Generalising Automaticity to Modal Properties of Finite Structures

Anuj Dawar

University of Cambridge Computer Laboratory, Cambridge CB3 0FD, UK.
anuj.dawar@cl.cam.ac.uk

Stephan Kreutzer

Institut für Informatik, Humboldt-Universität zu Berlin, 10099 Berlin, Germany.
kreutzer@informatik.hu-berlin.de

Abstract

We introduce a complexity measure of modal properties of finite structures which generalises the automaticity of languages. It is based on graph-automata like devices called *labelling systems*. We define a measure of the size of a structure that we call *rank*, and show that any modal property of structures can be approximated up to any fixed rank n by a labelling system. The function that takes n to the size of the smallest labelling system doing this is called the *labelling index* of the property. We demonstrate that this is a useful and fine-grained measure of complexity and show that it is especially well suited to characterise the expressive power of modal fixed-point logics. From this we derive several separation results of modal and non-modal fixed-point logics, some of which are already known whereas others are new.

1 Introduction

Modal logics are widely used to express properties of finite (and infinite) state systems for the purpose of automatic verification. In this context, propositional modal logic (also known as Hennessy-Milner logic) is found to be weak in terms of its expressive power and much attention has been devoted to extensions that allow some form of recursion. This may be in the form of path

⋆ Research supported by a Joint Project grant from the Royal Society and by the European Union Research Training Network on GAMES.
[1] First author supported in part by EPSRC grant GR/S06721.

quantifiers as with the branching time temporal logics CTL and CTL* or with a least fixed-point operator as with the μ-calculus. Other extensions have been considered for the purpose of understanding a variety of fixed-point operators or classifying their complexity. Examples include L^ω_μ, the higher dimensional μ-calculus introduced by Otto [10], and MIC, the modal iteration calculus, introduced in [4]. The former was introduced specifically to demonstrate a logic that exactly characterises the polynomial-time decidable bisimulation invariant properties of finite-state systems, while the latter was studied in an investigation into the difference between least and inflationary fixed points.

The study of these various extensions of propositional modal logic has thrown up a variety of techniques for analysing their expressive power. One can often show that one logic is at least as expressive as another by means of an explicit translation of formulas of the second into the first. Establishing separations between logics is, in general, more involved. This requires identifying a property expressible in one logic and proving that it is not expressible in the other. Many specialised techniques have been deployed for such proofs of inexpressibility, including diagonalisation, bisimulation and other Ehrenfeucht-Fraïssé style games, complexity hierarchies and automata-based methods such as the pumping lemma. But, as in other areas of computer science, proving lower bounds is difficult.

In this paper, we introduce an alternative complexity measure for modal properties of finite structures which we call the *labelling index* of the property and demonstrate its usefulness in analysing the expressive power of modal fixed-point logics and proving separations. The labelling index generalises the notion of the automaticity of languages (see [11]). The automaticity of a language (i.e. a set of strings) L is the function that maps n to the size of a minimal deterministic finite automaton which agrees with L on all strings of length n or less. We generalise this notion in two steps, first studying it for classes of finite trees and then for classes of finite, possibly cyclic, transition systems.

The generalisation to trees is straightforward. The automaticity of a class $\mathcal{T}$ of finite trees can be defined as the function that maps n to the size of the smallest tree automaton that agrees with $\mathcal{T}$ on trees of height n or less. In our definition, we use a version of bottom-up tree automata that ensures that the property defined is invariant under bisimulation. This notion of automaticity was used in [4] to establish a separation between the expressiveness of MIC and L^ω_μ. Here, we use it to show that MIC is not the bisimulation-invariant fragment of monadic IFP, addressing a question left open in [4].

In extending the notion of automaticity from trees to more general finite structures, we introduce automata-like devices called labelling systems and a measure on finite structures that we call *rank*. We show that any modal property of finite structures (or equivalently, any class of finite structures closed under bisimulation) can be approximated up to any fixed rank n by a labelling system. The function that takes n to the size of the smallest labelling system that does this is the labelling index of the property. We demonstrate that this is a useful and fine-grained measure of the complexity of modal properties by deriving a number of separation results using it, including some that were previously known and some that are new. We show that any property that is definable in propositional modal logic has constant labelling index. In contrast, any property that is definable in the μ-calculus has polynomial labelling index and moreover, there are properties definable in L_μ whose labelling indices have a linear lower bound. Similarly we obtain exponential upper and lower bounds on the labelling index of properties definable in MIC, generalising results on tree automaticity obtained in [4]. We also investigate the relationship between labelling index and conventional time and space based notions of complexity. For instance, the problem of determining bisimulation equivalence, which is decidable in polynomial time, has the worst possible complexity in terms of its labelling index. On the other hand, coarser equivalence relations, such as trace equivalence, which are computationally intractable, have lower labelling index. Trace equivalence problems of various kinds provide a particularly rich source of examples for exploring the notion of labelling index, and we do this in some detail in the final section.

An extended abstract of the present paper appeared as [5].

2 Background

In this section, we give a brief introduction to modal logic and its various fixed-point extensions. A detailed study of these logics can be found in [2,1,4].

2.1 Propositional Modal Logic

For the rest of the paper fix a set $\mathcal{A}$ of actions and a set $\mathcal{P}$ of atomic propositions. Modal logics are interpreted on *transition systems*, also called *Kripke structures*, which are edge and node labelled graphs. The labels of the edges come from the set $\mathcal{A}$ of actions, whereas the nodes are labelled by sets of propositions from $\mathcal{P}$. Formally, a transition system is a structure $\mathcal{K} := (V, (E_a)_{a\in\mathcal{A}}, (p)_{p\in\mathcal{P}})$ with universe V, a binary relation $E_a^{\mathcal{K}} \subseteq V \times V$ for every $a \in \mathcal{A}$, and a unary relation $p^{\mathcal{K}} \subseteq V$ for each $p \in \mathcal{P}$. When clear from the context, we omit the superscripts and only write E_a and p. At various places we will use two systems $\mathcal{K}$ and $\mathcal{K}'$ simultaneously. In this case, we refer to the relations of $\mathcal{K}'$ as E'_a and p' instead of $E_a^{\mathcal{K}'}$ and $p^{\mathcal{K}'}$. As a final bit of notation, we often write $u \xrightarrow{a} v$ to denote that there is an a-transition from u

to v.

Modal logic (ML) is inductively defined as follows: For every atomic proposition symbol $p \in \mathcal{P}$, p is a formula of ML. And if φ, ψ are ML-formulas then so are $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$ and $\langle a \rangle\varphi, [a]\varphi$ for every $a \in \mathcal{A}$. Formulas $\varphi \in$ ML are always evaluated at a particular node in a transition system. We write $\mathcal{K}, v \models \varphi$ if φ holds at the node v in the transition system $\mathcal{K}$. The semantics of ML-formulas is as usual with $\mathcal{K}, v \models p$ if $v \in p^{\mathcal{K}}$, $\mathcal{K}, v \models \langle a \rangle\varphi$ if there is an a-successor u of v such that $\mathcal{K}, u \models \varphi$ and, dually, $\mathcal{K}, v \models [a]\varphi$ if for all a-successors u of v, $\mathcal{K}, u \models \varphi$.

We are primarily concerned with the expressive power of modal logics on finite structures and we will therefore assume that all structures $\mathcal{K}$ are finite unless explicitly stated otherwise.

2.2 Bisimulations

Bisimulation is a notion of behavioural equivalence for transition systems. Modal logics, like ML, CTL, the μ-calculus etc. do not distinguish between transition systems that are bisimulation equivalent. Formally, given two transition systems $\mathcal{K} := (V, (E_a)_{a\in\mathcal{A}}, (p)_{p\in\mathcal{P}})$ and $\mathcal{K}' := (V', (E_a)_{a\in\mathcal{A}}, (p)_{p\in\mathcal{P}})$, with distinguished states v and v' respectively, we say that $\mathcal{K}, v$ is bisimulation equivalent to $\mathcal{K}', v'$, written $\mathcal{K}, v \sim \mathcal{K}', v'$, if there is a relation $R \subseteq V \times V'$ between the states of $\mathcal{K}$ and the states of $\mathcal{K}'$ such that:

(1) $(v, v') \in R$
(2) for each atomic proposition $p \in \mathcal{P}$ and each $(u, u') \in R$, $u \in p^{\mathcal{K}}$ if, and only if, $u' \in p^{\mathcal{K}'}$
(3) for every action $a \in \mathcal{A}$ and every pair $(u, u') \in R$:
  - for each $t \in V$ such that $(u, t) \in E_a^{\mathcal{K}}$, there is a $t' \in V'$ with $(u', t') \in E_a^{\mathcal{K}'}$ so that $(t, t') \in R$ and
  - for each $t' \in V'$ such that $(u', t') \in E_a^{\mathcal{K}'}$, there is a $t \in V$ with $(u, t) \in E_a^{\mathcal{K}}$ so that $(t, t') \in R$.

For a transition system $\mathcal{K}$ we write $\mathcal{K}_{/\sim}$ for its quotient under bisimulation. That is, $\mathcal{K}_{/\sim}$ is the transition system whose states are the equivalence classes of states of $\mathcal{K}$ under bisimulation and, if $[v]$ denotes the equivalence class containing v, then $[v] \in p^{\mathcal{K}_{/\sim}}$ if $v \in p^{\mathcal{K}}$ and there is an a-transition from $[u]$ to $[v]$ in $\mathcal{K}_{/\sim}$ if, and only if, there is an a-transition from u to some state $w \in [v]$ in $\mathcal{K}$. It is easily verified that $\mathcal{K}, v \sim \mathcal{K}_{/\sim}, [v]$.

We now consider two fixed-point extensions of modal logic: the modal μ-calculus and the modal iteration calculus (MIC).

Definition 1 *The* modal μ-calculus (L_μ) *is inductively defined by the rules for modal logic and the following formula building rule: If $X_1, \ldots, X_k$ are propositional variables and $\varphi_1, \ldots, \varphi_k$ are L_μ-formulas so that no variable X_i occurs negatively in any φ_j, then*

$$S := \begin{cases} X_1 \leftarrow \varphi_1 \\ \quad\vdots \\ X_k \leftarrow \varphi_k \end{cases}$$

is a system *of rules and $(\mu X_i : S)$ and $(\nu X_i : S)$ are formulas of L_μ.*

On any finite transition system $\mathcal{K}$ with universe V, such a system S of rules defines an monotone operator F_S taking a sequence $\overline{X} := (X_1, \ldots, X_k)$ of subsets of V to the sequence $(F_{S_1}(\overline{X}), \ldots, F_{S_k}(\overline{X}))$, where $F_{S_i}(\overline{X}) := \{u : (\mathcal{K}, (X_j)_{1 \leq j \leq k}), u \models \varphi_i\}$. As F_S is monotone, it has a least and a greatest fixed point. The semantics of a formula $(\mu X_i : S)$ is defined as $\mathcal{K}, u \models (\mu X_i : S)$ if, and only if, u occurs in the i^{th} component of the least fixed point of F_S. Analogously, $\mathcal{K}, u \models (\nu X_i : S)$ if, and only if, u occurs in the i^{th} component of the greatest fixed point of F_S.[2]

It is often useful to consider L_μ-formulas in a certain normal form, called *guarded normal form*.

Definition 2 *A μ-calculus formula is* guarded*, if all propositional variables that are bound by and thus occur in the scope of a fixed-point operator are also in the scope of a modal operator (i.e. $\Box$ or $\Diamond$) that is itself in the scope of the fixed-point operator.*

It was shown in [9] that every L_μ formula is equivalent to a guarded formula.

The modal iteration calculus was introduced in [4] as an extension of ML with an operator for inflationary fixed points. Inflationary fixed point operators have been extensively studied in the context of predicate logics. They are designed to overcome the restriction of the application of least fixed point

[2] In most presentations of the μ-calculus simultaneous inductions are not considered. Nothing is lost by such a restriction as the least fixed point of a system S can also be obtained by nested fixed points of simple inductions (see [1]).

operators to positive formulas, while still guaranteeing the existence of a meaningful fixed point.

Definition 3 *The* modal iteration-calculus (MIC) *is inductively defined by the rules for modal logic and the following formula building rule: If $X_1, \dots, X_k$ are propositional variables and $\varphi_1, \dots, \varphi_k$ are* MIC*-formulas, which may contain the variables X_i positively and negatively, then*

$$S := \begin{cases} X_1 \leftarrow \varphi_1 \\ \qquad \vdots \\ X_k \leftarrow \varphi_k \end{cases}$$

is a system *of rules and* $(\mathbf{ifp}\ X_i : S)$ *is a formula of* MIC.

Again, on any finite transition system $\mathcal{K}$ with universe V, such a system S of rules defines an operator F_S as above. However, this operator is no longer guaranteed to be monotone. It does, though, inductively define for each finite ordinal α a sequence of sets $(X_1^\alpha, \dots, X_k^\alpha)$, called the *induction stages*, as follows. For all i, $X_i^0 := \emptyset$ and for $0 < \alpha < \omega$, $X_i^\alpha := X_i^{\alpha-1} \cup F_{S_i}(\overline{X}^{\alpha-1})$.

By definition, the stages of the induction are increasing and lead to a fixed point $(X_1^\infty, \dots, X_k^\infty)$. The semantics of a formula $(\mathbf{ifp}\ X_i : S)$ is defined as $\mathcal{K}, u \models (\mathbf{ifp}\ X_i : S)$ if, and only if, $u \in X_i^\infty$.

By a well known result of Knaster and Tarski, the least fixed point is also reached as the limit $(X_1^\infty, \dots, X_k^\infty)$ of the sequence of stages as defined above, and the greatest fixed point is reached as the limit of a similar sequence of stages, where the induction is not started with the empty set but with the entire universe, i.e. $X_i^0 := V$. In fact, as the operator F_S is monotone, in this case the explicit union of $X_i^{\alpha-1} \cup F_{S_i}(\overline{X}^{\alpha-1})$ to define X_i^α is not neccessary.

Hence, the semantics of a L_μ-calculus formula $(\mu X_i : S)$ can equivalently be defined as $\mathcal{K}, u \models (\mu X_i : S)$ if, and only if, $u \in X_i^\infty$ and analogously for greatest fixed points. It follows, that every L_μ-formula can trivially be translated into an equivalent MIC-formula.

Simple MIC (or simple L_μ) consists of those formulas of MIC (resp. L_μ) in which, in every formula of the form $(\mathbf{ifp}\ X : S)$, S is a system consisting of a single rule $X \leftarrow \varphi$. We generally write such a formula as $(\mathbf{ifp}\ X : \varphi)$. It is well known that simple L_μ is as expressive as L_μ (see, for instance, [1]). However, it is shown in [4] that MIC is strictly more expressive than simple MIC.

Another fixed-point extension of modal logic that we consider is L_μ^ω, the higher-dimensional μ-calculus defined by Otto. We refer the reader to [10] for a precise definition. Here we only note that this logic permits the formation of least fixed points of positive formulas φ defining not a set X, but a relation X of any arity. Otto shows that, restricted to finite structures, this logic can express exactly the bisimulation-closed properties that are polynomial-time decidable.

It is immediate from the definitions that, in terms of expressive power, we have ML $\subseteq L_\mu \subseteq$ MIC $\subseteq$ IFP, where IFP denotes the extension of first-order logic by inflationary fixed points. As IFP is equivalent to least fixed-point logic (LFP) and L^ω_μ is the bisimulation invariant fragment of LFP, it follows that MIC $\subseteq L^\omega_\mu$. Indeed, all of these inclusions are proper. The separations of MIC from L_μ and L^ω_μ were shown in [4]. The analysis of the labelling index of properties expressible in the logics provides a uniform framework for both separations.

There is a natural translation of L_μ formulas into monadic second-order logic (MSO) using the fact that the least fixed point X^∞ of a monotone operator $\mathcal{F} : \mathrm{Pow}(M) \to \mathrm{Pow}(M)$ on a set M is determined by the equation $R = \bigcap\{P \subseteq M : \mathcal{F}(P) \subseteq P\}$. Furthermore, Janin and Walukiewicz [8] show that a formula of monadic second-order logic is bisimulation invariant if, and only if, it is equivalent to a formula of L_μ. This gives a very elegant characterisation of the expressive power of L_μ in terms of a standard *predicate* logic. It is not known whether this characterisation remains true if we restrict ourselves to finite structures. On the other hand, MIC cannot be translated to MSO, whether or not we allow infinite structures. In particular, on finite strings, MIC can define non-regular languages, while it is well known that MSO can only define regular languages. In [4], the question was posed whether MIC could be characterised as the bisimulation invariant fragment of any natural logic. The most natural candidate for this would appear to be the monadic fragment of IFP. In a sense, if MIC were the bisimulation-invariant fragment of any predicate logic, it would be this one. However, by an analysis of the labelling index of properties definable in this logic, we are able to show that it can express bisimulation-invariant properties that are not in MIC.

3 Automaticity on Strings and Trees

Given a fixed finite alphabet Σ, the automaticity of a language $L \subseteq \Sigma^*$ is the function that maps n to the size of a minimal deterministic automaton that agrees with L on all strings of length at most n. This function is eventually constant if, and only if, L is regular and is at most exponential for any language L.

In [4] it was shown that MIC is strictly less expressive than L^ω_μ. The method used to separate the logics is a generalisation of the definition of automaticity from string languages to classes of finite trees, closed under bisimulation. Automata that operate on trees have been widely studied in the literature (see, for instance, [7]). We consider a version of "bottom-up" automata that have the property that the class of trees accepted is necessarily closed under bisimulation. Formally, a bottom-up tree automaton is $\mathcal{A} = (Q, A, \delta, F, s)$, where $s \in Q$ is a start state, and $\delta : 2^{Q \times A} \to Q$. We say such an automaton accepts a tree $\mathcal{T}$, if there is a labelling $l : \mathcal{T} \to Q$ of the nodes of $\mathcal{T}$ such that:

- for every leaf v, $l(v) = s$;
- the root of $\mathcal{T}$ is labelled $q \in F$; and
- $l(v) = \delta(\{(l(w), a) : v \xrightarrow{a} w\})$.

We have, for simplicity, assumed that $\mathcal{T}$ is a transition system where the set of propositions $\mathcal{P}$ is empty. The automata are easily generalised to the case where such propositions are present. Indeed the labelling systems we introduce in Definition 16 below offer such a generalisation. The tree-automata as we have defined them are different from *ranked* tree automata as in [7]. There, the number of children a tree may have is fixed, and the state assigned by an automaton to a node depends on the *tuple* of states assigned to its children. As we are interested in properties invariant under bisimulation, the definition we use, where the state assigned to a node in a tree depends on the *set* of states assigned to its children, seems more natural. Indeed, it is not difficult to see that the class of trees accepted by such an automaton is closed under bisimulation.

For a bisimulation-closed class $\mathcal{C}$ of trees, its automaticity can be defined (see [4]) as the function mapping n to the smallest bottom-up tree automaton agreeing with $\mathcal{C}$ on all trees of *height* n. Height is the appropriate measure to use on a tree since it bounds the number of steps the automaton takes. This version of automaticity was used in particular to separate the expressive power of MIC from that of L^{ω}_{μ}. Indeed, one can establish the following facts about the automaticity of classes of trees definable in modal fixed-point logics.

Proposition 4 *(1) Every class of trees definable in L_μ has constant automaticity.*
(2) Every class of trees definable in MIC *has at most exponential automaticity.*
(3) There is a class of strings definable in MIC *that has exponential automaticity.*
(4) There is a class of trees definable in L^{ω}_{μ} that has non-elementary automaticity.

Statement (1) follows from the fact that for any formula φ of L_μ we can construct a bottom-up tree automaton which accepts exactly those trees that satisfy φ (see [15]). Statements (2), (3) and (4) are shown in [4]. However, (2) can also be derived as a special case of Theorem 27 proved below. The particular class of trees used to establish (4) is the *bisimulation problem*. This is the class of trees T such that for any two children of the root t_1 and t_2, we have $T, t_1 \sim T, t_2$. Its automaticity was shown in [4] to be non-elementary. It can be seen that the automaticity of this class is the maximum possible, in that any bisimulation-closed class of trees of height at most n can be recognised by a bottom-up automaton which has one state for each possible bisimulation class. Thus, the automaticity is bounded by the number of bisimulation equivalence classes.

These results on trees carry over *a fortiori* to classes of acyclic transitions systems.

3.1 Monadic Inflationary Fixed-Point Logic

We now look at the automaticity of the bisimulation-invariant fragment of monadic IFP on trees and show that there is no elementary lower bound for it. More particularly, we show that for any elementary function f there is a class of transition systems definable in monadic inflationary fixed-point logic whose labelling index dominates f. A consequence is that MIC is not the bisimulation invariant fragment of monadic IFP, something that might have been conjectured, just as the μ-calculus is the bisimulation-invariant fragment of monadic least fixed-point logic.

Monadic inflationary fixed-point logic (M-IFP) is the closure of first-order logic under a rule for forming fixed points of unary relations. If $\varphi(X, x)$ is a formula with a free set variable X and a free first-order variable x, then for any term t, $[\mathbf{ifp}_{X,x}\,\varphi](t)$ is also a formula. The semantics is defined as for MIC, i.e. $[\mathbf{ifp}_{X,x}\,\varphi](t)$ is true in a structure $\mathcal{K}$, if the interpretation of t is in the inflationary fixed point of the operator that maps X to $\{v : (\mathcal{K}, X, v) \models \varphi(x)\}$.

The classes of transition systems we are going to construct that are definable in M-IFP and have high automaticity are based on the use of trees to encode sets of integers in a number of ways of increasing complexity. To be precise, for each natural number k, we inductively define an equivalence relation $\simeq_k$ on trees as follows.

Definition 5 *For any two trees t and s, write $t \simeq_0 s$ just in case t and s have the same height and $t \simeq_{k+1} s$ just in case the set of $\simeq_k$-equivalence classes of the subtrees rooted at the children of the root of t is the same as the set of $\simeq_k$-equivalence classes of the subtrees rooted at the children of the root of s.*

By abuse of notation, we will also think of these relations as relations on the nodes of a tree $\mathcal{T}$. In this case, by $u \simeq_k v$ we mean $t_u \simeq t_v$ where t_u and t_v are the trees rooted at u and v respectively. A simple induction establishes the following lemma.

Lemma 6 *The number of distinct $\simeq_k$ equivalence classes of trees of height $n+k$ or less is k-fold exponential in n.*

Proof. The proof is by induction on k. Clearly, the number of $\simeq_0$ equivalence classes of trees of height n or less is n. Let N be the number of $\simeq_k$ equivalence classes of trees of height $n+k$ and $t_1, \ldots, t_N$ be a set of representatives for these classes. Then for every subset S of $\{1, \ldots, N\}$ we can form a tree of height $n+k+1$ such that the set of trees rooted at the children of the root is exactly $\{t_i : i \in S\}$. Such trees formed from distinct subsets are $\simeq_{k+1}$ inequivalent. Thus, there are 2^N $\simeq_{k+1}$-equivalence classes. $\square$

Let E_k be the function that maps n to the number of $\simeq_k$ equivalence classes of trees of height $n+k$ or less. Also, let $\mathcal{C}_k$ be the class of trees $\mathcal{T}, v$ with root v such that all successors of the root are $\simeq_k$-equivalent.

Lemma 7 *The automaticity of $\mathcal{C}_{k+1}$ is at least E_k.*

Proof. Suppose there were an automaton $\mathcal{A}$ with fewer than E_k states accepting $\mathcal{C}_{k+1}$. Then, by the definition of E_k, there are two $\simeq_k$-inequivalent trees t and s of height at most $n+k$ such that $\mathcal{A}$ reaches the same state at the root of t as it does at the root of s. Consider now the tree t_1 of height $n+k+1$ which consists of a root with two children that are the roots of two copies of t and the tree t_2 of height $n+k+1$ which consists of a root with two children one of which is the root of a copy of t and the other is a root of a copy of s. Clearly, $\mathcal{A}$ accepts t_1 if, and only if, it accepts t_2. However, $t_1 \in \mathcal{C}_{k+1}$ and $t_1 \notin \mathcal{C}_{k+1}$, so this is a contradiction. $\square$

By Lemma 7, the automaticity of $\mathcal{C}_k$ is at least k-fold exponential. By showing that $\simeq_k$-equivalence is M-IFP-definable, we establish the following theorem.

Theorem 8 *For every elementary function f, there is a property with automaticity $\Omega(f)$ definable in* M-IFP.

Proof. We first show by induction on k that the equivalence relation $\simeq_k$ is definable by a formula $\vartheta_k(u,v) \in$ M-IFP.

Basis. Let $\psi(x)$ be defined as

$$\begin{aligned}\psi(x;u,v) := [\mathbf{ifp}_{X,x}\ &((x \neq u \wedge x \neq v \wedge \forall y\,(Exy \rightarrow Xy)) \vee \\ &(x = u \wedge \forall y\,(Euy \rightarrow Xy) \wedge \exists y\,(Evy \wedge \neg Xy)) \vee \\ &(x = v \wedge \forall y\,(Evy \rightarrow Xy) \wedge \exists y\,(Euy \wedge \neg Xy)))](x).\end{aligned}$$

and define $\vartheta_0(u,v) := \forall x\,(x = u \lor x = v \to \neg\psi(x))$. Clearly, in each stage α, X^α contains all nodes of height less than α other than u and v and one of these just in case they are of different height one of which is less than α. Thus, ϑ_0 is true of a pair of nodes u, v just in case they are of the same height.

Induction Step. The definition of the relation given in Definition 5 actually shows that ϑ_{k+1} is obtained by a *first-order* formula from the relations $\simeq_k$.

By Lemma 6 there is a k such that the number of $\simeq_k$-equivalence classes is $\Omega(f)$. But then, $\mathcal{C}_{k+1}$ is a class of trees that is bisimulation-closed, it is M-IFP-definable, and its automaticity, by Lemma 7 is $\Omega(f)$. □

It follows from this that there are bisimulation invariant properties definable in M-IFP that are not definable in MIC. This contrasts with L_μ whose expressive power coincides precisely with the bisimulation invariant fragment of monadic LFP. This result dashes hopes of characterising MIC as the bisimulation-invariant fragment of a natural predicate logic, a question that was posed in [4].

Corollary 9 MIC *is strictly contained in the bisimulation invariant fragment of* M-IFP.

4 Labelling Index

We now generalise automaticity further to finite transition systems that are not necessarily acyclic. This necessitates some changes. First, we have to extend the automata model to devices operating on arbitrary finite transition systems. As the structures may have cycles, there is no natural start or end-point for an automaton. For this reason, we have refrained from calling the devices automata and adopted the term *labelling systems* instead. The systems are deterministic in that the label attached to a node is completely determined by the labels at its successors and the propositions that hold at the node. In this sense, the devices are also bottom-up. The formal definition is given in Definition 16.

However, in order to have a meaningful measure of the growth rate of these devices, we require a measure of the size of finite transitions systems that generalises the length of a string and the height of a tree. We proceed to this first.

Definition 10 *Let $\mathcal{K}$ be a transition system with universe V and let $v \in V$. The* rank of v in $\mathcal{K}$ *is the largest n such that there is a sequence of* distinct *nodes $v_1, \ldots, v_n$ in $\mathcal{K}$ with $v = v_1$ and a path from v_i to v_{i+1} for each i. The* rank of $\mathcal{K}$ *is the supremum of the rank of all nodes $v \in V$.*

Note that on a finite transition system $\mathcal{K}$ the rank of $\mathcal{K}$ is witnessed by a state v, i.e. there is a state v whose rank equals the rank of $\mathcal{K}$. To simplify notation, we will sometimes refer to the rank of a node v in $\mathcal{K}$ as the "rank of $\mathcal{K}, v$".

It is easy to see that the rank of a tree is indeed its height (taking the height of a tree with a single node as being 1) and the rank of any acyclic structure is equal to the length of the longest path. This observation can be further generalised by the following equivalent characterisation of rank.

Definition 11 *The* block decomposition *of a structure $\mathcal{K}$ is the acyclic graph $G = (V, E)$ whose nodes are the strongly connected components of $\mathcal{K}$ and $(s, t) \in E$ if, and only if, for some $u \in s$ and some $v \in t$, there is an action a such that $u \xrightarrow{a} v$. For each node s of G, we write $weight(s)$ for the number of nodes u of $\mathcal{K}$ such that $u \in s$. The* block rank *of a node s of G is defined inductively by*

$$rank(s) = weight(s) + \max\{rank(t) : (s, t) \in E\}.$$

The block rank *of a finite transition system $\mathcal{K}$ is defined as the supremum of the rank of all nodes v in $\mathcal{K}$.*

The following lemma is now immediate from the definitions.

Lemma 12 *For every transition system $\mathcal{K}$ and every node v, the rank of v in $\mathcal{K}$ is equal to its block rank.*

When relating tree-automata to fixed-point logics as in Proposition 4, the key property of the height of a tree is that it bounds the length of any simple fixed point induction that can be defined in L_μ or MIC. We show that this carries over to our definition of rank.

Let kMIC denote the class of formulas in MIC where every system of formulas defining a fixed-point induction has at most k rules. Note that we do allow arbitrary nesting of fixed-point operators inside a kMIC-formula. The *closure ordinal* of $\mathcal{K}$, in terms $\mathrm{cl}_{k\mathrm{MIC}}(\mathcal{K})$, is defined as the supremum of all α such that there is a system S of at most k formulas in kMIC for which α is the least ordinal with $S^\alpha = S^{\alpha+1}$. For a structure $\mathcal{K}, v$ with a distinguished node v, we write $\mathrm{cl}_{k\mathrm{MIC}}(\mathcal{K}, v)$ for the supremum of all α such that there is a system S of at most k formulas, defining variables $X_1, \ldots, X_k$, for which α is the least ordinal with $v \in X_1^\alpha$. We refer to $\mathrm{cl}_{k\mathrm{MIC}}(\mathcal{K}, v)$ as the *closure ordinal of v in $\mathcal{K}$*.

Lemma 13 *Let p be the rank of a node v in a structure $\mathcal{K}$. Then* $\mathrm{cl}_{k\mathrm{MIC}}(\mathcal{K}, v) \leq k \cdot p$.

Proof. Let S be a system of at most k formulas of kMIC defining variables $X_1, \ldots, X_k$ and let n is the least ordinal with $v \in X_1^n$. We construct a non-repeating sequence $u_1 \ldots u_m$ of states[3] in $\mathcal{K}$, where $m := \lceil \frac{n}{k} \rceil$, such that every u_{i+1} is reachable from u_i.

For this, consider the stages X_j^i induced by S, where $1 \leq j \leq k$ and $0 \leq i \leq n$. Clearly, for every stage $i > 0$ there must be a state u and a variable X_j such that $u \in X_j^i$ but $u \notin X_j^{i-1}$. We inductively define for every such state u and set X_j a dependency tree $\mathcal{T}(u, X_j)$ as follows. If $i = 1$, the tree consists of a single node labelled by u and X_j. If $i > 1$, then there must be an element w, reachable from u, and a set X_l such that $w \in X_l^{i-1} - X_l^{i-2}$. Otherwise, u would already be contained in X_j at some earlier stage. Now, the dependency tree for u and X_j consists of a node labelled by u and X_j and for each such w a successor which is the root of the dependency tree $\mathcal{T}(w, X_l)$ for w and X_l. A simple induction shows that the dependency tree for every state u and set X_j such that $u \in X_l^i - X_l^{i-1}$ is of height exactly $i-1$.

Now consider the dependency tree $\mathcal{T} := \mathcal{T}(v, X_1)$ for the distinguished state v and the set X_1. By definition of the closure ordinal of v in $\mathcal{K}$, $v \in X_1^n - X_1^{n-1}$. Clearly, no path in $\mathcal{T}$ can contain more than one node labelled by the same pair (w, X_l), for some w and X_l. Therefore, for every state w in $\mathcal{K}$ and every path $\mathcal{P}$ in $\mathcal{T}$, there are at most k nodes labelled by (w, X_l) for some l. Let $\mathcal{P} \subseteq \mathcal{T}$ be a path in $\mathcal{T}$ of maximal length, i.e. length n. As $\mathcal{P}$ cannot contain more than k nodes labelled by the same state w in $\mathcal{K}$, it follows that there are at least $m := \lceil \frac{n}{k} \rceil$ nodes which are labelled by different states from $\mathcal{K}$. Let $(u_1, \ldots, u_m)$ be the sequence of these states in decreasing order with respect to their height in $\mathcal{T}$. By construction of the dependency tree, every u_{i+1} is reachable from u_i. Thus, we have constructed a non-repeating sequence of length m where every node is reachable from its predecessor.

By the definition of rank, m must be less than or equal to p. As $n \leq k \cdot \lceil \frac{n}{k} \rceil = k \cdot m \leq k \cdot p$, the lemma is proved.

□

An immediate consequence of the lemma is the following corollary.

Corollary 14 *For every transition system $\mathcal{K}$, $\mathrm{cl}_{k\mathrm{MIC}}(\mathcal{K}) \leq k \cdot p$, where p is the rank of $\mathcal{K}$.*

In particular, taking $k = 1$, we have that closure ordinals of simple inductions are bounded by the rank of the structure. While the rank of a structure $\mathcal{K}$

[3] In this proof we refer to the nodes in $\mathcal{K}$ as *states* to distinguish them from the nodes of a tree that will be defined later on.

provides a combinatorial measure that bounds the closure ordinals of simple inductions, it is not an exact characterisation. Nor can we expect it to be exact because it is clear that the closure ordinals are invariant under bisimulation while rank is not. It may be more appropriate therefore to consider the rank, not of a structure $\mathcal{K}$, but of its quotient under bisimulation $\mathcal{K}_{/\sim}$. With this, we do indeed get a converse to Lemma 13, as seen below. Note that for the case of finite strings and trees this consideration was unnecessary. In the case of finite strings, no two elements can be bisimulation equivalent. In the case of finite trees, the rank is not changed by taking a quotient under bismulation: it is still equal to the height of the tree.

Lemma 15 *If the rank of $\mathcal{K}_{/\sim}$ is n, there is a formula $\varphi(X) \in \mathrm{ML}$, positive in X, whose closure ordinal on $\mathcal{K}$ is n.*

Proof. Since closure ordinals of modal formulas are invariant under bisimulation, it suffices to construct a formula φ whose closure ordinal in $\mathcal{K}_{/\sim}$ is n. Let $v_1, \ldots, v_n$ be a sequence of elements in $\mathcal{K}_{/\sim}$ witnessing its rank. Since, by construction of $\mathcal{K}_{/\sim}$, all nodes have distinct bisimulation types, we can write modal formulas $\varphi_1, \ldots, \varphi_n$ such that v_i is the unique element of $\mathcal{K}_{/\sim}$ satisfying φ_i. Moreover, for each i, consider the sequence of actions that leads from v_i to v_{i+1} and let p_i denote the corresponding pattern of modalities. For example, if v_2 is the a-successor of a b-succesor of v_1, then $p_1 = \langle b\rangle\langle a\rangle$.

Now, the required formula $\varphi(X)$ is given by:

$$\varphi(X) \equiv \varphi_n \vee \bigvee_{i<n} (\varphi_i \wedge p_i(\varphi_{i+1} \wedge X)).$$

$\square$

An immediate consequence of Lemmas 13 and 15 is that the maximal closure ordinals of simple MIC and simple L_μ formulas on any structure are the same.

We are now ready to introduce labelling systems, which generalise bottom-up tree automata to transition systems that are not necessarily acyclic.

Definition 16 *A* labelling system *$\mathcal{L}$ is a quintuple $\mathcal{L} := (Q, \mathcal{A}, \mathcal{P}, \delta, \mathcal{F})$, where Q is a finite set of labels, $\mathcal{A}$ a finite set of actions, $\mathcal{P}$ a finite set of proposition symbols, $\mathcal{F} \subseteq Q$ a set of* accepting labels, *and δ a total function $\delta : 2^{Q\times\mathcal{A}} \times 2^{\mathcal{P}} \to Q$, the* transition function.

For every Kripke-structure $\mathcal{K} := (V, (E_a)_{a\in\mathcal{A}}, (P_i)_{i\in\mathcal{P}})$ and node $v \in V$, the labelling system $\mathcal{L}$ accepts $\mathcal{K}, v$, denoted $\mathcal{K}, v \models \mathcal{L}$, if, and only if, there is a function $f : V \to Q$ such that $f(v) \in \mathcal{F}$ and for each $s \in V$,

$$f(s) = \delta\Big(\Big\{(f(s'), a) : a \in \mathcal{A} \textit{ and } (s, s') \in E_a\Big\}, \Big\{i : i \in \mathcal{P} \textit{ and } s \in P_i\Big\}\Big).$$

In words, a rooted structure $\mathcal{K}, v$ is accepted by a labelling system if there is some way of labelling the elements of $\mathcal{K}$ which is consistent with the local conditions specified by δ and which labels the root v by a label from $\mathcal{F}$. Moreover, the local conditions δ are such that the label of a node is completely determined by the set of labels assigned to its successors and the atomic propositions true at the node. This is a straightforward generalisation of bottom-up automata on trees but we have refrained from calling the systems automata as they have no specific starting point in the structure.

As δ is functional, labelling systems have some characteristics of deterministic devices, coinciding with bottom-up automata on trees. On the other hand, if the structures may contain cycles, some form of nondeterminism is present as acceptance is defined in terms of the *existence* of a labelling. Thus, for a given structure and a given labelling system, there may be more than one labelling function f witnessing the fact that $\mathcal{L}$ accepts $\mathcal{K}, v$.

The class of structures accepted by a labelling system is not necessarily closed under bisimulation. This can be seen in the following simple example.

Example 17 *Consider the labelling system $\mathcal{L} = (Q, \mathcal{A}, \mathcal{P}, \delta, \mathcal{F})$ given by*

- $Q = \{q_{\text{even}}, q_{\text{odd}}\}$,
- $\mathcal{A} = \{a\}$ *and* $\mathcal{P} = \emptyset$,
- $\mathcal{F} = \{q_{\text{even}}\}$ *and*
- *the function δ is given by the rules* $\delta(\emptyset) = q_{\text{even}}$, $\delta(\{(q_{\text{even}}, a)\}) = q_{\text{odd}}$, $\delta(\{q_{\text{odd}}, a\}) = q_{\text{even}}$ *and* $\delta(\{(q_{\text{even}}, a), (q_{\text{odd}}, a)\}) = q_{\text{even}}$.

Here we have dropped the second argument of δ as it is always $\emptyset$.

This labelling system accepts a transition system that only consists of a single cycle if, and only if, this cycle is of even length.

As we are especially interested in labelling systems that define bisimulation-closed classes of structures, we consider the following definition.

Definition 18 *A labelling system $\mathcal{L}$ is* $\sim$-consistent, *if for all Kripke-structures $\mathcal{K}, v$, whenever $\mathcal{K}, v \models \mathcal{L}$ then there is a labelling f witnessing this with the property that for all s, s', $\mathcal{K}, s \sim \mathcal{K}, s'$ implies $f(s) = f(s')$.*

It might seem that a more natural condition would be obtained just by requiring the class of structures defined by $\mathcal{L}$ to be closed under bisimulation, as in the following definition.

Definition 19 *A labelling system $\mathcal{L}$ is* $\sim$-invariant *if, whenever $\mathcal{K}, v \models \mathcal{L}$ and $\mathcal{K}, v \sim \mathcal{K}', v'$ then $\mathcal{K}', v' \models \mathcal{L}$.*

As it happens, these two definitions are equivalent for the structures that are of interest to us. Call $\mathcal{K}, v$ *rooted* if, for every node $u \neq v$, there is a path from v to u in $\mathcal{K}$.

Lemma 20 *On rooted structures, a labelling system is $\sim$-consistent if, and only if, it is $\sim$-invariant.*

Proof. Suppose $\mathcal{L} = (Q, A, P, \delta, \mathcal{F})$ is $\sim$-consistent and $\mathcal{K}, v \models \mathcal{L}$, with $f : V \to Q$ being a labelling that witnesses this, such that $f(s) = f(s')$ whenever $\mathcal{K}, s \sim \mathcal{K}, s'$. Consider a structure $\mathcal{K}'$ with universe V' such that $\mathcal{K}', v' \sim \mathcal{K}, v$. Define a labelling $f' : V' \to Q$ by $f'(u) = f(s)$ where $s \in Q$ is such that $\mathcal{K}', u \sim \mathcal{K}, s$. Such an s must exist since both $\mathcal{K}$ and $\mathcal{K}'$ are rooted structures and the roots v and v' are bisimulation equivalent. Thus, for any u reachable from v' there is an s reachable from v which is bisimulation equivalent to u. Moreover, the labelling f' is well defined as the labelling f is $\sim$-consistent and there is therefore only one label for each $\sim$-equivalence class in $\mathcal{K}$. It is easily verified that f' is then a labelling witnessing that $\mathcal{K}', v' \models \mathcal{L}$.

For the converse, suppose $\mathcal{L} = (Q, A, P, \delta, \mathcal{F})$ is $\sim$-invariant and $\mathcal{K}, v \models \mathcal{L}$. Consider the structure $\mathcal{K}_{/\sim}$ which is the quotient of $\mathcal{K}$ under $\sim$. Since $\mathcal{K}_{/\sim}, [v] \sim \mathcal{K}, v$, we have, by the $\sim$-invariance of $\mathcal{L}$ that $\mathcal{K}_{/\sim}, [v] \models \mathcal{L}$. Let f be a labelling that witnesses this latter fact. Define $f' : V \to Q$ by $f'(s) = f([s])$. Then f' is a valid labelling of $\mathcal{K}, v$, and it satisfies the condition that $\mathcal{K}, s \sim \mathcal{K}, s'$ implies $f'(s) = f'(s')$. $\square$

While any $\sim$-consistent labelling system $\mathcal{L}$ defines a class of $\sim$-invariant structures, not every bisimulation-closed class $\mathcal{C}$ of structures is given by such a labelling system. However, we see below how $\mathcal{C}$ can be defined by a family of systems. In order to define the family we use the rank of a structure as a measure of its size.

We begin with two simple observations. First, note that the rank of any structure $\mathcal{K}$ is bounded by its size. Secondly, we note the following simple lemma.

Lemma 21 *For any n, there are, up to bisimulation, only finitely many structures of rank n.*

Proof. Let $\mathcal{K}, v$ be a structure of rank n and let G be its block decomposition, as defined in Definition 11. We prove by induction on the height of G (as G is acyclic, this is well-defined) that there are only finitely many structures, up to bisimulation, with rank n and whose block decomposition has height h. Since h is at most n, this establishes the lemma.

If G has height 1, then $\mathcal{K}$ is strongly connected. Thus, its rank is equal to its size. As there are only finitely many structures, up to isomorphism (and, a fortiori, up to bisimulation), of size n, we are done. For the induction step, suppose G has height $h + 1$, let k be the weight of the block in G containing v and let $\mathcal{K}'$ be the substructure of $\mathcal{K}$ induced by the k elements in this block. Note that $k < n$. Every element of $\mathcal{K}$ that is not in $\mathcal{K}'$ can be seen as the root of a structure of rank at most $n - k$ whose block decomposition has height at most h. By induction hypothesis, there are only finitely many bisimulation equivalence classes among such structures. Let $\mathcal{T}$ be the collection of such classes. The bisimulation type of $\mathcal{K}$ is completely determined by the bisimulation type of $\mathcal{K}'$ when each element v is additionally coloured with the set of pairs (a, T) where $T \subseteq \mathcal{T}$ is the set of types of elements not in $\mathcal{K}'$ accessible from v by action a. As the size of $\mathcal{K}'$ is bounded, and there are only finitely many possible colours, we conclude that the number of bisimulation types with height $h + 1$ is bounded. $\square$

We show now that every bisimulation closed class of transition systems can be accepted by a family of labelling systems as follows.

Lemma 22 *Let $\mathcal{C}$ be a bisimulation closed class of finite structures. For each n there is a $\sim$-consistent labelling system $\mathcal{L}_n$ such that for any structure $\mathcal{K}$ with rank($\mathcal{K}$) $\leq n$, $\mathcal{L}_n$ accepts $\mathcal{K}$ if, and only if, $\mathcal{K} \in \mathcal{C}$.*

Proof. By Lemma 21, there are, for fixed n, only finitely many bisimulation equivalence classes. We define a labelling system $\mathcal{L} := (Q, \mathcal{A}, \mathcal{P}, \delta, \mathcal{F})$ by taking Q to be the collection of all bisimulation classes of structures of rank at most n and defining, for sets $M \subseteq Q \times \mathcal{A}$ and $P \subseteq \mathcal{P}$, $\delta(M, P)$ to be the bisimulation class of the structure consisting of a root labelled by the propositions in P with, for each $(q, a) \in M$, an a successor of the root such that the structure rooted at this successor is in the bisimulation class represented by q. Define $\mathcal{F}$ as the set of states representing the bisimulation class of a structure in $\mathcal{C}$. The lemma now follows immediately. $\square$

Considering a family $(\mathcal{L}_n)_{n<\omega}$ of labelling systems $\mathcal{L}_n$ such as in Lemma 22, the minimal size in terms of n of the labelling systems $\mathcal{L}_n$ in the family can be seen as a measure of the complexity of the class $\mathcal{C}$. This leads to the definition of the labelling index of classes of transition systems, which generalises the automaticity of languages and classes of trees.

Definition 23 *Let $\mathcal{C}$ be a bisimulation closed class of finite structures. The* labelling index *of $\mathcal{C}$ is defined as the function $f : n \mapsto |\mathcal{L}_n|$ mapping natural numbers n to the number of labels of a minimal labelling system $\mathcal{L}_n$ such that for any $\mathcal{K}, v$ of rank n or less, $(\mathcal{K}, v) \in \mathcal{C}$ if, and only if, $\mathcal{K}, v \models \mathcal{L}_n$.*

Automata models on graphs, generalising automata on trees, have been studied before. A common way in which such automata are defined is in terms of tiling systems, see [13,14]. A tiling system consists of a set of states Q and a set of finite graphs - called tiles - labelled by states from Q. A graph $\mathfrak{G}$ is accepted by a tiling system if it can be labelled by states from Q in such a way that the resulting labelled graph can be completely covered by overlapping tiles.

Superficially, tiling and labelling systems seem closely related and indeed we could rephrase the definitions above in terms of tiling systems. But there are subtle differences. One is that there is no preferred direction to a tiling system. A tile can constrain the label of a node both in terms of its predecessors as well as its successors. In contrast, in our labelling systems the labels only "look forwards" which is consistent with our intent to capture bisimulation-invariant properties. Additionally, tiling systems are inherently nondeterministic, permitting multiple possible labels for a node in a graph with the same set of labels on its neighbours. In our labelling systems the label of a node is determined by the set of labels on its successors. Finally, there is no notion of an accepting state in a tiling system

5 Labelling Indices of Modal Logics

In this section, we aim to establish upper and lower bounds on the labelling index of classes of structures definable in modal logics such as ML and its various fixed-point extensions.

5.1 The modal iteration calculus

It was shown by Dawar, Grädel, and Kreutzer in [4] that any class of *trees* definable in MIC has at most exponential automaticity. The proof translates easily to the labelling index on arbitrary structures. For this, we first need some notation.

If φ is a MIC-formula we write $\mathrm{sf}(\varphi)$ for the set of sub-formulas of φ. For the rest of this section we agree w.l.o.g. on the following naming convention: No fixed-point variable is bound twice in φ. Further, let $X_1, \ldots, X_k$ be an enumeration of all fixed-point variables occurring in φ ordered from the outside in, i.e. if $i < j$ then either X_j does not occur free in the formula defining X_i in φ or X_i and X_j are bound together in a simultaneous induction.

Let $\bar{\imath} := i_1, \ldots, i_k$ be a tuple of finite ordinals. If $\psi \in \mathrm{sf}(\varphi)$ is a sub-formula of φ and $\mathcal{K}, v$ is a transition system, we write $(\mathcal{K}, \overline{X}^{\bar{\imath}}), v \models \psi$ to indicate that ψ holds at the node v if each free fixed-point variable X_j in ψ is interpreted by the stage $X_j^{i_j}$ of the induction on $\mathcal{K}$.

Definition 24 (φ-types) *A* φ-type of rank n *of a formula* φ *is a function* $f : \{0, \dots, k \cdot n\}^k \to 2^{\mathrm{sf}(\varphi)}$ *such that there is a transition system* $\mathcal{K}, v$ *of rank* n *with*

$$f(\bar{\imath}) := \Big\{\psi : (\mathcal{K}, \overline{X}^{\bar{\imath}}), v \models \psi \text{ and if } X_j \text{ is bound in } \psi \text{ then } i_j = k \cdot n\Big\}.$$

In this case we also say that f *is the* φ-type of this transition system $\mathcal{K}$.

The set of *rank* n MIC-*formula types* is defined as the set of all functions f which are a φ-type of rank n, for some formula φ and some n.

We use the notion of MIC-formula types to define for each formula $\varphi \in$ MIC a family of labelling systems accepting precisely those structures which satisfy φ. It is easily seen that in any transition system of rank n the φ-type of rank n of a node v is uniquely determined by the atomic propositions true at v and the φ-types of the successors of v. (A proof of this fact is implicit in the proof of the next lemma.) This motivates the following definition.

Definition 25 *Let* φ *be a formula in* MIC. *For every* $n \in \omega$ *define the labelling system* $\mathcal{L}_\varphi(n) := (Q, \mathcal{A}, \mathcal{P}, \delta, \mathcal{F})$ *as follows.*

- $Q := \{f : f$ *is a* φ*-type of rank at most* $n\} \,\dot\cup\, \{\bot\}$.
- *Let sets* $M \subseteq Q \times \mathcal{A}$ *and* $P \subseteq \mathcal{P}$ *be given. If for some* a, $(\bot, a) \in M$ *then* $\delta(M, P) = \bot$. *If there is a transition system* $\mathcal{K}, v$ *of rank at most* n *such that*
  - *(i) at* v*, exactly the propositions in* P *are true,*
  - *(ii) for each pair* $(q', a) \in M$ *there is an* a*-successor of* v *whose* φ*-type is* q'*, and*
  - *(iii)* v *has no further successors*

  then define $\delta(M, P)$ *as the* φ*-type of* v *in* $\mathcal{K}$*. Otherwise,* $\delta(M, P) := \bot$.
- *Finally,* $\mathcal{F} \subseteq Q$ *is the set of labels* $q \neq \bot$ *such that* $\varphi \in q(\overline{k \cdot n})$.

Lemma 26 *Let* $\varphi \in$ MIC *be a formula,* $\mathcal{K}$ *be a structure and* v *be a node of rank* n *in* $\mathcal{K}$*. Then for every* $m \geq n$*,* $\mathcal{L}_\varphi(m)$ *accepts* $\mathcal{K}, v$ *if, and only if,* $\mathcal{K}, v \models \varphi$.

Proof. Recall the notational convention outlined above. In particular, let $X_1, \dots, X_k$ be the fixed-point variables defined in φ enumerated as described above. Let $\mathcal{K}$ be a transition system and v be a node of rank n. Let $m \geq n$. By Lemma 13, every induction defined by a sub-formula of φ must close on v in $\mathcal{K}$ after at most kn steps.

Suppose first that $\mathcal{K}, v \models \varphi$. We construct a labelling f witnessing that $\mathcal{L} := \mathcal{L}_\varphi(m)$ accepts $\mathcal{K}, v$. For every node $u \in \mathcal{K}$ reachable from v let $f(u)$ be the φ-type of $\mathcal{K}, u$. It is an immediate consequence of the definition of φ-types and the labelling systems $\mathcal{L}_\varphi(m)$ that this labelling is consistent with the transition function, i.e. it is indeed a labelling. As $\mathcal{K}, v \models \varphi$, it follows that $\varphi \in f(v)(\overline{k \cdot n})$ and thus $\mathcal{L}$ accepts $\mathcal{K}, v$.

Towards the converse, let f be a labelling witnessing that $\mathcal{K}, v \models \mathcal{L}$. It follows immediately from the definition of $\mathcal{L}$ that no node reachable from v can be labelled by $\perp$. (Otherwise, v would be labelled by $\perp$ contradicting the assumption that $\mathcal{L}$ accepts $\mathcal{K}, v$.) Hence all nodes reachable from v are labelled by a φ-type of rank m.

Let V be the universe of $\mathcal{K}$. We claim that for all $u \in V$, if $q := f(u)$ is the φ-type f assigns to u, then for all $\bar{\imath} \in \{0, \ldots, k \cdot m\}^k$ and all $\psi \in \mathrm{sf}(\varphi)$, $\psi \in q(\bar{\imath})$ if, and only if, $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models \psi$.

The claim is proved simultaneously for all nodes by simultaneous induction on the lexicographical ordering on $\bar{\imath}$ and the structure of the formulas $\psi \in q(\bar{\imath})$. Let $u \in V$, ψ and $\bar{\imath}$ be given and assume that the claim has already been proved for all triples $w \in V$, $\psi' \in \mathrm{sf}(\psi)$, and $\bar{\imath}' \leq \bar{\imath}$ so that $\psi' \neq \psi$ or $\bar{\imath}' \neq \bar{\imath}$. Here, and below, we write $\bar{\imath}' \leq \bar{\imath}$ for the the partial order obtained as the product of the orders on individual elements.

- If $\psi := p \in \mathcal{P}$ is an atomic proposition, then the claim follows immediately from Part (i) of Definition 25.
- The case of Boolean connectives is trivial.
- Now suppose $\psi := \langle a \rangle \vartheta$. If u is a leaf, then ψ is clearly false at u. Let $q := f(u)$ be the φ-type f assigns to u. As $q \neq \perp$, the definition of the transition function implies that q is the φ-type of a transition system $\mathcal{S}, v'$, such that v' has no successor. It follows that $\psi \notin q(\bar{\imath})$.

  Otherwise, if u is not a leaf, let $M := \{(q', a) \in Q \times \mathcal{A} :$ there is an a-successor of u labelled by $q'\}$. By definition, q is the φ-type of some transition system $\mathcal{S}, s$ such that for each $(q', a) \in M$, there is an a successor of s in $\mathcal{S}$ whose φ-type is q'.

  If $\psi \in q(\bar{\imath})$, then $(\mathcal{S}, \overline{X}^{\bar{\imath}}), s \models \psi$ and hence there must be an a-successor s' of s in $\mathcal{S}$ such that $(\mathcal{S}, \overline{X}^{\bar{\imath}}), s' \models \vartheta$. It follows, that there exists a pair $(q', a) \in M$ such that $\vartheta \in q'(\bar{\imath})$. Let u' be an a-successor of u in $\mathcal{K}$ whose label is q'. By induction on the structure of the formulas, this implies that $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u' \models \vartheta$ and thus $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models \psi$.

  Conversely, if $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models \psi$, then there is some a-successor u' of u such that $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u' \models \vartheta$. Let q' be the label of u'. By induction on the structure of the formulas, this implies that $\vartheta \in q'(\bar{\imath})$ and therefore $\psi \in q(\bar{\imath})$.
- The case for $\psi := [a]\vartheta$ is analogous.
- Now, let $\psi := X_j$ be an atom where X_j is one of the fixed-point variables occurring in φ. Suppose X_j is bound in a system S that also binds $X_{r_1}, \ldots, X_{r_l}$.

  If $\bar{\imath} = \bar{0}$, then, since $X_j^0 = \emptyset$ for all j, $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \not\models X_j$ and also, as q is a

φ-type of some transition system, $X_j \notin q(\bar{\imath})$.

If $\bar{\imath} > \overline{0}$ then, since q is the φ-type of some transition system, this implies that there is a tuple $\bar{\imath}' < \bar{\imath}$ which agrees with $\bar{\imath}$ on all positions except $j, r_1, \ldots, r_l$, such that $\varphi_j \in q(\bar{\imath}')$, where φ_j is the defining formula of X_j. By induction on the stages, this implies that $(\mathcal{K}, \overline{X}^{\bar{\imath}'}), u \models \varphi_j$ and therefore $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models X_j$ by the inflationary semantics of the **ifp** -operator.

The converse is analogous.

- Finally, suppose $\psi := (\mathbf{ifp} \quad X_{r_1} : S)$, where

$$S := \begin{cases} X_{r_1} \leftarrow \varphi_{r_1}(X_{r_1}, \ldots, X_{r_l}) \\ \qquad \vdots \\ X_{r_l} \leftarrow \varphi_{r_l}(X_{r_1}, \ldots, X_{r_l}) \end{cases}$$

is a system of formulas. Suppose that $\psi \in q(\bar{\imath})$. W.l.o.g. assume that $r_1 < r_2 < \cdots < r_l$. By the definition of φ-types, for all $1 \leq j \leq l$, $i_{r_j} = km$, as the variables $X_{r_1}, \ldots, X_{r_l}$ are bound by ψ. Further, q is the φ-type of some transition system and therefore there is a sequence of stages $\overline{X}^{\bar{\imath}'}$ such that
· $i_j = i'_j$, for all $j \neq r_s$, $1 \leq s \leq l$,
· $i'_{r_s} < i_{r_s}$ for all $1 \leq s \leq l$, and
· $\varphi_{r_1} \in q(\bar{\imath}')$.
Thus $\bar{\imath}' < \bar{\imath}$ and, by induction on the order $\preceq$, we get that $(\mathcal{K}, \overline{X}^{\bar{\imath}'}), u \models \varphi_{r_1}$ and therefore $(\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models \psi$.

Again, the converse is analogous.

This concludes the induction. We have shown that for all nodes u labelled by $f(u) = q$, all $\psi \in \mathrm{sf}(\varphi)$, and all $\bar{\imath} \in \{0, \ldots, k \cdot m\}^k$,

$$\psi \in q(\bar{\imath}) \text{ if, and only if, } (\mathcal{K}, \overline{X}^{\bar{\imath}}), u \models \psi.$$

Since $\mathcal{L}$ accepts $\mathcal{K}, v$ we have that $\varphi \in f(v)(\overline{k \cdot n})$ and therefore $\mathcal{K}, v \models \varphi$. This finishes the proof of the lemma. □

The following theorem follows immediately.

Theorem 27 *Every* MIC *definable class of transition systems has at most exponential labelling index and there are classes of structures definable in* MIC *which have an exponential lower bound on their labelling index.*

Proof. In Lemma 26 we have proved that the family of labelling systems in Definition 25 defined for a formula φ accepts the class of all transition systems satisfying φ. Clearly, for every n, the size of $\mathcal{L}_\varphi(n)$ is exponential in n. From this, the upper bound follows immediately.

The lower bound follows from the fact that automaticity and labelling index coincide on classes of words. As shown in [4], there are languages definable in MIC which have an exponential automaticity. □

Another corollary of the results refers to modal logic. As ML-formulas can be seen as MIC-formulas without any fixed-point operators, the number of φ-types for a ML-formula φ depends only on φ and is therefore constant. Thus we immediately get the following.

Corollary 28 *Every property definable in* ML *has constant labelling index.*

Note that the converse fails, as, for instance, not all regular languages are definable in ML.

5.2 The modal μ-calculus

The main difference between the argument for MIC considered above and L_μ is monotonicity. This has a major impact on the definition of labelling systems accepting L_μ definable classes of structures. Consider the labelling systems as defined for MIC-formulas φ. The label assigned to any node u of the structures records, for every sub-formula of φ, all tuples $(i_1, \ldots, i_k)$ of induction stages where the sub-formula becomes true at u. As L_μ formulas are monotone, if a sub-formula is true at a tuple of stages $(i_1, \ldots, i_k)$ it will also be true at all stages obtained by increasing indices corresponding to μ operators and decreasing indices for ν-operators. Thus, it seems it should suffice to record, for each μ operator, the least stage at which the corresponding sub-formula becomes true at u and similarly the greatest stage for a ν operator. For instance, if we only had one fixed-point operator, say μX, it would suffice to mark each node u of the structure by the number of the stage at which it is included into the fixed point of X or to indicate that it was not included at all. We would thus only have a linearly bounded number of labels in the labelling system.

The situation with nested fixed-point operators is somewhat more complicated as the stage at which u enters the fixed-point of an induction may depend on what values are assigned to the free variables, i.e. at what stages the outer induction variables are being interpreted. But monotonicity also helps if there is more than one fixed-point operator. To see this, consider a formula defining a least fixed point. If the formula is true at a node u and a tuple of stages $\bar{\imath}$, then it is also true at u if all or some of its free fixed-point variables are interpreted by their fixed points. With this, it turns out to be sufficient to consider in each node u of the transition system only those tuples $\bar{\imath}$ of stages where at most one fixed-point induction has not reached its fixed point. As there are only polynomially many such tuples we get a polynomial upper bound on the size of the labelling systems. And, as we shall see, for greatest fixed-points it is not even necessary to record the stage at all as a node u is in the greatest fixed point if, and only if, it is in *some* fixed point. We now give a detailed proof of the following theorem, implementing the ideas presented above.

Theorem 29 *For every formula φ of L_μ, there is a polynomial p such that the property defined by φ has labelling index $O(p)$.*

Notation. For the rest of this section we fix a formula $\varphi \in L_\mu$ in guarded normal form (see Definition 2). Let $\mathrm{sf}(\varphi)$ be the set of sub-formulas of φ. Further, let $X_1, \ldots, X_k$ be the fixed-point variables occurring in φ and let, for $1 \leq i \leq k$, φ_i be the formula defining X_i, i.e. the variable X_i is bound by a fixed-point operator $\lambda_i X_i.\varphi_i$ with $\lambda_i \in \{\mu, \nu\}$. We call variables bound by a μ-operator *μ-variables* and variables bound by a ν-operator *ν-variables.*

We also fix a set $\mathcal{A}$ of actions and a set $\mathcal{P}$ of proposition symbols for the Kripke-structures we consider in this section. Let $\mathcal{K}$ be such a Kripke-structure. We write X_i^∞ for the fixed point of X_i on $\mathcal{K}$. To be precise, $X_1^\infty := \lambda_1 X_1.\varphi_1$ and $X_i^\infty := \lambda_i X_i.\varphi_i(X_1^\infty, \ldots, X_{i-1}^\infty)$ (recall that fixed-point variables are numbered from the outside in). Similarly, we write X_i^n for $\varphi_i^n(X_1^\infty, \ldots, X_{i-1}^\infty)$, i.e. the n^{th} stage of the fixed-point induction on $\varphi_i(X_1^\infty, \ldots, X_{i-1}^\infty)$. Whenever we use this notation it will be clear from the context which Kripke-structure we mean.

To define the set of labels of our labelling system, we need an analogue of the φ-types of Definition 24. However, as we are aiming at a polynomial size labelling system, the φ-types are too rich a set of labels. Instead, the information we seek to encode in a label is a function that maps formulas in $\mathrm{sf}(\varphi)$ to tuples of integers which list the least stages of the μ-inductions at which the particular formula becomes true. This is formalised in the next two definitions, which are somewhat more involved than the definition of φ-types.

Definition 30 *Let $P \subseteq \mathcal{P}$ be a set of propositions. A function $q : \Phi \to \{0, \ldots, n{+}1, \bot\}^k$ is* locally P-consistent*, if the following conditions hold for all $\psi \in \Phi$. We write $\big(q(\psi)\big)_i$ to denote the i^{th} component of the image of a formula ψ. To simplify notation, we agree that* $\min\{\bot, i\} := i$ *and* $\max\{\bot, i\} := \bot$ *for all $i \leq n+1$.*

- *If $\psi = p$, then $q(\psi) := (0, \ldots, 0)$ if $p \in P$ and $q(\psi) := (\bot, \ldots, \bot)$ otherwise.*
- *If $\psi = \neg p$, then $q(\psi) := (\bot, \ldots, \bot)$ if $p \in P$ and $q(\psi) := (0, \ldots, 0)$ otherwise.*
- *For all i, $\big(q(\psi_1 \vee \psi_2)\big)_i := \min\{\big(q(\psi_1)\big)_i, \big(q(\psi_2)\big)_i\}$.*
- *For all i, $\big(q(\psi_1 \wedge \psi_2)\big)_i := \max\{\big(q(\psi_1)\big)_i, \big(q(\psi_2)\big)_i\}$.*

- *If $\psi := \lambda_i X_i.\varphi_i$, with $\lambda_i \in \{\mu, \nu\}$, then for all j,*

$$\Big(q(\psi)\Big)_j := \begin{cases} n+1 & \text{if } j = i, \lambda_i = \mu \text{ and } \Big(q(\varphi_i)\Big)_i \neq \bot \\ 0 & \text{if } j = i, \lambda_i = \nu \text{ and } \Big(q(\varphi_i)\Big)_i = 0 \\ \Big(q(\varphi_i)\Big)_j & \text{if } j < i \\ \bot & \text{otherwise.} \end{cases}$$

- *If $\psi := X_i$ and X_i is a variable bound by a μ-operator, then for all j,*

$$\Big(q(X_i)\Big)_j := \begin{cases} \Big(q(\varphi_i)\Big)_j + 1 & \text{if } j = i \text{ and } \Big(q(\varphi_i)\Big)_i \neq \bot \\ 0 & \text{if } j > i \text{ and } \Big(q(\varphi_i)\Big)_i \neq \bot \\ \bot & \text{otherwise.} \end{cases}$$

- *If $\psi := X_i$ and X_i is a variable bound by a ν-operator, then for all j,*

$$\Big(q(X_i)\Big)_j := \begin{cases} 0 & \text{if } j \geq i \text{ and } \Big(q(\varphi_i)\Big)_i \neq \bot \\ \bot & \text{otherwise.} \end{cases}$$

A function q is locally consistent *if it is locally P-consistent for some $P \subseteq \mathcal{P}$.*

The previous definition takes care of consistency of the truth assignments at an individual node in the transition system. The following definition extends this to ensure consistency with the actions.

Definition 31 *Let Q be the set of all locally consistent functions $q : \mathrm{sf}(\varphi) \to \{0, \ldots, n+1, \bot\}^k$ and let $M \subseteq Q \times \mathcal{A}$. A function $q \in Q$ is* globally (M, P)-consistent, *if it is locally P-consistent and in addition the following conditions hold:*

- *If $\psi := \langle a \rangle \psi'$, then for all i, $\Big(q(\psi)\Big)_i := \min(\{\bot\} \cup \{\Big(q'(\psi')\Big)_i : (q', a) \in M\})$.*
- *If $\psi := [a]\psi'$, then for all i, $\Big(q(\psi)\Big)_i := \max(\{0\} \cup \{\Big(q'(\psi')\Big)_i : (q', a) \in M\})$.*

The function is globally consistent, if it is globally (M, P)-consistent for some pair (M, P).

We first prove some facts about globally-consistent functions.

Proposition 32 *(i) For each pair $(M, P) \subseteq \Big(Q \times \mathcal{A}\Big) \times \mathcal{P}$ as in Definition 31, there is exactly one function that is globally (M, P)-consistent.*
(ii) For all i such that X_i is a ν-variable, for all formulas $\psi \in \mathrm{sf}(\varphi)$, and all globally consistent functions q, either $\Big(q(\psi)\Big)_i = 0$ or $\Big(q(\psi)\Big)_i = \bot$.

Proof. For Part (i), recall that the formula φ is in guarded normal form. Thus, there is no circularity in the conditions in Definition 30, i.e. the values $q(\varphi_i)$,

$q(\mu X_i.\varphi_i)$, and $q(\nu X_i.\varphi_i)$ do not depend on the value of $q(X_i)$. Furthermore, all the conditions are deterministic, that is to say, there is a unique value of $q(\psi)$ satisfying the conditions for any sub-formula ψ.

Part (ii) can easily be proved by induction on the structure. $\square$

We are now ready to define the labelling systems for L_μ-formulas.

Definition 33 *For every $n \in \omega$ define the labelling system $\mathcal{L}_\varphi(n)$ as follows.*

- *Q is the set of all globally consistent functions $q : \mathrm{sf}(\varphi) \to \{0, \ldots, n+1, \bot\}^k$.*
- *$\mathcal{A}$ is the set of actions in the signature and $\mathcal{P}$ is the set of proposition symbols in the signature.*
- *For each $M \subseteq Q \times \mathcal{A}$ and $P \subseteq \mathcal{P}$, $\delta(M, P)$ is the function q that is globally (M, P)-consistent.*
- *Finally, $\mathcal{F} := \{q \in Q : q(\varphi) \neq \bot\}$.*

It follows from Proposition 32 above that $\mathcal{L}_\varphi$ is well-defined. Also, by construction, the size of any such labelling system $\mathcal{L}_\varphi$ is bounded by a polynomial in n whose exponent only depends on φ.

We now prove that the labelling systems of Definition 33 work as intended. The proof is split into two separate lemmas.

Lemma 34 *Let $\mathcal{K}, v$ be a transition system of rank no more than n. If $\mathcal{K}, v \models \varphi$ then $\mathcal{L}_\varphi(n)$ accepts $\mathcal{K}, v$.*

Proof. For each node u define a function $q_u : \mathrm{sf}(\varphi) \to \{0, \ldots n+1, \bot\}$ as follows. Let $\psi \in \mathrm{sf}(\varphi)$ be a sub-formula. For all i such that ψ is not a sub-formula of φ_i define $\Big(q(\psi)\Big)_i := \bot$. For all other i consider the stages of the induction on φ_i where again all free variables of φ_i other than X_i are interpreted by their fixed points. If $\mathcal{K}, u \models \psi(X_i^\infty)$ then let n be the least ordinal such that $\mathcal{K}, u \models \psi(X_i^n)$ and set $\Big(q(\psi)\Big)_i := n$ if i corresponds to a μ-operator and set $\Big(q(\psi)\Big)_i = 0$ if it corresponds to a ν-operator. Otherwise define $\Big(q(\psi)\Big)_i := \bot$.

It is now a simple observation that the function $f : V \to Q$ assigning to each node $u \in V$ the function q_u is a $\sim$-consistent labelling witnessing that $\mathcal{L}_\varphi$ accepts $\mathcal{K}, v$. $\square$

We now turn to the converse.

Lemma 35 *Let $\mathcal{K}, v$ be a Kripke-structure of rank at most n and let $\varphi \in L_\mu$ be a formula. If $\mathcal{L}_\varphi(n)$ accepts $\mathcal{K}, v$, then $\mathcal{K}, v \models \varphi$.*

Proof. Let f be a labelling witnessing that $\mathcal{L}_\varphi$ accepts $\mathcal{K}, v$. We claim that for all nodes $u \in V$, all $i \leq k$, and $\psi \in \mathrm{sf}(\varphi)$, if $q = f(u)$ is the label assigned to u by f $\Big(q(\psi)\Big)_i = m \neq \bot$ then

- $\mathcal{K}, u \models \psi(X_i^m)$ if X_i is a μ-variable and
- $\mathcal{K}, u \models \psi(X_i^\infty)$ if X_i is a ν-variable.

Clearly, this implies the lemma as by the definition of acceptance, v must be labelled by a label in $\mathcal{F}$ and these labels all map φ to something other than $\bot$.

We prove the claim, simultaneously for all nodes u, by simultaneous induction along the lexicographic order on the tuples $f(u)(\psi)$, and the structure of the formula ψ. Let $q := f(u)$ be the label assigned to the node u. Further, let P be the set of propositions true at u and let

$$M := \{(q', a) : a \in \mathcal{A} \text{ and } q' = f(u') \text{ for some } a\text{-successor } u' \text{ of } u\}.$$

Now, assume $\psi \in \mathrm{sf}(\varphi)$ is a sub-formula of φ and $q(\psi) = m \neq \bot$.

- Suppose $\psi := p$ and let $i \leq k$. By definition, if q is the label assigned to u by f, and $\Big(q(p)\Big)_i \neq \bot$, then this implies that $p \in P$ and thus $\mathcal{K}, u \models \psi$.
- The case $\psi := \neg p$ is analogous.
- The cases for Boolean connectives and modal operators are obvious.
- Let $\psi := \lambda X_i.\varphi_i$ for $\lambda \in \{\mu, \nu\}$. Suppose first that $\lambda = \mu$ and consider some $j < i$. By definition of δ, if $\Big(q(\psi)\Big)_j = m \neq \bot$, then $\Big(q(\psi)\Big)_j = \Big(q(\varphi_i)\Big)_j$. By induction on the structure of the formulas, this implies that $(\mathcal{K}, X_j^m), u \models \varphi_i$. Hence, u occurs in the fixed point of X_i and $(\mathcal{K}, \tilde{X}_j^m), u \models \psi$.
  Now suppose $j = i$. By definition, $\Big(q(\psi)\Big)_i \neq \bot$ implies $\Big(q(\psi)\Big)_i = n + 1$ and $\Big(q(\varphi_i)\Big)_i = m \neq \bot$. By induction on the stages, this implies that $(\mathcal{K}, X_i^m), u \models \varphi_i$ and therefore also $(\mathcal{K}, X_i^m), u \models \psi$.
  The case $\lambda = \nu$ is analogous.
- Finally, suppose $\psi := X_i$. Assume first that X_i is bound by a μ-operator. If $\Big(q(X_i)\Big)_i \neq \bot$, then, by definition, $\Big(q(\varphi_i)\Big)_i = m \neq \bot$ and $\Big(q(X_i)\Big)_i = m + 1$. Thus, by induction on the stages, $(\mathcal{K}, X_i^m), u \models \varphi_i$ and therefore $(\mathcal{K}, X_i^{m+1}), u \models X_i$.
  Now consider some $j > i$. By definition, $\Big(q(X_i)\Big)_j := 0$. We have to show that $(\mathcal{K}, X_j^0), u \models X_i$. Recall that this means that X_i is true at the node u under the interpretation of X_j by its 0^{th} stage and where all other fixed point variables, including X_i, are interpreted by their respective fixed points. As proved in the preceding paragraph, u occurs in the fixed point of X_i. From this, the claim follows immediately.
  Now suppose X_i is bound by a ν-operator. By definition, $\Big(q(X_i)\Big)_j \neq \bot$

implies $\Big(q(X_i)\Big)_j = 0$. As X_i is bound by a greatest fixed-point operator, X_i^0 contains the whole universe and therefore X_i holds true at u.

This proves the claim and thus the lemma. □

This concludes the proof of Theorem 29. A consequence of the proof is that if a L_μ-formula does not use any μ-operators, the class of structures defined by it has constant labelling index. Thus, to give an example of a L_μ definable class of structures with non-constant labelling index, the exclusive use of ν-operators is not sufficient. But it can easily be seen, using pumping arguments, that to express reachability, constant size labelling systems are not sufficient.

Proposition 36 *There is an L_μ-definable class $\mathcal{C}$ of structures that has a linear lower bound on its labelling index.*

Proof. Let $\mathcal{C}$ be the class of transition systems such that there is a node reachable from the root labelled by the proposition p, i.e. the class of transition systems $\mathcal{K}, v$ satisfying the L_μ-formula $\mu X.(p \vee \Diamond X)$.

Obviously, $\mathcal{C}$ can be accepted by a family of labelling systems of linear size. We wish to prove that we cannot do better. Assume otherwise and suppose that for some $n > 2$ there is a labelling system $\mathcal{L}$ of size less than n accepting the class $\mathcal{C}_n$ of structures from $\mathcal{C}$ of rank at most n. Consider the structure $\mathcal{K} := (\{0, \dots n-1\}, E, P)$ with $E := \{(i, i+1) : 0 \leq i < n-1\}$ and $P := \{n-1\}$. Obviously $\mathcal{K}, 0 \in \mathcal{C}_n$ and thus $\mathcal{K}, 0$ is accepted by $\mathcal{L}$. As there are fewer than n labels, there must be two different nodes $u < v < n-1$ in $\mathcal{K}$ labelled by the same label q in $\mathcal{L}$. But then the same labelling also witnesses that the system $\mathcal{K}' := (\{0, \dots, v\}, E', P')$ where $E' := \{(i, i+1) : 0 \leq i < v\} \cup \{(v, u+1)\}$ and $P' := \emptyset$, would be accepted by $\mathcal{L}$. As $\mathcal{K}', 0 \notin \mathcal{C}$ we get a contradiction. □

Theorem 29 along with Proposition 36 establishes both upper and lower bounds on the labelling index of properties definable in L_μ. There remains a gap between these. We conjecture that Proposition 36 can be strengthened to show that for any polynomial p , there is a property definable in L_μ with labelling index $\Omega(p)$.

Proposition 36 also shows that there are properties definable in various ML extensions like LTL, CTL, or CTL* which have non-constant labelling index, as reachability can be expressed in these logics.

6 Labelling Index and Complexity

We begin by contrasting labelling index with the usual notion of computational complexity in terms of running time on a machine measured as a function of the size of the structure. We demonstrate that the two measures are not really comparable by exhibiting a class of structures that is decidable in polynomial time but has non-elementary labelling index and on the other hand an NP-complete problem that has exponential labelling index.

The first of these is the class of finite trees $\mathcal{F}$ such that if t_u are t_v are subtrees rooted at a successor of the root, then $t_u \sim t_v$. As was shown in [4] and also explained in the paragraph following Proposition 4, there is no elementary bound on the automaticity of this class, but it is decidable in time polynomial in the *size* of the tree. This yields the following result.

Proposition 37 *There is a polynomial-time decidable class of Kripke structures with non-elementary labelling index.*

In contrast, we can construct an NP-complete problem of much lower labelling index. We obtain this by encoding propositional satisfiability as a class of structures $\mathcal{S}$ closed under bisimulation, and demonstrate that it is accepted by an exponential family of labelling systems.

Theorem 38 *There are* NP*-complete classes with exponential labelling index.*

Proof. Let φ be a propositional formula formed from the propositional variables $V_1, \ldots, V_n$ using the Boolean operations $\wedge$, $\vee$ and $\neg$. We define an encoding of φ as a Kripke structure $\mathcal{T}_\varphi$ in the vocabulary with a single action a and propositional vocabulary $\{\wedge, \vee, \neg, V, \mathrm{Count}\}$. $\mathcal{T}_\varphi$ is defined inductively as follows.

- If φ is a variable V_i, $\mathcal{T}_\varphi$ consists of a root labelled V which is connected to a chain of length i of nodes labelled Count, and with an edge from the root to itself.
- If φ is $\psi_1 \wedge \psi_2$, $\mathcal{T}_\varphi$ consists of a root labelled $\wedge$ with two successors which are roots of $\mathcal{T}_{\psi_1}$ and $\mathcal{T}_{\psi_2}$. The rule for $\psi_1 \vee \psi_2$ is similar.
- If φ is $\neg\psi$, $\mathcal{T}_\varphi$ consists of a root labelled $\neg$ with a single successor which is the root of $\mathcal{T}_\psi$.

Now, the class of structures $\mathcal{S}$ is defined by

$$\mathcal{S} := \{\mathcal{K} : \mathcal{K} \sim \mathcal{T}_\varphi \text{ for a satisfiable formula } \varphi\}.$$

It is immediate from the definition that $\mathcal{S}$ is bisimulation closed and NP-completeness follows from the obvious reduction. We now demonstrate an exponential family of labelling systems that accepts $\mathcal{S}$.

Let TA_n denote the set of possible truth assignments to the variables $\{V_1, \ldots, V_n\}$. Thus, TA_n is a set of size 2^n. We define the labelling system $\mathcal{L}_n$ to have the set of labels $\{T, F\} \times \mathrm{TA}_n \cup \{\mathrm{C}_i : 1 \le i \le n\} \cup \{\mathrm{Fail}\}$. The transition function δ is defined so that:

- For every node where the proposition Count holds, the node must be labelled C_{i+1} if all its successors are labelled C_i, and Fail otherwise.
- If the proposition V holds at a node, and there is an i and a state $(v, As) \in \{T, F\} \times \mathrm{TA}_n$ with $v = As(i)$ such that every successor of the node is labelled either C_i or (v, As), then the node must be labelled (v, As), and it must be labelled Fail otherwise.
- A node where $\wedge$ holds is labelled (T, As), where $As \in \mathrm{TA}_n$ if all of its successors are labelled (T, As). It is labelled (F, As) if one of its successors is labelled (F, As) and the others are labelled (v, As) for some $v \in \{T, F\}$. In all other cases it is labelled Fail.
- A node where $\vee$ holds is labelled (F, As), if all of its successors are labelled (F, As). It is labelled (T, As) if one of its successors is labelled (T, As) and the others are labelled (v, As) for some $v \in \{T, F\}$. In all other cases it is labelled Fail.
- A node where $\neg$ holds is labelled (T, As) if all of its successors are labelled (F, As). It is labelled (F, As) if all of its successors are labelled (T, As). In all other cases it is labelled Fail.

Finally, the set $\mathcal{F}$ of accepting states is the set $\{T\} \times \mathrm{TA}_n$.

It is easily seen that, for a propositional formula φ with no more than n variables, $\mathcal{L}_n$ accepts $\mathcal{T}_\varphi$ if, and only if, φ is satisfiable. Moreover, $\mathcal{L}_n$ is clearly bisimulation invariant and we therefore conclude that the family $(\mathcal{L}_n : n \in \omega)$ accepts $\mathcal{S}$. $\square$

It is an open question whether the exponential bound in Theorem 38 is optimal. In particular, is there an NP-complete class of transitions systems with constant labelling index?

6.1 The trace-equivalence problem

We now apply our methods to a particular problem that is of interest from the point of view of verification—the *trace-equivalence* problem. We determine exactly the labelling index of a number of variations of the problem and thereby derive results about their expressibility in various modal fixed-point logics.

Consider a Kripke structure $\mathcal{K}, v$ with set of actions $\mathcal{A}$ and a distinguished proposition symbol $\mathcal{F}$ denoting accepting nodes. We define the set of *traces* of the structure to be the set $\mathcal{T} \subseteq \mathcal{A}^*$ such that $t \in \mathcal{T}$ just in case there is a path labelled t from v to a node in $\mathcal{F}$. Two structures are said to be *trace equivalent* if they have the same set of traces.

To define the decision problem of trace equivalence as a bisimulation-closed class of structures, we consider

$$\mathcal{E} = \{\mathcal{K}, v : \text{ if } v \to u \text{ and } v \to w \text{ then } \mathcal{K}, u \text{ and } \mathcal{K}, w \text{ are trace equivalent}\}.$$

In other words, it is the set of rooted structures such that any two successors of the root admit the same set of traces.

The unary trace-equivalence problem is $\mathcal{E}$ restricted to structures over a vocabulary with a single action, i.e. $\mathcal{A} = \{a\}$. Similarly, we define binary trace equivalence to be the class of structures over a vocabulary with action set $\{a, b\}$ that are also in $\mathcal{E}$.

The problem of deciding the trace equivalence of two structures is computationally equivalent to deciding language equivalence of nondeterministice finite automata. In particular, the unary trace equivalence problem corresponds to equivalence of automata over a one-letter alphabet and the binary version to equivalence of automata over a two-letter alphabet. These problems are known to be co-NP-complete and PSPACE-complete respectively (see [6]). However, we are interested in the complexity of these problems in terms of their rank rather than their size. We also find it useful to distinguish between the problems for acyclic structures and structures which may have cycles. In terms of classical complexity, the trace equivalence problem on acyclic structures is solvable in polynomial time in both the unary and binary cases by a straightforward algorithm that recurses on the height. In the following, we will regard a rooted structure $\mathcal{K}, v$ as a finite automaton with start state v, with the states satisfying $\mathcal{F}$ being the final states.

Lemma 39 *If $\mathcal{K}, v$ and $\mathcal{K}', v'$ are two acyclic structures with rank$(\mathcal{K})$, rank$(\mathcal{K}') \leq n$ that are trace inequivalent, then they are distinguished by some trace of length at most n.*

Proof. Trivial, as neither structure admits any trace of length greater than n. □

Proving a bound on the length of traces needed to distinguish states in a structure with cycles is somewhat more involved. It is possible to show that if $\mathcal{K}, v$ and $\mathcal{K}, v'$ are distinguished by some trace, then they are distinguished by a trace that is of length at most 2^{2n}, where n is the *size* of $\mathcal{K}$ (see, for instance, [3, p.167]). However, we know that the rank of a structure can be much smaller than its size, so this does not yield a corresponding bound in terms of the rank of $\mathcal{K}$. Nonetheless, using Lemma 39, we can establish an upper bound on the labelling index of the trace equivalence problem for acyclic structures.

Theorem 40 *(i) On acyclic structures, unary trace equivalence has at most exponential labelling index.*
(ii) On acyclic structures, binary trace equivalence has at most double exponential labelling index.

Proof.

We know, by Lemma 39, that if two nodes in an acyclic structure of rank n are trace inequivalent, then there is a trace of length at most n that distinguishes them. We therefore aim to associate with each node v the set of traces of length at most n available from v. We call this set D_v the *discriminating set* of v and note that if u and v are not trace equivalent then $D_u \neq D_v$.

Thus, writing $\mathcal{A}^{\leq n}$ to denote the set of strings over alphabet $\mathcal{A}$ of length n or less, we define $\mathcal{D}$, the collection of discriminating sets for the two separate cases as follows:

(i) for unary acyclic structures $\mathcal{D} = \mathrm{Pow}(\{a\}^{\leq n})$;
(ii) for binary acyclic structures $\mathcal{D} = \mathrm{Pow}(\{a, b\}^{\leq n})$;

Note that the number of elements in $\mathcal{D}$ is exponential in n in case 1 and double exponential in case 2.

Now, to establish the result, we define a labelling system $\mathcal{L}_n$ with state set $\mathcal{D} \times \{e, n\}$ and a transition function such that if a node v has a-successors labelled $(S_1, x_1), \ldots, (S_k, x_k)$ and b-successors labelled $(T_1, y_1), \ldots, (T_l, y_l)$, then v is labelled (S, x) where

- $S = \{as : s \in S_i \text{ for some } i\} \cup \{bt : t \in t_i \text{ for some } i\} \cup X$, where $X = \{\varepsilon\}$ if $v \in \mathcal{F}$ and $X = \emptyset$ otherwise; and
- $x = e$ if $S_1 = \cdots = S_k = T_1 = \cdots = T_l$ and $x = n$ otherwise.

It is then easily seen that on structures with rank n or less, if there is a valid labelling of a structure with the system $\mathcal{L}_n$ then a node gets label (S, e) if, and only if, S is the discriminating set of that node, and all successors of the node have the same discriminating set. Thus, setting $\mathcal{D} \times \{e\}$ to be the set of accepting labels, we have our required family of labelling systems. □

Proving matching lower bounds is relatively straightforward.

Theorem 41 • *On acyclic structures, unary trace equivalence has at least exponential labelling index.*

- *On acyclic structures, binary trace equivalence has at least double exponential labelling index.*

Proof. If $\mathcal{D}$ is as in the proof of Theorem 40, it is easy to see that for every set $D \in \mathcal{D}$ we can construct a rooted structure $\mathcal{K}_D, v$ of rank n such that its discriminating set is exactly D. Indeed, such a structure can be a tree with a branch for every string in D. Assume, towards a contradiction, that there is a labelling system $\mathcal{L}$ with fewer than $|\mathcal{D}|$ labels that accepts trace equivalence up to rank $n+1$. Then, there must be distinct D and D' such that the roots of $\mathcal{K}_D$ and $\mathcal{K}_{D'}$ receive the same label in an accepting labelling by $\mathcal{L}$. It then follows that the structure $\mathcal{K}_E$ consisting of a root with two successors which are both roots of disjoint copies of $\mathcal{K}_S$ and the structure $\mathcal{K}_N$ consisting of a root with two successors which are roots of $\mathcal{K}_S$ and $\mathcal{K}_{S'}$ both receive the same label. However, as both $\mathcal{K}_E$ and $\mathcal{K}_N$ are of rank $n+1$, $\mathcal{K}_E \in \mathcal{E}$ and $\mathcal{K}_N \notin \mathcal{E}$, we have a contradiction. $\square$

For the case of structures with cycles, we can also establish a lower bound though the construction is not as straightforward. We are able to establish a double exponential lower bound on the labelling index of the trace equivalence problem for both unary and binary structures. This relies on the following lemma, which is based on a construction used by Stockmeyer and Meyer [12] to show the NP-completeness of the language inequivalence problem for nondeterministic finite automata.

Lemma 42 *There is a polynomial p such that there is a collection of 2^{2^n} unary structures of rank $p(n)$ with pairwise distinct discriminating sets.*

Proof. Theorem 6.1 in [12] gives a logarithmic space reduction from the problem of satisfiability of 3CNF formulas to the problem of determining language inequivalence of nondeterministic automata over a one-letter alphabet (which is, of course, equivalent to the problem of trace inequivalence of unary structures). The essence of the reduction defines an encoding of truth assignments over the Boolean variables $x_1, \ldots, x_n$ as numbers. Then, it is shown that we can construct, for each 3CNF formula φ an automaton which accepts a string of length z if, and only if, z encodes a truth assignment satisfying φ.

Now, for any fixed truth assignment t to the variables $x_1, \ldots, x_n$ it is easy to construct a 3CNF formula (with possibly additional variables, but linearly bounded in n) which is satisfied uniquely by t. Thus, by the reduction defined in [12], we have an automaton A_t which accepts a string of length z if, and only if, z encodes t. Moreover, as the reduction is computable in logarithmic space, there is a polynomial $q(n)$ that bounds the size of A_t.

Now take any set T of truth assignments (and there are 2^{2^n} of these) and define the automaton which branches from its start state to the start state of a copy of A_t for each $t \in T$. The rank of this automaton is $q(n) + 1$, though its size may be exponential in n. But, it is clear that the automaton accepts a string of length $z + 1$ if, and only if, z encodes a truth assignment in T. In other words, there are as many pairwise distinct discriminating sets as there are sets of truth assignments. □

Theorem 43 *On structures which may have cycles, unary trace equivalence has at least double exponential labelling index.*

Proof. The argument is exactly as in the proof of Theorem 41 except that in place of trees we use the structures obtained in Lemma 42. □

It follows from Theorems 41 and 43 that none of the trace equivalence properties is definable in L_μ. However, it can be shown that unary trace equivalence on acyclic structures is definable in MIC, giving another example of a property separating these two logics. Moreover, it also follows that binary trace equivalence on acyclic structures is not definable in MIC. Since this property is polynomial time decidable and bisimulation invariant, it gives us another instance of a property separating MIC from L_μ^ω. Finally, we note that on arbitrary structures, neither the unary nor the binary trace equivalence problem is definable in MIC. Since the former problem is co-NP-complete and the latter is PSPACE-complete, we do not expect that either is definable in L_μ^ω, but it would be difficult to prove that they are not.

References

[1] A. Arnold and D. Niwiński. *Rudiments of μ-calculus.* North Holland, 2001.

[2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press, 2001.

[3] A. Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143:154–174, 1998.

[4] A. Dawar, E. Grädel, and S. Kreutzer. Inflationary fixed points in modal logic. *ACM Transactions on Computational Logic*, pages 282–315, 2004. A preliminary version appeared in *Proc. of the 10th Conf. on Computer Science Logic, LNCS 2142*.

[5] A. Dawar and S. Kreutzer. Generalising automaticity to modal properties of finite structures. In *Proc. 22nd Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 2556 of *LNCS*, pages 109–120. Springer, 2002.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

[7] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 1–68. Springer, 1997.

[8] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of 7th International Conference on Concurrency Theory CONCUR '96*, volume 1119 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

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

[10] M. Otto. Bisimulation-invariant Ptime and higher-dimensional mu-calculus. *Theoretical Computer Science*, 224:237–265, 1999.

[11] J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. *Journal of Computer and System Sciences*, 53:10–25, 1996.

[12] L. Stockmeyer and A.R. Meyer. Word problems requiring exponential time. In *5th ACM Symp. on Theory of Computing*, pages 1–9, 1973.

[13] W. Thomas. On logics, tilings, and automata. In J. Leach et al., editor, *Automata, Languages, and Programming*, Lecture Notes in Computer Science Nr. 510, pages 441–453. Springer-Verlag, 1991.

[14] W. Thomas. Finite-state recognizability and logic: from words to graphs. In *13th World Computer Congress 94*, volume 1, pages 499–506. Elsevier Science, 1994.

[15] W. Thomas. Languages, automata and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.