calculus formula ξ, a formula of *minimal* closure size among the alphabetic variants of ξ. Furthermore, we may think of the formulas in the closure of $\widehat{\xi}$ as representing the $=_\alpha$-cells in $Clos(\xi)$.

Given the semantic equivalence of the formulas ξ and $\widehat{\xi}$, this indicates that we may obtain a truly succinct graph representation of μ-calculus formulas as follows. We already mentioned that the main contribution of Kupke et alii [16] is a construction that associates with every μ-calculus formula ψ a succinct parity formula $\mathbb{G}_\psi$ that is based on the set $Clos(\psi)$ and has an index bounded by the alternation depth of ψ. We may now improve on this by taking, as an even more succinct graph representation of a μ-calculus formula ξ, the parity formula we obtain from applying the construction of [16] to the renaming $\widehat{\xi}$ of ξ:

$$\mathbb{P}_\xi := \mathbb{G}_{\widehat{\xi}}.$$

It is then easy to see that this definition meets the requirements of Theorem 1.1.

Related version Some of the more technical proofs can be found in the technical report [?].

2 Preliminaries

In this section we recall the syntax and semantics of the modal μ-calculus; for more information we refer to [2, 3, 8, 11]. We also briefly discuss the graph representation of its formulas as *parity formulas.*

2.1 Syntax of the μ-calculus

We will assume that μ-calculus formulas are in negation normal form; that is, the language μML of (modal) μ-calculus formulas is given by the following grammar:

$$\mu\mathsf{ML} \ni \varphi \quad ::= \quad p \mid \overline{p} \mid \bot \mid \top \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \Diamond\varphi \mid \Box\varphi \mid \mu x\, \varphi \mid \nu x\, \varphi,$$

where p, x are variables, and the formation of the formulas $\mu x\, \varphi$ and $\nu x\, \varphi$ is subject to the constraint that φ is *positive* in x, i.e., there are no occurrences of $\overline{x}$ in φ. Elements of μML will be called *μ-calculus formulas* or *standard formulas.* We define $\mathtt{Lit}(\mathsf{Q}) := \{p, \overline{p} \mid p \in \mathsf{Q}\}$ as the set of *literals* over Q, and $\mathtt{At}(\mathsf{Q}) := \{\bot, \top\} \cup \mathtt{Lit}(\mathsf{Q})$ as the set of *atomic formulas* over Q. Formulas of the form $\mu x.\varphi$ or $\nu x.\varphi$ will be called *fixpoint formulas.* We will associate μ and ν with the odd and even numbers, respectively, and use η, λ as metavariables for these two fixpoint binders. For $\eta \in \{\mu, \nu\}$ define $\overline{\eta}$ by putting $\overline{\mu} := \nu$ and $\overline{\nu} := \mu$. The notion of *subformula* is defined as usual; we write $\varphi \trianglelefteq \psi$ if φ is a subformula of ψ, and define $Sfor(\psi)$ as the set of subformulas of ψ.

We use standard terminology related to the binding of variables. We write $BV(\xi)$ and $FV(\xi)$ for, respectively, the set of *bound* and *free variables* of a formula ξ. We fix a set Q of proposition letters and let $\mu\mathsf{ML}(\mathsf{Q})$ denote the set of formulas ξ with $FV(\xi) \subseteq \mathsf{Q}$.

We let $\varphi[\psi/x]$ denote the formula φ, with every free occurrence of x replaced by the formula ψ; for the time being[3] we only apply this substitution operation if ψ is *free for x* in φ, meaning that no free variable of ψ gets bound after substituting. Formally we say that ψ is *free for x in ξ* if ξ is positive in x and for every variable $y \in FV(\psi)$, every occurrence of x in a subformula $\eta y.\chi$ of ξ is in the scope of a fixpoint operator λx in ξ, i.e., bound in ξ by some occurrence of λx. With this constraint, we inductively define the *substitution* $[\psi/z]$ as the following partial operation on μML:

$$\begin{array}{lcl}
x[\psi/z] & := & \begin{cases} \psi & \text{if } x = z \\ x & \text{if } x \neq z \end{cases} \\
(\heartsuit\varphi)[\psi/z] & := & \heartsuit\varphi[\psi/z] \\
(\varphi_0 \odot \varphi_1)[\psi/z] & := & \varphi_0[\psi/z] \odot \varphi_1[\psi/z] \\
(\eta x.\varphi)[\psi/z] & := & \begin{cases} \eta x.\varphi & \text{if } x = z \\ \eta x.\varphi[\psi/z] & \text{otherwise,} \end{cases}
\end{array}$$

where $\heartsuit \in \{\Diamond, \Box\}$, $\odot \in \{\vee, \wedge\}$ and $\eta \in \{\mu, \nu\}$.

The *unfolding* of a formula $\eta x.\chi$ is the formula $\mathtt{unf}(\eta x\ \chi) := \chi[\eta x.\chi/x]$. Given our constraint on the substitution operation, the unfolding of a formula ξ is only properly defined if ξ is *tidy*,[4] that is, if $FV(\xi) \cap BV(\xi) = \varnothing$. The (Fischer-Ladner) *closure* of a tidy formula $\xi \in \mu\mathsf{ML}$ is the smallest set containing ξ which is closed under taking boolean and modal subformulas, and under taking unfoldings of fixpoint formulas. We will need some detail.

For every tidy formula $\xi \in \mu\mathsf{ML}$ define the set $Clos_0(\xi)$ with the following case distinction:

$$\begin{array}{lcll}
Clos_0(\varphi) & := & \varnothing & (\varphi \in \mathtt{At}(\mathsf{Q})) \\
Clos_0(\varphi_0 \odot \varphi_1) & := & \{\varphi_0, \varphi_1\} & (\odot \in \{\wedge, \vee\}) \\
Clos_0(\heartsuit\varphi) & := & \{\varphi\} & (\heartsuit \in \{\Diamond, \Box\}) \\
Clos_0(\eta x.\varphi) & := & \{\varphi[\eta x.\varphi/x]\} & (\eta \in \{\mu, \nu\}).
\end{array}$$

We write $\xi \to_C \varphi$ if $\varphi \in Clos_0(\xi)$ and refer to $\to_C$ as the *trace* relation on μML. We define the relation $\twoheadrightarrow_C$ as the reflexive and transitive closure of $\to_C$, and define $Clos(\xi) := \{\varphi \mid \xi \twoheadrightarrow_C \varphi\}$; formulas in this set are said to be *derived* from ξ. Given a set of tidy formulas Ψ, we put $Clos(\Psi) :=$

[3] This constraint saves us from involving alphabetic variants when substituting. After we have introduced α-equivalence, we can lift this constraint, extending substitution to a total operation in Definition 3.9.

[4] In the literature, some authors make a distinction between proposition letters (which can only occur freely in a formula), and propositional variables, which can be bound. Our tidy formulas correspond to *sentences* in this approach, that is, formulas without free variables.

$\bigcup_{\psi \in \Psi} Clos(\psi)$. We call the set $Clos(\xi)$ the *closure* of ξ. The *closure graph* of ξ is the directed graph $(Clos(\xi), E^C_\xi)$, where E^C_ξ is the trace relation $\to_C$, restricted to the set $Clos(\xi)$. Finally, we call a $\to_C$-path $\psi_0 \to_C \psi_1 \to_C \cdots \to_C \psi_n$ a *(finite) trace* . We can use induction on the length of traces originating at ξ to prove statements about formulas in $Clos(\xi)$. It is easy to show that all formulas in $Clos(\xi)$ are tidy.

The size of a formula can be measured in at least three different ways: First, there is the *length* $|\xi|^\ell$ of the formula $\xi \in \mu\mathsf{ML}$ which is defined in the obvious way as the length of the string (or tree) representation of ξ. Alternatively, the *subformula size* of a (clean) formula ξ is defined as the number of its subformulas: $|\xi|^s := |Sfor(\xi)|$; and the *closure size* of a (tidy) formula ξ is simply given as the size of its closure:

$$|\xi|^c := |Clos(\xi)|.$$

Next to its size, the most important complexity measure of a μ-calculus formula is its *alternation depth*. There are various ways to make this notion precise; here we shall work with the most widely used definition from Niwiński [18]. By natural induction we define classes $\Theta^\mu_n, \Theta^\nu_n$ of μ-calculus formulas. With $\eta, \lambda \in \{\mu, \nu\}$ arbitrary, we set:

1. all atomic formulas belong to Θ^η_0;
2. if $\varphi_0, \varphi_1 \in \Theta^\eta_n$, then $\varphi_0 \vee \varphi_1, \varphi_0 \wedge \varphi_1, \Diamond\varphi_0, \Box\varphi_0 \in \Theta^\eta_n$;
3. if $\varphi \in \Theta^\eta_n$ then $\overline{\eta}x.\varphi \in \Theta^\eta_n$;
4. if $\varphi(x), \psi \in \Theta^\eta_n$, then $\varphi[\psi/x] \in \Theta^\eta_n$, provided that ψ is free for x in φ;
5. all formulas in Θ^λ_n belong to Θ^η_{n+1}.

The *alternation depth* $ad(\xi)$ of a formula ξ is defined as the least n such that $\xi \in \Theta^\mu_n \cap \Theta^\nu_n$.

Intuitively, the class Θ^η_n consists of those μ-calculus formulas where n bounds the length of any alternating nesting of fixpoint operators of which the most significant formula is an η-formula. The alternation depth is then the maximal length of an alternating nesting of fixpoint operators.

As an example, consider the formula

$$\xi = \mu x.\nu y.(\Box y \wedge \mu z.(\Diamond x \vee z)),$$

which looks like a $\mu/\nu/\mu$-formula in the sense that it contains a nested fixpoint chain $\mu x/\nu y/\mu z$. However, the variable y does not occur in the subformula $\mu z.(\Diamond x \vee z)$, and so we may in fact consider ξ as a μ/ν-formula. Formally, we observe that $\mu z.\Diamond x \vee z \in \Theta^\nu_0 \subseteq \Theta^\nu_1$ and $\nu y.\Box y \wedge p \in \Theta^\mu_0 \subseteq \Theta^\nu_1$; from this it follows by the substitution rule that the formula $\nu y.(\Box y \wedge \mu z.(\Diamond x \vee z))$ belongs to the set Θ^ν_1 as well; from this we easily conclude that $\xi \in \Theta^\nu_1$. It is not hard to show that $\xi \notin \Theta^\mu_1$, but since $\xi \in \Theta^\mu_2 \cap \Theta^\nu_2$ we find $ad(\xi) = 2$.

2.2 Compositional semantics of the μ-calculus

The modal μ-calculus is interpreted over Kripke structures. A *Kripke structure* or *transition system* over a set Q of proposition letters is a triple $\mathbb{S} = (S, R, V)$ where S is a set of *states*, $R \subseteq S \times S$ is a binary relation, and $V : \mathsf{Q} \to \mathsf{P}(S)$ is called a Q-*valuation* on S. A *pointed* Kripke structure is a pair $(\mathbb{S}, s)$ where $s \in S$ is a designated state. Given a Kripke structure $\mathbb{S} = (S, R, V)$, a variable x and a set $A \subseteq S$, we define $V[x \mapsto A]$ as the $\mathsf{Q} \cup \{x\}$-valuation given by

$$V[x \mapsto A](p) := \begin{cases} A & \text{if } p = x, \\ V(p) & \text{if } p \neq x \end{cases}$$

and we let $\mathbb{S}[x \mapsto A]$ denote the structure $(S, R, V[x \mapsto A])$.

The semantics of the μ-calculus is defined as follows. By induction on the complexity of μ-calculus formulas, we define a meaning function $[\![\cdot]\!]$, which assigns to a formula $\varphi \in \mu\mathsf{ML}(\mathsf{Q})$ its *meaning* $[\![\varphi]\!]^{\mathbb{S}} \subseteq S$ in any Kripke model $\mathbb{S} = (S, R, V)$ over Q.

$$[\![p]\!]^{\mathbb{S}} := V(p) \quad [\![\overline{p}]\!]^{\mathbb{S}} := S \setminus V(p) \quad [\![\bot]\!]^{\mathbb{S}} := \varnothing \quad [\![\top]\!]^{\mathbb{S}} := S$$

$$[\![\varphi \vee \psi]\!]^{\mathbb{S}} := [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} \qquad [\![\varphi \wedge \psi]\!]^{\mathbb{S}} := [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}}$$

$$[\![\Diamond\varphi]\!]^{\mathbb{S}} := \{s \in S \mid R[s] \cap [\![\varphi]\!]^{\mathbb{S}} \neq \varnothing\}$$

$$[\![\Box\varphi]\!]^{\mathbb{S}} := \{s \in S \mid R[s] \subseteq [\![\varphi]\!]^{\mathbb{S}}\}$$

$$[\![\mu x.\varphi]\!]^{\mathbb{S}} := \bigcap\{A \in \mathsf{P}(S) \mid [\![\varphi]\!]^{\mathbb{S}[x \mapsto A]} \subseteq A\}$$

$$[\![\nu x.\varphi]\!]^{\mathbb{S}} := \bigcup\{A \in \mathsf{P}(S) \mid [\![\varphi]\!]^{\mathbb{S}[x \mapsto A]} \supseteq A\}.$$

If a state $s \in S$ belongs to the set $[\![\varphi]\!]^{\mathbb{S}}$, we write $\mathbb{S}, s \Vdash \varphi$, and say that s *satisfies* φ. Two formulas φ and ψ are *equivalent*, notation: $\varphi \equiv \psi$, if $[\![\varphi]\!]^{\mathbb{S}} = [\![\psi]\!]^{\mathbb{S}}$ for any structure $\mathbb{S}$.

2.3 Parity formulas

In this paper we take the *parity formulas* of [16] as a uniform, graph-based representation of μ-calculus formulas. Generalising the usual tree-based representation of formulas, parity formulas are defined as arbitrary graphs where the vertices are labeled with logical connectives. Additionally, parity formulas come with a priority map to ensure that despite their cyclic nature they have a well-defined semantics in terms of parity games.

Definition 2.1. *A* parity formula over Q *is a quintuple* $\mathbb{G} = (V, E, L, \Omega, v_I)$, *where*

- (V, E) *is a finite, directed graph;*
- $L : V \to \mathtt{At}(\mathsf{Q}) \cup \{\wedge, \vee, \Diamond, \Box, \epsilon\}$ *is a labelling function;*
- $\Omega : V \overset{\circ}{\to} \omega$ *is a partial map, the* priority *map of* $\mathbb{G}$*; and*
- v_I *is a vertex in* V*, referred to as the* initial *node of* $\mathbb{G}$*;*

such that (with $E[v] := \{u \in V \mid Evu\}$*):*

1. $|E[v]| \leq 2$ *for every vertex* v*;* $|E[v]| = 0$ *if* $L(v) \in \mathtt{At}(\mathsf{Q})$*, and* $|E[v]| = 1$ *if* $L(v) \in \{\Diamond, \Box\} \cup \{\epsilon\}$*;*
2. *every cycle of* (V, E) *contains at least one node in* $\mathsf{Dom}(\Omega)$.

The elements of $\mathsf{Dom}(\Omega)$ *are called* states.

The semantics of parity formulas is defined in terms of the following parity game.

Definition 2.2. *Let* $\mathbb{S} = (S, R, U)$ *be a model, and let* $\mathbb{G} = (V, E, L, \Omega, v_I)$ *be a parity formula. We define the* model checking game $\mathcal{E}(\mathbb{G}, \mathbb{S})$ *as the parity game of which the board (or*

Position		Player	Moves
(v, s)	with $L(v) = p$ and $s \in U(p)$	$\forall$	$\varnothing$
(v, s)	with $L(v) = p$ and $s \notin U(p)$	$\exists$	$\varnothing$
(v, s)	with $L(v) = \overline{p}$ and $s \in U(p)$	$\exists$	$\varnothing$
(v, s)	with $L(v) = \overline{p}$ and $s \notin U(p)$	$\forall$	$\varnothing$
(v, s)	with $L(v) = \epsilon$	-	$E[v] \times \{s\}$
(v, s)	with $L(v) = \vee$	$\exists$	$E[v] \times \{s\}$
(v, s)	with $L(v) = \wedge$	$\forall$	$E[v] \times \{s\}$
(v, s)	with $L(v) = \Diamond$	$\exists$	$E[v] \times R[s]$
(v, s)	with $L(v) = \Box$	$\forall$	$E[v] \times R[s]$

Table 1. The model checking game $\mathcal{E}(\mathbb{G}, \mathbb{S})$ of Definition 2.2.

arena) consists of the set $V \times S$*, the priority map* $\Omega' : V \times S \to \omega$ *is given by putting* $\Omega'(v, s) := \Omega(v)$ *if* $v \in \mathsf{Dom}(\Omega)$ *and* $\Omega'(v, s) := 0$ *otherwise, and the game graph is given in Table 1.* $\mathbb{G}$ holds at *or* is satisfied by *the pointed model* $(\mathbb{S}, s)$*, notation:* $\mathbb{S}, s \Vdash \mathbb{G}$*, if the pair* (v_I, s) *is a winning position for* $\exists$ *in* $\mathcal{E}(\mathbb{G}, \mathbb{S})$.

Parity formulas can be seen as variations of Wilke's alternating tree automata [11, 22], but they are also closely related to hierarchical equation systems [2, 8, 19], and μ-calculus in vectorial form [2]. For a detailed discussion of these connections we refer to [16].

The main reason to prefer parity formulas[5] over these other representations is that, given the straightforward definition of their semantics in terms of parity games, they allow for a clear and completely perspicuous definition of their most relevant complexity measures: size and index. The size of a parity formula is simply the number of its vertices and its index corresponds to the maximal length of a suitably defined alternating chain in the range of its priority map.

For this reason, parity formulas serve as an ideal *yardstick* for comparing various complexity measures of standard formulas. In particular, we can use parity formulas to define the notion of a *size measure* for μ-calculus formulas. Say that a parity formula $\mathbb{G}$ *represents*[6] a formula $\xi \in \mu\mathsf{ML}$ if $\mathbb{G}$ and ξ are equivalent (in the obvious way of being satisfied by the same pointed models). Then we call an attribute $s : \mu\mathsf{ML} \to \omega$ a *size measure* of μ-calculus formulas if (†) it is *induced* by some representation $\xi \mapsto \mathbb{G}_\xi$ in the sense that $s(\xi) = |\mathbb{G}_\xi|$. For instance, the following fact from [16] indicates that closure size is a size measure indeed.

Fact 2.3. *[16] There is an effective way to represent any tidy formula* ξ *by a parity formula* $\mathbb{G}_\xi = (Clos(\xi), E^C_\xi, \Omega_\xi, \xi)$*, of which the index is bounded by the alternation depth of* ξ.

[5] Nothing in our paper hinges on this choice, all results can be formulated in terms of alternating tree automata or hierarchical equation systems as well.

[6] This notion of representation is quite weak. In practice we shall focus on constructions that preserve quite a bit of the syntactic structure of the standard formula, but we do not need to adapt the definition accordingly.

3 Alphabetic equivalence

In formalisms that feature some kind of variable binding, the meaning of a syntactic expression usually does not depend on the exact choice of its bound variables. In such a setting α-equivalent formulas, i.e., formulas that can be obtained from one another by a suitable renaming of bound variables, are often taken to be identical. In this section we formally introduce the notion of alphabetical equivalence, and we prove some of its basic properties. We discuss its impact of α-equivalence on the notion of closure, and quickly use it to extend the operation of substitution to a total operation.

Definitions

Let us start with giving a proper definition of the notion of alphabetic equivalence.

Definition 3.1. *An equivalence relation* $\sim$ *on the set* $\mu\mathsf{ML}$ *of formulas will be called a* (syntactic) congruence *if it satisfies the following two conditions:*

1) if $\varphi_0 \sim \psi_0$ *and* $\varphi_1 \sim \psi_1$ *then* $\varphi_0 \odot \varphi_1 \sim \psi_0 \odot \psi_1$*, for* $\odot \in \{\vee, \wedge\}$*;*

2) if $\varphi \sim \psi$ *then* $\heartsuit\varphi \sim \heartsuit\psi$*, for* $\heartsuit \in \{\Diamond, \Box\}$.

We define the relation $=_\alpha$ *as the smallest congruence* $\sim$ *on* $\mu\mathsf{ML}$ *which is closed under the rule:*

3) if $\varphi_0[z/x_0] \sim \varphi_1[z/x_1]$*, where* z *is* fresh *for* φ_0 *and* φ_1*, then* $\eta x_0.\varphi_0 \sim \eta x_1.\varphi_1$*, for* $\eta \in \{\mu, \nu\}$.

The α-equivalence class *or* α-cell *of a formula* φ *is denoted as* $(\!|\varphi|\!)$*. If* $\varphi =_\alpha \psi$ *we call* φ *and* ψ α-equivalent*, or* alphabetic variants *of one another. A* renaming *is a map assigning an alphabetic variant to every formula.*

The following definition will play a prominent role.

Definition 3.2. *Call a set* Φ *of* μ*-calculus formulas* lean *if the relations of* α*-equivalence and syntactic identity coincide on* Φ.

It will be convenient to have a formal system in place by which we can *derive* the α-equivalence of two formulas — this will enable us to prove statements about $=_\alpha$ using induction on the complexity of such derivations.

Definition 3.3. *With* $\doteq$ *denoting a formal identity symbol, an* equation *is an expression of the form* $\varphi \doteq \psi$ *with* $\varphi, \psi \in \mu\mathsf{ML}$*. We define* $\vdash_\alpha$ *as the derivation system on such equations, which consists of the axiom* $\varphi \doteq \varphi$ *and the obvious rules corresponding to the conditions 1) – 3) above. In case an equation* $\varphi \doteq \psi$ *is derivable in this system we write* $\vdash_\alpha \varphi \doteq \psi$.

Note that the absence of rules for symmetry or transitivity in $\vdash_\alpha$ makes the system a very useful proof tool. This absence is justified by the following proposition.

Proposition 3.4. *The derivation system* $\vdash_\alpha$ *for* $=_\alpha$ *is sound and complete for* α*-equivalence, that is, for any pair of* $\mu\mathsf{ML}$*-formulas* φ, ψ *we have*

$$\varphi =_\alpha \psi \text{ iff } \vdash_\alpha \varphi \doteq \psi.$$

Proof. Soundness, i.e., the implication from right to left, is obvious. For the opposite implication, one shows by induction on φ that $\vdash_\alpha \varphi \doteq \psi$ and $\vdash_\alpha \psi \doteq \xi$ imply $\vdash_\alpha \varphi \doteq \xi$, which obviously implies that the relation generated by $\vdash_\alpha$-deductions is transitive. Similarly, one can show that the relation of $\vdash_\alpha$-derivable equivalence is symmetric. From this it is immediate that $\varphi =_\alpha \psi$ implies $\vdash_\alpha \varphi \doteq \psi$ as required. □

In the sequel we will use the above proposition without warning; we will also be somewhat sloppy concerning notation and terminology, for instance allowing ourselves to write that '$\varphi =_\alpha \psi$ is derivable' if we mean that $\vdash_\alpha \varphi \doteq \psi$.

Basic observations

We first provide some key information about α-equivalence. The first proposition states that many basic concepts of μ-calculus formulas are invariant under α-equivalence (here $\mathrm{fd}(\varphi)$ denotes the fixpoint depth of φ).

Proposition 3.5. *The following hold, for any pair φ_0, φ_1 of μ-calculus formulas:*

1. *if $\varphi_0 =_\alpha \varphi_1$ then $\varphi_0 \equiv \varphi_1$;*
2. *if $\varphi_0 =_\alpha \varphi_1$ then $|\varphi_0|^\ell = |\varphi_1|^\ell$;*
3. *if $\varphi_0 =_\alpha \varphi_1$ then $FV(\varphi_0) = FV(\varphi_1)$;*
4. *if $\varphi_0 =_\alpha \varphi_1$ then $\mathrm{fd}(\varphi_0) = \mathrm{fd}(\varphi_1)$;*
5. *if $\varphi_0 =_\alpha \varphi_1$ then $ad(\varphi_0) = ad(\varphi_1)$.*

Below we gather some technical observations, which are used in the proof of Proposition 3.5 and in many of the proofs in the next section. Some of these observation are of some interest in their own right, such as item (10) stating that $=_\alpha$ is a congruence with respect to the unfolding operation.

Proposition 3.6. *Let $\varphi, \varphi_0, \varphi_1, \psi, \psi_0, \psi_1$ and χ be μ-calculus formulas, and let $\eta, \eta_0, \eta_1 \in \{\mu, \nu\}$. Then the following hold:*

1) *if $\varphi =_\alpha \psi$ then $\varphi[z/x] =_\alpha \psi[z/x]$ for any z that is fresh for φ and ψ;*
2) *if $\eta_0 x_0.\varphi_0 =_\alpha \psi_1$ then ψ_1 is of the form $\psi_1 = \eta_1 y.\varphi_1$, where $\eta_0 = \eta_1$;*
3) *if $\eta x_0.\varphi_0 =_\alpha \eta x_1.\varphi_1$ then $\varphi_0[z/x_0] =_\alpha \varphi_1[z/x_1]$, for any fresh variable z;*
4) *if $\eta x.\varphi_0 =_\alpha \eta x.\varphi_1$ then $\varphi_0 =_\alpha \varphi_1$;*
5) *if $\eta x.\varphi_0 \odot \varphi_1 =_\alpha \eta y.\psi_0 \odot \psi_1$ then $\eta x.\varphi_i =_\alpha \eta y.\psi_i$, for $i \in \{0,1\}$ and $\odot \in \{\wedge, \vee\}$;*
6) *if $\eta x.\heartsuit\varphi =_\alpha \eta y.\heartsuit\psi$ then $\eta x.\varphi =_\alpha \eta y.\psi$ for $\heartsuit \in \{\Diamond, \Box\}$;*
7) *if $\eta x.\lambda z.\varphi =_\alpha \eta y.\lambda z.\psi$ then $\eta x.\varphi =_\alpha \eta y.\psi$ for $\lambda \in \{\mu, \nu\}$;*
8) *if $\varphi =_\alpha \psi$, $y \notin FV(\varphi)$ and y is free for x in ψ, then $\eta x.\varphi =_\alpha \eta y.\psi[y/x]$;*
9) *if $\varphi_0 =_\alpha \varphi_1$, $\psi_0 =_\alpha \psi_1$ and ψ_i is free for x in φ_i, then $\varphi_0[\psi_0/x] =_\alpha \varphi_1[\psi_1/x]$;*
10) *if $\eta x_0.\varphi_0 =_\alpha \eta x_1.\varphi_1$ for tidy formulas $\eta x_i.\varphi_i$ then $\varphi_0[\eta x_0.\varphi_0/x_0] =_\alpha \varphi_1[\eta x_1 \varphi_1/x_1]$;*
11) *if $\varphi_0 =_\alpha \varphi_1$ then $\eta x.\varphi_0 =_\alpha \eta x.\varphi_1$.*
12) *if $\varphi =_\alpha \psi[\chi/x]$, then $\varphi = \psi'[\chi'/x']$ for some formulas ψ', χ' and a fresh variable x' such that $\psi =_\alpha \psi'[x/x']$ and $\chi =_\alpha \chi'$.*

The proof of these propositions can be found in the technical report [?].

Alphabetic equivalence and size measures

Although α-equivalent formulas have the same *length*, their *(closure or subformula) sizes* may differ exponentially. The following observation by Kupke, Marti & Venema [16], which was mentioned in the introduction, states that the commonly made assumption that in the μ-calculus one may without loss of generality work with *clean* formulas, is not as innocent as it may seem when it comes to size considerations.

Proposition 3.7. *There is a family $(\xi_n)_{n\in\omega}$ of tidy formulas such that $|\xi_n|^c \leq 2\cdot n$, while for any sequence of clean formulas χ_n such that $\xi_n =_\alpha \chi_n$ for all n, we have $|\chi_n|^c \geq 2^n$.*

Proposition 3.7 also indicates that closure size is not such an appealing size measure since it is not α-invariant: α-equivalent but distinct formulas may have distinct sizes. In fact, closure size fails to be α-invariant for another reason as well: the closure of a formula may contain α-equivalent but distinct formulas.

In case one wants to define a succinct α-invariant size measure, the following proposition (which was discussed as statement (1) in the introduction) is a promising first step.

Proposition 3.8. *Let ξ_0 and ξ_1 be tidy μ-calculus formulas such that $\xi_0 =_\alpha \xi_1$. Then*
1) for every $\varphi_0 \in Clos(\xi_0)$ there is a $\varphi_1 \in Clos(\xi_1)$ such that $\varphi_0 =_\alpha \varphi_1$, and vice versa;
2) as a corollary, $|Clos(\xi_0)/{=_\alpha}| = |Clos(\xi_1)/{=_\alpha}|$.

Proof. We prove part 1) of this proposition by induction on the length of the shortest trace from ξ_0 to φ_0. In the base case we have $\varphi_0 = \xi_0$, so that we may take $\varphi_1 := \xi_1$.

In the inductive case we assume some formula $\psi_0 \in Clos(\xi_0)$ which can be reached by a shorter trace from ξ_0 and is such that φ_0 is either (1/2) a direct modal or boolean subformula of ψ_0 or else (3) ψ_0 is a fixpoint formula $\eta x_0.\chi_0$ of which φ_0 is the unfolding. An instance of the first case is where ψ_0 is of the form $\Diamond\varphi_0$. By the induction hypothesis this formula has an alphabetic variant ψ_1 in the closure set of ξ_1; it is then easy to see that ψ_1 must be of the form $\Diamond\varphi_1$ for some formula φ_1. But then it is immediate that $\varphi_1 \in Clos(\xi_1)$ and that $\varphi_1 =_\alpha \varphi_0$, as required. The case where φ_0 is a boolean subformula of ψ_0 is dealt with in a similar way, and in the third case we use Proposition 3.6(10)).

For part 2) of the proposition, observe that as an immediate consequence of part 1), we find a bijection between the sets of $Clos(\xi_0)/{=_\alpha}$ and $Clos(\xi_1)/{=_\alpha}$. □

Part 2) of the Proposition states that *up to α-equivalence*, the closure sets of α-equivalent formulas have the same size, as announced in the introduction. A natural suggestion would then be to take the number of α-cells of its closure as the size of a formula; this would certainly provide a fully α-invariant notion of size. Note however, that Proposition 3.8 on its own is not enough to consider the proposed definition as a proper *size measure*. The problem is that it is not a priori clear that the definition meets our requirement (†) that any reasonable size measure should be based on some transformation of a μ-calculus formula into an equivalent parity formula. As we will see in the next section, this is where Theorem 1.1 comes in.

Substitution revisited

As promised in section 2, we will now provide a proper definition of the substitution operation $[\psi/x]$, i.e., one that is also applicable to formulas φ in which ψ is *not* free for x, in a way that avoids variable capture. Our approach here is completely standard.

Definition 3.9. *Given two μ-calculus formulas φ and ψ, we define*

$$\varphi[\psi/x] := \begin{cases} \varphi[\psi/x] & \text{if } \psi \text{ is free for } x \text{ in } \varphi \\ \mathrm{ren}_\psi(\varphi)[\psi/x] & \text{otherwise.} \end{cases}$$

where we let $\mathrm{ren}_\psi(\varphi)$ be a canonically chosen alphabetic variant of φ such that ψ is free for x in $\mathrm{ren}_\psi(\varphi)$.

4 α-Invariance via skeletal renaming

The aim of this section is to provide a renaming function which maps an arbitrary μ-calculus formula ξ to an alphabetic variant $\widehat{\xi}$ satisfying the conditions (3) stating that the map $\widehat{\cdot}$ picks a fixed element of every α-cell, and (4) requiring that for every formula $\xi \in \mu\mathsf{ML}$, the closure of its renaming $\widehat{\xi}$ is *lean* (i.e., α-equivalence is the identity relation on $Clos(\widehat{\xi})$). As we saw in the introduction, this suffices to prove the main theorem of the paper.

The key concept involved in the definition of $\widehat{\xi}$ will be that of a *skeletal* (set of) formula(s), to be introduced in Definition 4.3 below, and the key property that we shall need of skeletal formulas is that they have a lean closure, as stated in Proposition 4.11. We then proceed to defining the *skeletal renaming* $\widehat{\cdot}$, of which we subsequently prove that it is, indeed, a renaming, and satisfies the conditions (3) and (4). We finish the section by providing, in Definition 4.20, a new and fully α-invariant size measure, and we show that it has some desirable properties, for instance in relation to the substitution operation.

Skeletal formulas

Throughout this section we fix a placeholder variable s, which we assume to be 'fresh' in the sense that it does not occur in any formula in $\mu\mathsf{ML}$.[7]

[7]To do this in a precise way we could introduce the set $\mu\mathsf{ML}_s$ of formulas that are allowed to contain the placeholder s as a special variable.

Definition 4.1. *Given a set U of variables, we define the* skeleton $\mathrm{sk}_U(\varphi)$ *of a formula φ relative to a set of variables U by induction on the structure of φ. Throughout this induction we will define*

$$\mathrm{sk}_U(\varphi) \quad := \quad s \quad \text{if } U \cap FV(\varphi) = \varnothing,$$

so that in the inductive definition itself we may focus on the case where $U \cap FV(\varphi) \neq \varnothing$:

$$\begin{array}{llll} \mathrm{sk}_U(x) & := & x & \text{for } x \in U \\ \mathrm{sk}_U(\varphi_0 \odot \varphi_1) & := & \mathrm{sk}_U(\varphi_0) \odot \mathrm{sk}_U(\varphi_1) & (\odot \in \{\vee, \wedge\}) \\ \mathrm{sk}_U(\heartsuit\varphi) & := & \heartsuit \mathrm{sk}_U(\varphi) & (\heartsuit \in \{\Diamond, \Box\}) \\ \mathrm{sk}_U(\eta z.\varphi) & := & \eta z.\mathrm{sk}_{U \cup \{z\}}(\varphi) & (\eta \in \{\mu, \nu\}) \end{array}$$

For a single variable x we write sk_x as abbreviation for $\mathrm{sk}_{\{x\}}$.

The intuition behind this map is that we replace 'U-free' subformulas, that is, subformulas not taking any free variable from the set U, with the place holder s, and that this set U of critical variables grows by collecting bound variables as we move down (i.e., away from the root) in the syntax tree of the formula. A couple of examples are in order.

Example 4.2. *1) Let $\varphi = p \vee \Diamond x$, then $\mathrm{sk}_x(\varphi) = s \vee \Diamond x$.*
2) Let $\varphi = ((p \vee \mu z.(q \wedge \Box z)) \wedge \mu y.((q \vee \Diamond y) \vee \Box x))$, then $\mathrm{sk}_x(\varphi) = s \wedge \mu y.((s \vee \Diamond y) \vee \Box x)$.

Definition 4.3. *We call a set of formulas Φ* skeletal *if for any pair of formulas $\varphi_0 = \eta_0 x_0.\psi_0$ and $\varphi_1 = \eta_1 x_1.\psi_1$ in $\bigcup_{\varphi \in \Phi} Sfor(\varphi)$ we have*

$$x_0 = x_1 \quad \text{iff} \quad \eta_0 x_0.\mathrm{sk}_{x_0}(\psi_0) =_\alpha \eta_1 x_1.\mathrm{sk}_{x_1}(\psi_1). \tag{5}$$

We will call a single formula ξ skeletal if the singleton $\{\xi\}$ is skeletal.

Intuitively, the formula $\eta x.\mathrm{sk}_x(\varphi)$ is obtained by leaving every part of $\eta x.\varphi$ that has some bearing on choosing a suitable alternative name for the bound variable x unchanged, but replacing every other part with the placeholder s. In more technical terms, the function sk ensures that all elements of $Clos(\eta x.\mathrm{sk}_x(\varphi))$ are either equal to s or contain $\eta x.\mathrm{sk}_x(\varphi)$ as a subformula.

Example 4.4. *Consider the formulas $\alpha = \mu x.\nu y.\Diamond x \wedge \Box y$, and $\beta = \nu y'.\Diamond\alpha \wedge \Box y'$. Clearly β is an alphabetic variant of the unfolding $\mathsf{unf}(\alpha) = \nu y.\Diamond\alpha \wedge \Box y$ of α.*

The formula α is obviously skeletal. It is not hard to see that $\mathsf{unf}(\alpha)$ is skeletal as well, since the skeletons of both the outer and the inner ν-formula are equal to $\nu y.s \wedge \Box y$. We leave it as an exercise for the reader to verify that the set $Clos(\alpha)$ is lean.

Now consider the formula $\beta \vee \alpha$. Its subformulas β and α bear witness to the fact that $\beta \vee \alpha$ is not skeletal. In line with this, the closure of $\beta \vee \alpha$ contains both the formula β and its alphabetic variant $\mathsf{unf}(\alpha)$, and thus it will not be lean.

Our skeletal renaming of $\beta \vee \alpha$, on the other hand, will ensure that the variables y and y' are renamed into a single variable z, as both relevant ν-subformulas of $\beta \vee \alpha$ have a skeleton of the form $\nu y.s \wedge \Box y$.

Basic observations

We will now see in detail that skeletal formulas have indeed the desired properties. We start with some basic observations about the skeletal function. The proof of the first Proposition is straightforward — we omit the details.

Proposition 4.5. *Let φ be a formula and let $x \notin FV(\varphi)$. Then*

$$\mathrm{sk}_{U\cup\{x\}}(\varphi) = \mathrm{sk}_U(\varphi).$$

Proposition 4.6. *Let ψ be a formula, let U be a set of variables and let x be a variable with $x \notin U$. Furthermore let β be a formula which is free for x in ψ, and such that $U \cap FV(\beta) = \emptyset$. Then*

$$\mathrm{sk}_U(\psi) = \mathrm{sk}_U(\psi[\beta/x]). \tag{6}$$

In particular, if x and y are variables such that $x, y \notin U$ and y is free for x in ψ, then $\mathrm{sk}_U(\psi) = \mathrm{sk}_U(\psi[y/x])$

Proof. Consider first the case where $U \cap FV(\psi) = \emptyset$. We have $FV(\psi[\beta/x]) \subseteq FV(\psi) \cup FV(\beta)$, which, together with our assumption on $FV(\beta)$, implies $U \cap (FV(\psi[\beta/x])) = \emptyset$. Therefore we obtain $\mathrm{sk}_U(\psi) = s = \mathrm{sk}_U(\psi[\beta/x])$ as required.

In the case that $U \cap FV(\psi) \neq \emptyset$ the claim is proved by induction on ψ. In the base step of the induction, we make a case distinction. If $\psi \neq x$ then $\psi = \psi[\beta/x]$ so that (6) follows immediately. If, on the other hand, we have $\psi = x$, then $\mathrm{sk}_U(x) = s = \mathrm{sk}_U(\beta) = \mathrm{sk}_U(x[\beta/x])$, where the second equality holds as $U \cap FV(\beta) = \emptyset$.

The boolean and modal cases are easy. For instance, in the case of a Boolean operator, we have $\psi = \psi_0 \odot \psi_1$, with $\odot \in \{\wedge, \vee\}$. By our assumption that $U \cap FV(\psi) \neq \emptyset$, there is an i with $U \cap FV(\psi_i) \neq \emptyset$. Now for $j \in \{0, 1\}$ we may use the induction hypothesis in the case that $U \cap FV(\psi_j) \neq \emptyset$, and the fact that the lemma is already proved for the case that $U \cap FV(\varphi_i) = \emptyset$. Using these facts, we find

$$\begin{aligned}
&\mathrm{sk}_U(\psi_0 \odot \psi_1) \\
&= sk_U(\psi_0) \odot \mathrm{sk}_U(\psi_1) && (U \cap FV(\psi) \neq \emptyset) \\
&= sk_U(\psi_0[\beta/x]) \odot \mathrm{sk}_U(\psi_1[\beta/x]) && \text{(explained above)} \\
&= sk_U(\psi_0[\beta/x] \odot \psi_1[\beta/x]) && (U \cap FV(\psi[\beta/x]) \neq \emptyset) \\
&= sk_U\big((\psi_0 \odot \psi_1)[\beta/x]\big) && \text{(definition substitution)}
\end{aligned}$$

Finally, in the case that $\psi = \eta z.\varphi$, we recall that $U \cap FV(\eta z.\varphi) \neq \emptyset$, and calculate

$$\begin{aligned}
\mathrm{sk}_U(\eta z.\varphi) &= \eta z.\mathrm{sk}_{U\cup\{z\}}(\varphi) \\
&= \eta z.\mathrm{sk}_{U\cup\{z\}}(\varphi[\beta/x]) && \text{(IH)} \\
&= \mathrm{sk}_U(\eta z.\varphi[\beta/x]) && (*)
\end{aligned}$$

Observe that the induction hypothesis is applicable, since by assumption β is free for x in ψ, which implies that $z \notin FV(\beta)$. The final equality (*) uses the fact that $\emptyset \neq U \cap FV(\eta z.\varphi) \subseteq U \cap FV(\eta z.\varphi[\beta/x])$, which holds since by assumption $x \notin U$. □

Proposition 4.7. *Let φ be a formula, let U be a set of variables, and let x and z be variables such that $x \in U$, $z \notin U \cup FV(\varphi)$ and z is free for x in φ. Then*

$$\mathrm{sk}_U(\varphi)[z/x] = \mathrm{sk}_{U[z/x]}(\varphi[z/x])$$

where $U[z/x] := (U \setminus \{x\}) \cup \{z\}$.

Proof. In the case that $x \notin FV(\varphi)$ we also have $x \notin FV(\mathrm{sk}_U(\varphi))$ and thus $(\mathrm{sk}_U(\varphi))[z/x] = \mathrm{sk}_U(\varphi)$. In addition,

$$\mathrm{sk}_{U[z/x]}(\varphi[z/x]) = \mathrm{sk}_{U[z/x]}(\varphi) = \mathrm{sk}_U(\varphi)$$

where the last equality follows from Proposition 4.5 as z and x do not occur freely in φ.

If, on the other hand, we have that $x \in FV(\varphi)$ we prove the claim by induction on φ. In the base case of this induction, where $\varphi = x$, the claim is an easy calculation: $(\mathrm{sk}_U(x)[z/x] = z = \mathrm{sk}_{U[z/x]}(x[z/x])$.

If $\varphi = \varphi_1 \odot \varphi_2$ with $\odot \in \{\vee, \wedge\}$, we have

$$\begin{aligned}
\mathrm{sk}_U(\varphi_1 \odot \varphi_2)[z/x] &= \mathrm{sk}_U(\varphi_1)[z/x] \odot \mathrm{sk}_U(\varphi_2)[z/x] \\
&\overset{(*)}{=} \mathrm{sk}_{U[z/x]}(\varphi_1[z/x]) \odot \mathrm{sk}_{U[z/x]}(\varphi_2[z/x]) \\
&= \mathrm{sk}_{U[z/x]}((\varphi_1 \odot \varphi_2)[z/x])
\end{aligned}$$

where (*) is either by the induction hypothesis, or by the previous case if x does not occur in φ.

The case where $\varphi = \heartsuit\psi$ for $\heartsuit \in \{\Box, \Diamond\}$ is similar to the previous one.

Finally, we consider the case where $\varphi = \eta y.\psi$. By our assumptions we have $y \neq x$ since $x \in FV(\varphi)$ and — as z is free for x in φ — we also have $z \neq y$. We calculate:

$$\begin{aligned}
\mathrm{sk}_U(\eta y.\psi)[z/x] &= \big(\eta y.\mathrm{sk}_{U\cup\{y\}}(\psi)\big)\,[z/x] && \text{(Def. sk)} \\
&= \eta y.\,\big(\mathrm{sk}_{U\cup\{y\}}(\psi)[z/x]\big) && \text{(IH)} \\
&= \eta y.\,\big(\mathrm{sk}_{(U\cup\{y\})[z/x]}(\psi[z/x])\big) && \text{(Def. sk)} \\
&= \mathrm{sk}_{U[z/x]}(\eta y.\psi[z/x]).
\end{aligned}$$

□

Proposition 4.8. *Let φ_0 and φ_1 be formulas such that $\varphi_0 =_\alpha \varphi_1$ and let U be a set of variables. Then $\mathrm{sk}_U(\varphi_0) =_\alpha \mathrm{sk}_U(\varphi_1)$.*

Proof. Assume that $\varphi_0 =_\alpha \varphi_1$, then clearly $FV(\varphi_0) = FV(\varphi_1)$ and so we find $U \cap FV(\varphi_0) = \emptyset$ iff $U \cap FV(\varphi_1) = \emptyset$. This means that in case $U \cap FV(\varphi_0) = \emptyset$ we have $\mathrm{sk}_U(\varphi_0) = \mathrm{sk}_U(\varphi_1) = s$.

In case that $U \cap FV(\varphi_0) \neq \emptyset$ we prove the claim by induction on the length of φ_0. We only treat the fixpoint case, that is, where φ_0 is of the form $\varphi_0 = \eta x_0.\psi_0$. As $\varphi_0 =_\alpha \varphi_1$ the formula φ_1 must be of the form $\varphi_1 = \eta x_1.\psi_1$.

Fix a fresh variable z, then we have (*) $\psi_i = \psi_i[z/x_i][x_i/z]$. We will now bring each $\mathrm{sk}_U(\varphi_i)$, for $i = 0, 1$, into a certain shape. Note that by Proposition 4.5 we may without loss of generality assume that $x_i \notin U$. Then we may calculate

$$\begin{aligned} \mathrm{sk}_U(\varphi_i) &= \eta x_i.\mathrm{sk}_{U\cup\{x_i\}}(\psi_i) && \text{(Def. of sk)}\\ &= \eta x_i.\mathrm{sk}_{U\cup\{x_i\}}(\psi_i[z/x_i][x_i/z]) && (*)\\ &= \eta x_i.(\mathrm{sk}_{U\cup\{z\}}(\psi_i[z/x_i]))[x_i/z] && \text{(Prop. 4.7)}\\ &=_\alpha \eta z.\mathrm{sk}_{U\cup\{z\}}(\psi_i[z/x_i]) && \text{(Prop. 3.6(8))} \end{aligned}$$

Now observe that by Proposition 3.6(3)) it follows from $\eta x_0.\psi_0 =_\alpha \eta x_1.\psi_1$ that $\psi_0[z/x_1] =_\alpha \psi_1[z/x_1]$. Hence by the induction hypothesis we obtain that $\mathrm{sk}_{U\cup\{z\}}(\psi_0[z/x_0]) =_\alpha \mathrm{sk}_{U\cup\{z\}}(\psi_1[z/x_1])$, so that by Proposition 3.6(11)), we find that

$$\eta z.\mathrm{sk}_{U\cup\{z\}}(\psi_0[z/x_0]) =_\alpha \eta z.\mathrm{sk}_{U\cup\{z\}}(\psi_1[z/x_1]).$$

But then from the above calculation of $\mathrm{sk}_U(\varphi_i)$ we may conclude that $\mathrm{sk}_U(\varphi_0) =_\alpha \mathrm{sk}_U(\varphi_1)$, as required. □

Skeletal Formulas & Their Closure

The key property of skeletal formulas is that they have lean closure sets. To prove this, we first show that skeletal sets of formulas are lean themselves.

Proposition 4.9. *Let Φ be a skeletal set of formulas. Then Φ is lean.*

Proof. Suppose Φ is skeletal. We will show that in fact

$$\varphi_0 =_\alpha \varphi_1 \quad \text{implies} \quad \varphi_0 = \varphi_1. \tag{7}$$

holds for every pair of formulas $\varphi_0, \varphi_1 \in \bigcup_{\varphi\in\Phi} Sfor(\varphi)$. This proves the Proposition, since obviously $\Phi \subseteq \bigcup_{\varphi\in\Phi} Sfor(\varphi)$. Our proof of (7) proceeds by induction on the structure of φ_0.

If φ_0 is a literal, the claim is trivial. In case φ_0 is a conjunction, disjunction or a modal formula of the form $\heartsuit\psi_1$, the claim easily follows by induction.

Now suppose that $\varphi_0 = \eta_0 x_0.\psi_0$, Then by Proposition 3.6(2), φ_1 must be of the form $\varphi_1 = \eta_1 x_1.\psi_1$, where $\eta_0 = \eta_1$ — so that we may write η in the sequel. By Proposition 3.6(3) we have $\psi_0[z/x_0] =_\alpha \psi_1[z/x_1]$ for a fresh variable z. It follows from Proposition 4.8 that $\mathrm{sk}_z(\psi_0[z/x_0]) =_\alpha \mathrm{sk}_z(\psi_1[z/x_1])$. By Proposition 4.7 we have $\mathrm{sk}_z(\psi_i[z/x_i]) = \mathrm{sk}_{x_i}(\psi_i)[z/x_i]$ for $i = 0, 1$. Therefore, as z was fresh, we obtain $\eta x_0.\mathrm{sk}_{x_0}(\psi_0) =_\alpha \eta x_1.\mathrm{sk}_{x_1}(\psi_1)$ by definition of $=_\alpha$. As Φ is skeletal this implies $x_0 = x_1 = x$ and thus by Proposition 3.6(4)) that $\psi_0 =_\alpha \psi_1$. The induction hypothesis yields $\psi_0 = \psi_1$ which obviously implies $\varphi_0 = \varphi_1$ as required. □

The next proposition states that the closure of a skeletal set is skeletal.

Proposition 4.10. *Let Ψ be a skeletal set of tidy formulas. Then $Clos(\Psi)$ is skeletal as well.*

Proof. Clearly it suffices to show that, if Φ' is obtained from a skeletal set Φ of tidy formulas by applying one of the rules for deriving the closure, then Φ' is also skeletal.

The only case where this is non-trivial is when $\Phi' = \Phi \cup \{\varphi[\eta x\varphi/x]\}$ for some formula $\eta x.\varphi \in \Phi$. Consider a pair of formulas $\varphi_0 = \eta_0 x_0.\psi_0$ and $\varphi_1 = \eta_1 x_1.\psi_1$ that are subformulas of some formulas in Φ'. In order to show that φ_0 and φ_1 satisfy (5), we distinguish the following cases.

Case 1: Both φ_0 and φ_1 are subformulas of formulas in Φ. Then (5) follows from the fact that Φ is skeletal.

Case 2: Neither φ_0 nor φ_1 is a subformula of a formula in Φ. In this case, both φ_0 and φ_1 are subformulas of $\varphi[\eta x\varphi/x]$, and since they cannot be subformulas of $\eta x\varphi/x \in \Phi$, this means that φ_0 and φ_1 are of the form $\varphi_0 = \eta_0 x_0.\psi_0'[\eta x.\varphi/x]$ and $\varphi_1 = \eta_1 x_1.\psi_1'[\eta x.\varphi/x]$, respectively, for subformulas $\eta_0 x_0.\psi_0'$ and $\eta_1 x_1.\psi_1'$ of φ. Then we have $\mathrm{sk}_{x_i}(\psi_i'[\eta x.\varphi/x]) = \mathrm{sk}_{x_i}(\psi_i')$ for $i \in \{0, 1\}$ by Proposition 4.6. Thus we find

$$\begin{aligned} &x_0 = x_1\\ &\text{iff } \eta_0 x_0.\mathrm{sk}_{x_0}(\psi_0') =_\alpha \eta_1 x_1.\mathrm{sk}_{x_1}(\psi_1')\\ &\text{iff } \eta_0 x_0.\mathrm{sk}_{x_0}(\psi_0'[\eta x.\varphi/x]) =_\alpha \eta_1 x_1.\mathrm{sk}_{x_1}(\psi_1'[\eta x.\varphi/x]), \end{aligned}$$

where the first equivalence is a consequence of the fact that property (5) holds for Φ by assumption.

Case 3: Exactly one of φ_0 and φ_1 is a subformula of a formula in Φ. Say, without loss of generality, that φ_0 is a subformula of a formula in Φ, while (reasoning as in the previous case) $\varphi_1 = \eta_1 x_1.\psi'[\eta x.\varphi/x]$ with $\eta_1 x_1.\psi' \trianglelefteq \varphi$. As Φ is skeletal we have

$$x_0 = x_1 \text{ iff } \eta_0 x_0.\mathrm{sk}_{x_0}(\psi_0) = \eta_1 x_1.\mathrm{sk}_{x_1}(\psi').$$

By Proposition 4.6 we have $\mathrm{sk}_{x_1}(\psi') = \mathrm{sk}_{x_1}(\psi'[\eta x.\varphi/x]) = \mathrm{sk}_{x_1}(\psi_1)$ and thus we obtain

$$x_0 = x_1 \text{ iff } \eta_0 x_0.\mathrm{sk}_{x_0}(\psi_0) = \eta_1 x_1.\mathrm{sk}_{x_1}(\psi_1)$$

as required. □

As an immediate consequence of the Propositions 4.10 and 4.9, we establish the key property of skeletal formulas.

Proposition 4.11. *Let φ be a tidy skeletal formula. Then the set $Clos(\varphi)$ is lean.*

The skeletal renaming

We are now ready to define the renaming map $\widehat{\cdot}$. It will be convenient to introduce a set Z of fresh variables from which we will draw the bound variables of the formulas $\widehat{\xi}$.

Definition 4.12. *Let X and Z be two (disjoint) sets of variables. We let $\mu\mathsf{ML}_X$ denote the set of μ-calculus formulas taking their variables (free or bound) from X, and we let $\mu\mathsf{ML}_{X,Z}$ denote the set of formulas ξ in $\mu\mathsf{ML}_{X\cup Z}$ such that $BV(\xi) \subseteq X$.*

In the definition below we assume that the set Z contains a distinct variable z_E for every α-equivalence class E of $\mu\mathsf{ML}_X$-formulas.

Definition 4.13. *We define the renamed version $\widehat{\varphi} \in \mu\mathsf{ML}_{Z,X}$ of a formula $\varphi \in \mu\mathsf{ML}_X$ as follows:*

$$
\begin{array}{lll}
\widehat{\varphi} & := & \varphi \qquad (\varphi \text{ atomic}) \\
\widehat{\heartsuit\varphi} & := & \heartsuit\widehat{\varphi} \qquad (\heartsuit \in \{\Diamond, \Box\}) \\
\widehat{\varphi_0 \odot \varphi_1} & := & \widehat{\varphi_0} \odot \widehat{\varphi_1} \qquad (\odot \in \{\vee, \wedge\}) \\
\widehat{\eta x.\varphi} & := & \eta z_E.\widehat{\varphi}[z_E/x] \qquad (\eta \in \{\mu, \nu\})
\end{array}
$$

where, in the last clause, $E = (\!|\eta x.\mathrm{sk}_x(\varphi)|\!)$.

Remark 4.14. *The renamed version of $\varphi \in \mu\mathsf{ML}_X$ will only contain variables from the set Z that are bound. These bound variables can be replaced by fresh variables from X in order to obtain a renamed version in $\mu\mathsf{ML}_X$.*

Example 4.15. *1) Compare the formulas ξ_0 and ξ_1, where*

$$\xi_i := \mu x_i.\nu y_i.\Diamond x_i \vee (p \wedge \Box y_i),$$

for $i \in \{0,1\}$. We will abbreviate $\psi_i := \Diamond x_i \vee (p \wedge \Box y_i)$. Clearly we have $\xi_0 =_\alpha \xi_1$, and so we want to obtain $\widehat{\xi_0} = \widehat{\xi_1}$.

To see that this will indeed be the case, observe that

$$
\begin{array}{lll}
\mathrm{sk}_{x_i}(\nu y_i.\psi_i) & = & \nu y_i.\Diamond x_i \vee (s \wedge \Box y_i) \\
\mathrm{sk}_{y_i}(\psi_i) & = & s \vee (s \wedge \Box y_i)
\end{array}
$$

Defining $E_i := (\!|\mu x_i.\nu y_i.\Diamond x_i \vee (s \wedge \Box y_i)|\!)$ and $F_i := (\!|\nu y_i.s \vee (s \wedge \Box y_i)|\!)$, we observe that these definitions in fact do not depend on i, so that we may simply denote these α-cells as E and F, respectively. We then compute, for each $i \in \{0,1\}$:

$$
\begin{aligned}
\widehat{\xi_i} &= \mu z_E.\widehat{\nu y_i.\psi_i}[z_E/x_i] \\
&= \mu z_E.\Big(\nu z_F.\widehat{\psi_i}[z_F/y_i]\Big)[z_E/x_i] \\
&= \mu z_E.\Big(\nu z_F.\psi_i[z_F/y_i]\Big)[z_E/x_i] \\
&= \mu z_E.\Big(\nu z_F.\Diamond x_i \vee (p \wedge \Box z_F)\Big)[z_E/x_i] \\
&= \mu z_E.\nu z_F.\Diamond z_E \vee (p \wedge \Box z_F)
\end{aligned}
$$

and find that $\widehat{\xi_0} = \widehat{\xi_1}$ as desired.

2) Now consider the formula

$$\varphi = \nu y.(\Diamond(\mu x.(\nu z.\Diamond(x \wedge z)) \wedge y)),$$

which is α-equivalent to the unfolding $(\nu y.\Diamond(x \wedge y))[\psi/x]$ of $\psi = \mu x.\nu y.\Diamond(x \wedge y)$. Furthermore let $E_1 = (\!|\nu y.\Diamond(s \wedge y)|\!)$ and $E_2 = (\!|\mu x.\nu y.\Diamond(x \wedge y)|\!)$. Then

$$\widehat{\varphi} = \nu z_{E_1}.(\Diamond(\mu z_{E_2}.(\nu z_{E_1}.\Diamond(z_{E_2} \wedge z_{E_1})) \wedge z_{E_1})),$$

where we point out the re-use of the variable z_{E_1}. Note that φ is an example where $Clos(\widehat{\varphi})$ is properly smaller than $Clos(\varphi)$.

Our first goal is to show that the map $\widehat{\cdot}$ is indeed a renaming, i.e., that the renamed version $\widehat{\varphi}$ of a formula φ is α-equivalent to φ. To this aim we need the following rather technical lemma.

Proposition 4.16. *Let x and y be variables, let U be a set of variables with $y \in U$, and let φ and $\eta x.\psi$ be formulas such that $y \in FV(\eta x.\psi)$ and $\eta x.\psi \trianglelefteq \varphi$, while there is no formula of the form $\lambda y.\chi$ such that $\eta x.\psi \trianglelefteq \lambda y.\chi \trianglelefteq \varphi$. Then $\mathrm{sk}_x(\psi) \neq_\alpha \mathrm{sk}_U(\varphi)$.*

Proof. By Proposition 3.5 it suffices to show that

$$|\mathrm{sk}_x(\psi)|^\ell < |\mathrm{sk}_U(\varphi)|^\ell,$$

and we will prove this by induction of the length of the shortest direct-subformula chain $\eta x.\psi \triangleleft_0 \cdots \triangleleft_0 \varphi$ witnessing that $\eta x.\psi$ is a subformula of φ. Further details can be found in [?]. □

Proposition 4.17. *Let ξ be a μ-calculus formula. Then $\widehat{\xi}$ is tidy and $\xi =_\alpha \widehat{\xi}$.*

Proof. The proof that $\widehat{\xi}$ is tidy is easy and therefore left to the reader. We prove the claim that $\xi =_\alpha \widehat{\xi}$ by a formula induction on ξ. If ξ is atomic, then ξ and $\widehat{\xi}$ are identical, and so, certainly α-equivalent.

For the induction step, distinguish cases. If ξ is of the form $\xi = \xi_0 \odot \xi_1$ for $\odot \in \{\wedge, \vee\}$, then the claim is an immediate consequence of the induction hypothesis and the fact that $\widehat{\xi_0 \odot \xi_1} = \widehat{\xi_0} \odot \widehat{\xi_1}$. The case where ξ is of the form $\xi = \heartsuit\xi'$ for $\heartsuit \in \{\Diamond, \Box\}$ is equally simple.

The interesting case is where ξ is of the form $\xi = \lambda y.\varphi$. Then $\widehat{\xi} = \lambda z_E.\widehat{\varphi}[z_E/y]$, with $E = (\!|\lambda y.\mathrm{sk}_y(\varphi)|\!)$. We first claim that

$$z_E \text{ is free for } y \text{ in } \widehat{\varphi}. \tag{8}$$

To see this, suppose for contradiction that y occurs freely in the scope of a binder ηz_E in $\widehat{\xi}$. Then there must be a subformula $\eta x.\psi$ of φ with $\widehat{\eta x.\psi} = \eta z_E.\widehat{\psi}[z_E/x]$ such that $y \in FV(\psi)$. By definition of $\widehat{\cdot}$ we have $E = (\!|\eta x.\mathrm{sk}_x(\psi)|\!)$ and so $\eta x.\mathrm{sk}_x(\psi) =_\alpha \eta y.\mathrm{sk}_y(\varphi)$ by our assumption that $E = (\!|\eta y.\mathrm{sk}_y(\varphi)|\!)$. It follows by Proposition 4.16 that there must be a formula $\lambda y.\chi$ such that $\eta x.\psi \trianglelefteq \lambda y.\chi \trianglelefteq \varphi$; without loss of generality we may take $\lambda y.\chi$ to be the smallest such formula (in terms of the subformula ordering). But from this we may infer that actually, when computing the formula $\widehat{\xi}$, the variable $y \in FV(\eta x.\psi)$ will be replaced by the variable $z_{E'}$, where $E' = (\!|\lambda y.\mathrm{sk}_y(\chi)|\!)$. In other words, the alleged free occurrence in $\widehat{\xi}$ of the variable y, within the scope of a binder ηz_E, is not actually possible. Clearly this implies (8).

From this we reason as follows. By the induction hypothesis we obtain that $\widehat{\varphi} =_\alpha \varphi$. Now, because of (8), we may apply Proposition 3.6(8) and obtain $\widehat{\xi} = \lambda z_E.\widehat{\varphi}[z_E/y] =_\alpha \lambda y.\varphi = \xi$ as required. □

We now show that the renaming operation always produces skeletal formulas.

Proposition 4.18. *Let φ be a μ-calculus formula. Then $\widehat{\varphi}$ is skeletal.*

Proof. As a preparatory step, consider an arbitrary subformula of $\widehat{\varphi}$ of the form $\eta z_E.\psi$. By definition of $\widehat{\cdot}$ there is a subformula $\eta x.\xi$ of φ such that $E = (\!|\eta x.\mathrm{sk}_x(\xi)|\!)$ and $\psi = \widehat{\xi}[z_E/x][z_1/x_1]\dots[z_n/x_n]$. Then we have

$$\begin{array}{lll} & \eta z_E.\mathrm{sk}_{z_E}(\psi) & \\ = & \eta z_E.\mathrm{sk}_{z_E}(\widehat{\xi}[z_E/x][z_1/x_1]\dots[z_n/x_n]) & \\ = & \eta z_E.\mathrm{sk}_{z_E}(\widehat{\xi}[z_E/x]) & \text{(Prop. 4.6)} \\ = & \eta z_E.\mathrm{sk}_x(\widehat{\xi})[z_E/x] & \text{(Prop. 4.7)} \\ =_\alpha & \eta x.\mathrm{sk}_x(\widehat{\xi}) & \\ =_\alpha & \eta x.\mathrm{sk}_x(\xi) & \text{(Prop. 4.17)} \end{array}$$

where the last statement uses the instantiation of Proposition 4.8 stating that $\varphi_0 =_\alpha \varphi_1$ implies $\mathrm{sk}_x(\varphi_0) =_\alpha \mathrm{sk}_x(\varphi_1)$.

We now turn to the argument as to why $\widehat{\varphi}$ is skeletal. Suppose that we have two subformulas $\eta_0 z_{E_0}.\psi_0$ and $\eta_1 z_{E_1}.\psi_1$ of $\widehat{\varphi}$. We need to prove that

$$z_{E_0} = z_{E_1} \text{ iff } \eta_0 z_{E_0}.\mathrm{sk}_{z_{E_0}}(\psi_0) =_\alpha \eta_1 z_{E_1}.\mathrm{sk}_{z_{E_1}}(\psi_1). \quad (9)$$

By the earlier observation there must be formulas $\eta_i x_i.\xi_i \trianglelefteq \varphi$ such that, with $E_i = (\!|\eta_i x_i.\mathrm{sk}_{x_i}(\xi_i)|\!)$, we have $\eta_i z_{E_i}.\mathrm{sk}_{z_{E_i}}(\psi_i) =_\alpha \eta_i x_i.\mathrm{sk}_{x_i}(\xi_i)$.

In order to prove (9), first assume that $z_{E_0} = z_{E_1}$. Then $E_0 = E_1$, so that $\eta_0 x_0.\mathrm{sk}_{x_0}(\xi_0) =_\alpha \eta_1 x_1.\mathrm{sk}_{x_1}(\xi_1)$. It follows that $\eta_0 = \eta_1$ and so we find

$$\begin{aligned} \eta_0 z_{E_0}.\mathrm{sk}_{z_{E_0}}(\psi_0) &=_\alpha \eta_0 x_0.\mathrm{sk}_{x_0}(\xi_0) \\ &=_\alpha \eta_1 x_1.\mathrm{sk}_{x_1}(\xi_1) \\ &=_\alpha \eta_1 z_{E_1}.\mathrm{sk}_{z_{E_1}}(\psi_1) \end{aligned}$$

as required.

Conversely, if $\eta z_{E_1}.\mathrm{sk}_{z_{E_1}}(\psi_1) =_\alpha \eta z_{E_2}.\mathrm{sk}_{z_{E_2}}(\psi_2)$, then we have $\eta x_1.\mathrm{sk}_{x_1}(\xi_1) =_\alpha \eta x_2.\mathrm{sk}_{x_2}(\xi_2)$ which implies $E_1 = E_2$ and thus $z_{E_1} = z_{E_2}$. □

What is left to show is condition (3).

Proposition 4.19. *Let ξ_0 and ξ_1 be formulas such that $\xi_0 =_\alpha \xi_1$. Then $\widehat{\xi_0} = \widehat{\xi_1}$.*

Proof. We can use a trick here. Let ξ_0 and ξ_1 be formulas such that $\xi_0 =_\alpha \xi_1$, and consider the formula $\xi := \xi_0 \wedge \xi_1$. Since we have $\widehat{\xi} = \widehat{\xi_0} \wedge \widehat{\xi_1}$, both formulas $\widehat{\xi_0}$ and $\widehat{\xi_1}$ belong to the closure of $\widehat{\xi}$. But $\widehat{\xi}$ is skeletal by Proposition 4.18, so $Clos(\widehat{\xi})$ must be lean by Proposition 4.11. In particular, this means that $\widehat{\xi_0} = \widehat{\xi_1}$, as required. □

Summarizing properties of the skeletal renaming

We now briefly check that the map $\widehat{\cdot} : \mu\mathsf{ML}_X \to \mu\mathsf{ML}_{Z,X}$ has all the properties that are required for the proof of Theorem 1.1. First of all, we proved in Proposition 4.17 that $\widehat{\cdot}$ is indeed a renaming, which takes care of (2). The same Proposition also states that $\widehat{\xi}$ is always tidy (even if ξ itself is not). We saw in Proposition 4.19 that $\widehat{\cdot}$ maps α-equivalent formulas to the same representative element of their $=_\alpha$-cell, which means that $\widehat{\cdot}$ meets condition (3). Finally, as an immediate consequence of Proposition 4.18 and Proposition 4.11 we see that it also satisfies (4): for every μ-calculus formula ξ, the closure of its renaming $\widehat{\xi}$ is lean indeed.

An α-invariant size measure

Recall that a size measure for μ-calculus formulas is an attribute $s : \mu\mathsf{ML} \to \omega$ that is induced by some representation $\xi \mapsto \mathbb{G}_\xi$ of $\mu\mathsf{ML}$-formulas as parity formulas in the sense that $s(\xi) = |\mathbb{G}_\xi|$. In the previous section we saw that although closure size is a suitable size measure, it is not α-invariant. As a further contribution of this paper, we can now provide the definitions of a size measure that is invariant under alphabetic equivalence, and defined for arbitrary (i.e., not necessarily tidy) formulas.

Definition 4.20. *We define the* size *of a μ-calculus formula ξ by putting*

$$|\xi| := |Clos(\xi)/{=_\alpha}|. \quad (10)$$

Theorem 4.21. *The map $|\cdot|$ provides an α-invariant size measures for μ-calculus formulas.*

Proof. As in the proof of Theorem 1.1 (given in the introduction), we define $\mathbb{P}_\xi := \mathbb{G}_{\widehat{\xi}}$ for any μ-calculus formula ξ, where $\xi \mapsto \mathbb{G}_\xi$ is the construction referred to in Fact 2.3. Since $\mathbb{P}_\xi$ is equivalent to ξ, in order to prove the Proposition it suffices to show that

$$|\xi| = |\mathbb{P}_\xi|.$$

But this is rather straightforward:

$$\begin{array}{rll} |\xi| &= |Clos(\xi)/{=_\alpha}| & \text{(def. } |\cdot|) \\ &= |Clos(\widehat{\xi})/{=_\alpha}| & \text{(Prop's 4.17 \& 3.8)} \\ &= |Clos(\widehat{\xi})| & \text{(Prop's 4.18 \& 4.11)} \\ &= |\mathbb{P}_\xi| & \text{(Fact 2.3)} \end{array}$$

Finally, the α-invariance of $|\cdot|$ as a size measures is immediate by its definition and Proposition 3.8. □

The following observation shows that the size measure (10) interacts nicely with the notion of substitution (as defined in the previous section for arbitrary formulas). Its proof can be found in the technical report [?].

Proposition 4.22. *Let ξ and ψ be μ-calculus formulas. Then*

$$|\xi[\psi/x]| \le |\xi| + |\psi|. \quad (11)$$

5 Conclusion

5.1 Main conclusion

The algorithms that are used to solve computational problems related to the modal μ-calculus generally do not take the formulas themselves as input, but operate on some kind of graph representation of standard formulas. In this paper we studied the impact of alphabetic equivalence on a uniform representation of this kind: parity formulas. Our main result, Theorem 1.1, states that with a μ-calculus formula ξ, we may associate a parity formula of size at most $|Clos(\xi)/{=_\alpha}|$ and index at most $ad(\xi)$. As a consequence, complexity results that are rooted in algorithms operating on parity formulas (or on alternating tree automata or hierarchical equation systems) can be formulated without ambiguity for standard μ-calculus formulas, where the size measure of a formula $\xi \in \mu\mathsf{ML}$ is taken to be the number of formulas in the closure of ξ, *up to alphabetic equivalence*.

5.2 Discussion: other ways to represent α-cells

In the introduction to this paper we already mentioned the existence of alternative proofs of our main result, Theorem 1.1. As in the approach followed in this paper, the idea underlying these alternative proofs is to construct, given a fixed but arbitrary formula ξ, a parity formula $\mathbb{P}_\xi$ of which the vertices somehow represent the $=_\alpha$-cells of the set $Clos(\xi)$.

To motivate alternative approaches, it can be argued that the representation of $=_\alpha$-cells via the renaming map $\widehat{\cdot}$ is somewhat arbitrary. One might prefer a more canonical representation, for instance one that uses so-called *de Bruijn indices*. These originate from the theory of the *λ-calculus* [7] and provide a tool for writing down expressions (in a language that features binding) without naming the bound variables.

Concretely, de Bruijn indices are natural numbers that represent bound variables. More specifically, an occurrence of an index n in an expression represents the variable that is bound at the unique place in the construction tree that is reached from the occurrence by moving up, in the syntax tree of the expression, until the n-th binder is reached. As an example, the μ-calculus formula $\mu x\,(\Diamond x \wedge \nu y\, \Box((x \wedge y) \vee \nu x\,(x \wedge p)))$ would be written as $\mu\,(\Diamond 1 \wedge \nu\, \Box((2 \wedge 1) \vee \nu(1 \wedge p)))$ using de Bruijn indices. Here, the key feature of interest of this tool is that

$$\xi_0 =_\alpha \xi_1 \text{ iff } \xi_0^{\mathsf{dB}} = \xi_1^{\mathsf{dB}}$$

where ξ^{dB} denotes the formula $\xi \in \mu\mathsf{ML}$, converted into de Bruijn format.

Based on this observation one could take, for the carrier of the parity formula $\mathbb{P}_\xi$, the set $[Clos(\xi)]^{\mathsf{dB}} := \{\psi^{\mathsf{dB}} \mid \psi \in Clos(\xi)\}$, which has the same cardinality as the set $Clos(\xi)/{=_\alpha}$. Alternatively, one might set out to construct the parity formula $\mathbb{P}_\xi$ *inside the de Bruijn version of the modal μ-calculus*, i.e., start with defining the closure set $Clos^{\mathsf{dB}}(\xi^{\mathsf{dB}})$ of the de Bruijn conversion ξ^{dB} of ξ, and then redo the construction of [16] on the basis of this set. This would certainly be interesting but also a rather formidable undertaking since it would involve the development of a "de Bruijn version" of the entire syntactic framework of the modal μ-calculus. We leave this as an interesting direction for further research.

Next to using de Bruijn indices, there are other ways to associate the vertices of a parity formula $\mathbb{P}_\xi$ with the $=_\alpha$-cells of the closure of ξ. For instance, one might work directly with the $=_\alpha$-cells themselves, or equivalently, construct $\mathbb{P}_\xi$ on the basis of identifying α-equivalent formulas throughout. This approach would also be interesting and certainly closer to the principle of α-invariance as formulated in the introduction. On the other hand, it also might involve cumbersome technicalities since the construction would undoubtedly involve working with concrete formulas (as opposed to their equivalence classes). Furthermore, note that in some sense, our approach here takes care of such technicalities by means of the renaming function $\widehat{\cdot}$.

Before finishing this discussion of alternative constructions supporting the proof of Theorem 1.1, however, we want to stress that the importance of the result lies in the *existence* of a parity formula $\mathbb{P}_\xi$ satisfying the conditions listed in its statement. The question as to *how exactly* the $=_\alpha$-cells are represented in $\mathbb{P}_\xi$ is of secondary importance. After all, the names of the vertices of $\mathbb{P}_\xi$ are nothing more than mere place holders, so that a priori there is no added benefit if these place holders are variable-free formulas. In particular, one should see the parity formula $\mathbb{P}_\xi$ *itself* as a variable-free representation of the μ-calculus formula ξ.

5.3 Suggestions for further research

Here are two other directions for further research. First, we focussed on the *closure graph* of a μ-calculus formula rather than its subformula dag, since (by the results of Bruse, Friedmann & Lange [4]) the closure graph can be exponentially more succinct. Nevertheless, one may have reasons to work with the subformula dag (corresponding to measuring a formula by its subformula-size), and still be interested in a (relatively) succinct, α-invariant way of representing formulas. In fact, similar to the skeletal renaming $\widehat{\cdot}$, one may define a renaming $\widetilde{\cdot}$ of μ-calculus formulas with the properties that $\xi_0 =_\alpha \xi_1$ iff $\widetilde{\xi_0} = \widetilde{\xi_1}$, and alphabetic equivalence is the identity relation on the collection of subformulas of $\widetilde{\xi}$ (i.e., $Sfor(\widetilde{\xi})$ is lean). We hope to get back to this in future work.

Second, parity formulas, combining features of formulas and automata, are interesting objects in their own right. A first step in the development of their theory would be the definition of appropriate notions of *morphisms* and structural equivalence relations ("bisimulations") between parity formulas. It would then be of particular interest to study the notion of alphabetic equivalence in this light, as well as the skeletal renaming introduced in this paper.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions. The research of the first author was funded by a Leverhulme Trust Research Project Grant, project nr. RPG-2020-232. The research of the second author has been made possible by a grant from the Dutch Research Council NWO, project nr. 617.001.857.

References

[1] B. Afshari and G. Leigh. 2017. Cut-free Completeness for Modal Mu-Calculus. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic In Computer Science (LICS'17)*. IEEE Computer Society, 1–12.

[2] A. Arnold and D. Niwiński. 2001. *Rudiments of μ-calculus*. Studies in Logic and the Foundations of Mathematics, Vol. 146. North-Holland Publishing Co., Amsterdam.

[3] J. Bradfield and C. Stirling. 2006. Modal μ-calculi. In *Handbook of Modal Logic*, J. van Benthem, P. Blackburn, and F. Wolter (Eds.). Elsevier, 721–756.

[4] F. Bruse, O. Friedmann, and M. Lange. 2015. On guarded transformation in the modal μ-calculus. *Logic Journal of the IGPL* 23, 2 (2015), 194–216.

[5] C.S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. 2017. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, (STOC 2017)*, H. Hatami, P. McKenzie, and V. King (Eds.). 252–263.

[6] G. D'Agostino and M. Hollenberg. 2000. Logical questions concerning the μ-calculus. *Journal of Symbolic Logic* 65 (2000), 310–332.

[7] N. G. de Bruijn. 1972. Lambda Calculus notation with nameless dummies: a tool for automatic formula manipulation. *Indagationes Mathematicae* 34 (1972), 381âĂŞ392.

[8] S. Demri, V. Goranko, and M. Lange. 2016. *Temporal Logics in Computer Science: Finite-State Systems*. Cambridge University Press.

[9] E.A. Emerson and C.S. Jutla. 1991. Tree automata, mu-calculus and determinacy (extended abstract). In *Proceedings of the 32nd Symposium on the Foundations of Computer Science*. IEEE Computer Society Press, 368–377.

[10] G. Fontaine and Y. Venema. 2018. Some model theory for the modal mu-calculus: syntactic characterizations of semantic properties. *Logical Mewthods in Computer Science* 14, 1 (2018).

[11] E. Grädel, W. Thomas, and T. Wilke (Eds.). 2002. *Automata, Logic, and Infinite Games*. LNCS, Vol. 2500. Springer.

[12] D. Janin and I. Walukiewicz. 1995. Automata for the modal μ-calculus and related results. In *Proceedings of the Twentieth International Symposium on Mathematical Foundations of Computer Science, MFCS'95 (LNCS)*, Vol. 969. Springer, 552–562.

[13] D. Janin and I. Walukiewicz. 1996. On the Expressive Completeness of the Propositional μ-Calculus w.r.t. Monadic Second-Order Logic. In *Proceedings of the Seventh International Conference on Concurrency Theory, CONCUR '96 (LNCS)*, Vol. 1119. 263–277.

[14] D. Kozen. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science* 27 (1983), 333–354.

[15] D. Kozen and R. Parikh. 1983. A decision procedure for the propositional μ-calculus. In *Proceedings of the Workshop on Logics of Programs 1983 (LNCS)*. 313–325.

[16] C. Kupke, J. Marti, and Y. Venema. 2022. Succinct graph representations of μ-calculus formulas. In *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic, CSL 2022 (LIPIcs)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[17] A. Mader. 1995. Modal μ-Calculus, Model Checking and Gauß Elimination. In *Proceedings of the First International Workshop onTools and Algorithms for Construction and Analysis of Systems, (TACAS '95) (LNCS)*, E. Brinksma, R. Cleaveland, K. G. Larsen, T. Margaria, and B. Steffen (Eds.), Vol. 1019. Springer, 72–88.

[18] D. Niwiński. 1986. On fixed point clones. In *Proceedings of the 13th International Colloquium on Automata, Languages and Programming (ICALP 13) (LNCS)*, L. Kott (Ed.), Vol. 226. 464–473.

[19] H. Seidl and A. Neumann. 1999. On guarding nested fixpoints. In *Proceedings of the 8th EACSL Annual Conference on Computer Science Logic, CSL '99*. 484–498.

[20] C. Stirling. 2001. *Modal and Temporal Properties of Processes*. Springer-Verlag.

[21] I. Walukiewicz. 2000. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Information and Computation* 157 (2000), 142–182.

[22] T. Wilke. 2001. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society* 8 (2001), 359–391.