Generalising Automaticity to Modal Properties of Finite Structures

A. Dawar[1]* and S. Kreutzer[2]

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

[2] LuFG Mathematische Grundlagen der Informatik, RWTH Aachen,
`kreutzer@informatik.rwth-aachen.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 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_μ^ω, the higher dimensional μ-calculus introduced by Otto [6], and MIC, the modal iteration calculus, introduced in [3]. 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 formulae of the first into the second. Establishing separations between logics is, in general, more involved. This requires identifying a property expressible in one logic and proving

* Research supported by EPSRC grant GR/N23028.

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.

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. The labelling index generalises the notion of the automaticity of languages (see [7].) The automaticity of a language (i.e. set of strings) L is the function that maps n to the size of the least 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.

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. We demonstrate that MIC is not the bisimulation-invariant fragment of monadic IFP. We also investigate the relationship between labelling index and conventional time and space based notions of complexity. Finally, we investigate the labelling index of the trace equivalence problem over specific classes of structures and deduce interesting results about its expressibility in various fixed-point logics.

Due to lack of space, proofs of the results are only sketched.

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, 3].

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 $\mathcal{P}$.

Modal logic (ML) is built up from atomic propositions $p \in \mathcal{P}$ using boolean connectives and the *next-modalities* $\langle a \rangle$, $[a]$ for each $a \in \mathcal{A}$. Formulae $\varphi \in \mathrm{ML}$ 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-formulae is as usual with $\mathcal{K}, v \models \langle a \rangle \varphi$ 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, $\mathcal{K}, u \models \varphi$.

Bisimulations. Bisimulation is a notion of behavioural equivalence for transition systems (see, e.g. [8] for a definition). Modal logics, like ML, CTL, the μ-calculus etc. do not distinguish between transition systems that are bisimulation equivalent. We write $\mathcal{K}, v \sim \mathcal{K}', v'$ to denote that the two transition systems are equivalent by bisimulation.

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 v in $\mathcal{K}$. It is easily verified that $\mathcal{K}, v \sim \mathcal{K}_{/\sim}, [v]$.

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

Syntactically they are defined as the closure of modal logic under the following formula building rules. Let $\varphi_1, \dots, \varphi_k$ be formulae with free proposition symbols $X_1, \dots,$ X_k and let $S := \{X_1 \leftarrow \varphi_1, \dots, X_k \leftarrow \varphi_k\}$ be a system of rules. Then $\mu X_i : S$ and $\nu X_i : S$ are formulae of L_μ and (**ifp** $X_i : S$) is a formula of MIC, where, in the case of L_μ, the rule is restricted to systems S where all formulae φ_i in S are positive in all fixed-point variables X_j.

On any finite transition system $\mathcal{K}$ with universe V, such a system S of rules defines an operator F_S taking a sequence $(X_1, \dots, X_k)$ of subsets of V to the sequence $(F_S(X_1), \dots, F_S(X_k))$, where $F_S(X_i) := \{u : (\mathcal{K}, (X_i)_{1 \le i \le k}), u \models \varphi_i\}$. This operator, again, inductively defines for each finite ordinal α, a sequence of sets $(X_1^\alpha, \dots, X_k^\alpha)$ as follows. For all i, $X_i^0 := \emptyset$ and for $0 < \alpha < \omega$, $X_i^\alpha := (F_S(\overline{X}^{\alpha-1}))_i$.

As the formulae in the μ-calculus are required to be positive in their free fixed-point variables, the operator F_S induced by a system of L_μ-operators is monotone and thus always has a least and a greatest fixed point. By a well known result of Knaster and Tarski, the least fixed point is also reached as the fixed point $(X_1^\infty, \dots, X_k^\infty)$ 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. $X_i^0 := V$. 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 if, and only if, $u \in X_i^\infty$. 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. [1]

The next fixed-point extension of ML we consider is the modal iteration calculus introduced in [3]. It is designed to overcome the restriction of L_μ to positive formulae, but still guarantee the existence of a meaningful fixed point. This is achieved by taking at each induction step the union with the previous stage, i.e. $X_i^{\alpha+1}$ is defined as $X_i^{\alpha+1} := X_i^\alpha \cup (F_S(\overline{X}^\alpha))_i$. Thus, the stages of the induction are increasing and lead to a fixed point $(X_1^\infty, \dots, X_k^\infty)$. Again, $\mathcal{K}, u \models$ **ifp** $X_i : S$ if, and only if, $u \in X_i^\infty$.

[1] 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]).

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 [6] for a precise definition. Here we only note that this logic permits the formation of least fixed points of positive formulae 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 $\mathrm{ML} \subseteq L_\mu \subseteq \mathrm{MIC} \subseteq \mathrm{IFP}$, 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_μ^ω is the bisimulation invariant fragment of LFP, is follows that $\mathrm{MIC} \subseteq L_\mu^\omega$. Indeed, all of these inclusions are proper. The separations of MIC from L_μ and L_μ^ω were shown in [3]. 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 formulae into monadic second-order logic. Indeed, Janin and Walukiewicz [5] 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. Thus, the separation of MIC from L_μ shows that MIC can express properties that are not definable in monadic second-order logic. In [3], the question was posed whether MIC could be characterised as the bisimulation invariant fragment of any natural logic. The most natural candidate for this appears to be the monadic fragment of IFP—the extension of first order predicate logic with inflationary fixed points. However, by an analysis of the labelling index of properties definable in this logic, we show that it can express bisimulation-invariant properties that are not in MIC.

3 Automaticity on Strings and Trees

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

In [3] it was shown that MIC is strictly less expressive than L_μ^ω. The full version of that paper makes it clear that 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, [4]). We consider "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 4.6 below offer such a generalisation.

For a bisimulation-closed class $\mathcal{C}$ of trees, its automaticity can be defined (see the full version of [3]) 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_μ^ω. Indeed, one can establish the following facts about the automaticity of classes of trees definable in modal fixed-point logics.

Proposition 3.1. *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_μ^ω 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 [11]). Statements (2), (3) and (4) are shown in [3]. However, (2) can also be derived as a special case of Theorem 5.1 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 subtrees T_1 and T_2 rooted at children of the root of T, we have $T_1 \sim T_2$. It can be seen that the automaticity of this class is the maximum possible.

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. A consequence is that MIC is not the bisimulation invariant fragment of monadic IFP, something that could naturally be conjectured, given that the μ-calculus is the bisimulation-invariant fragment of monadic least fixed-point logic.

We first introduce monadic inflationary fixed-point logic $(\mathrm{M}\text{-}\mathrm{IFP})$ as the closure of first-order logic under the following rule. If $\varphi(X, x)$ is a formula with a free unary relational 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. $[\mathbf{ifp}_{X,x}\, \varphi]$ defines the inflationary fixed point of the operator induced by φ.

The properties 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 3.2. *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 3.3. *The number of distinct $\simeq_k$ equivalence classes of trees of height $n + k$ or less is k-fold exponential in n.*

Now, 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. By Lemma 3.3, the automaticity of $\mathcal{C}_k$ is at least k-fold exponential. Also it is easy to see that $\simeq_k$-equivalence is M-IFP-definable. This establishes the following theorem.

Theorem 3.4. *For every elementary function f, there is a property with automaticity $\Omega(f)$ definable in* M-IFP.

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 [3].

Corollary 3.5. 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 starting or ending 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 4.6.

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 4.1. *The* rank *of a structure $\mathcal{K}, v$ 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 there is a path from v_i to v_{i+1} for each i.*

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 non-repeating path. This observation can be further generalised by the following equivalent characterisation of rank.

Definition 4.2. *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 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 rooted finite transition system $\mathcal{K}, v$ is defined as the rank of the block containing v in the block decomposition of $\mathcal{K}$.*

Lemma 4.3. *The block rank of $\mathcal{K}, v$ is equal to its rank.*

When relating tree-automata to fixed-point logics as in Proposition 3.1, the key property of the height of the 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.

Lemma 4.4. *If $\varphi(X)$ is a formula of* MIC*, and n is the rank of $\mathcal{K}, v$, then $\mathcal{K}, v \models$ $(\textbf{ifp}\ X : \varphi)$ if, and only if, $v \in X^n$.*

Proof. The proof is by induction on n. The basis, $n = 1$ is trivial. For the induction step, we show that for $k \geq 1$, if $v \in X^{k+1}$ but $\boldsymbol{v} \notin X^k$, then there must be a u reachable from v such that $u \in X^k \setminus X^{k-1}$. □

While the rank of a structure $\mathcal{K}, v$ 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 the required converse to Lemma 4.4.

Lemma 4.5. *If the rank of $\mathcal{K}_{/\sim}$ is n, there is a formula $\varphi(X)$, positive in X, whose closure ordinal on $\mathcal{K}$ is n.*

An immediate consequence of Lemmas 4.4 and 4.5 is that the maximal closure ordinals of simple MIC and simple L_μ formulae 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 4.6. *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 for each $s \in V$, $f(s) = \delta(\{(f(s'), a) : a \in \mathcal{A}$ and $(s, s') \in E_a\}, \{i : i \in \mathcal{P}$ and $s \in P_i\})$, and $f(v) \in \mathcal{F}$.

As δ is functional, labelling systems are deterministic devices. Indeed, on well-founded trees, labelling systems and bottom-up tree automata are equivalent. 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 that witnesses 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 4.7. Consider the labelling system $\mathcal{L} = (Q, A, P, \delta, \mathcal{F})$ given by: $Q = \{q, q'\}$, $A = \{a\}, P = \emptyset, \mathcal{F} = \{q\}$ and where δ is given by the rules $\delta(\emptyset) = q$, $\delta(\{(q, a)\}) = q'$, $\delta(\{q', a\}) = q$ and $\delta(\{(q, a), (q', a)\}) = q$, where we have dropped the second argument to δ as it is always $\emptyset$.

This labelling system accepts a simple cycle if, and only if, it 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 4.8. *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 and 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 4.9. *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 $\mathcal{K}, v$ *connected* if, for every node u, there is a path from v to u in $\mathcal{K}$.

Lemma 4.10. *On connected structures, a labelling system is $\sim$-consistent if, and only if, it is $\sim$-invariant.*

While any $\sim$-consistent labelling systems $\mathcal{L}$ defines a class of $\sim$-invariant structures, not every bisimulation-closed class $\mathcal{C}$ of structures is given by such a labelling system. However, as we see below, $\mathcal{C}$ is 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 show now that every bisimulation closed class of transition systems can be accepted by a family of labelling systems as follows. For this, note that the rank is trivially bounded by the size of the transition system and that there are only finitely many bisimulation equivalence classes of structures of a given rank n. Taking a state for each such class yields the desired labelling system.

Lemma 4.11. *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 $\mathfrak{A}$ with rank$(\mathfrak{A}) \leq n$, $\mathcal{L}_n$ accepts $\mathfrak{A}$ if, and only if, $\mathfrak{A} \in \mathcal{C}$.*

The minimal size in terms of n of the labelling systems in a family such as that in Lemma 4.11 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 4.12. *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 the smallest labelling system such that for any $\mathcal{K}, v$ of rank n or less, $(\mathcal{K}, v) \in \mathcal{C}$ if, and only if, $\mathcal{K}, v \models \mathcal{L}_n$.*

A comparison of labelling systems with other automata models on graphs, such as tiling systems [9, 10] is instructive. Significant differences are that tiling systems are generally nondeterministic and the label attached to a node depends on its predecessors as well as its successors.

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.

The modal iteration calculus. It was shown by Dawar, Grädel, and Kreutzer in [3] that any class of trees definable in MIC has at most exponential automaticity. The proof translates easily to the labelling index on arbitrary structures, as sketched below.

Let φ be a formula in MIC and let Φ be the set of sub-formulae of φ. Further let $X_1, \dots, X_k$ be the fixed-point variables occurring in φ. Clearly, for every transition system $\mathcal{K}, v$ of rank n the fixed point of each inflationary induction must be reached after at most n stages.

For every transition system $\mathcal{K}, v$ of rank n we define the *φ-type of $\mathcal{K}, v$* as the function $f : \{0, \dots, k \cdot n\}^k \to 2^{\Phi}$ such that $\psi \in \Phi$ occurs in $f(\bar{\imath})$ if, and only if, ψ holds at the root v of $\mathcal{K}$ if the variables occurring free in ψ are interpreted by the stages $X_j^{i_j}$. A function $f : \{0, \dots, k \cdot n\}^k \to 2^{\Phi}$ is a *φ-type* if it is a φ-type of a transition system.

We are able to define for each formula in MIC a family of labelling systems accepting the class of its models, where the φ-types serve as labels. This gives us the following theorem.

Theorem 5.1. *Every* MIC *definable class of transition systems has at most exponential labelling index.*

There is a corresponding lower bound, as it is shown in [3] that there are MIC-definable classes of structures with exponential labelling index.

Another corollary of the results refers to modal logic. As ML-formulae can be seen as MIC-formulae 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 5.2. *Every property definable in* ML *has constant labelling index.*

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-formulae φ. In each node u of the structures we remembered for the sub-formulae of φ every tuple $(i_1, \dots, i_k)$ of induction stages where the sub-formula becomes true at u. As L_μ formulae are monotone, if a sub-formula is true at a tuple of stages $(i_1, \dots, i_k)$ it will also be true at all higher stages of μ and all lower stages of ν-operators. Thus, 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 give it a special label if it is not included at all. We would thus only have linearly many labels in the labelling system.

But monotonicity also helps if there are more than one fixed-point operator. The reason is that if a 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 respective 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. We omit a detailed proof for lack of space.

Theorem 5.3. *Every class of transition systems definable in L_μ has at most polynomial labelling index.*

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.

Lemma 5.4. *There is a 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. Obviously, $\mathcal{C}$ can be accepted by a family of labelling systems with linear growth function.

On the other hand, each family of labelling systems accepting $\mathcal{C}$ must have at least linear size. 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 less than n labels, there must be two different nodes $u < v$ in $\mathcal{K}$ labelled by the same label q in $\mathcal{L}$. But then the same labelling by label from $\mathcal{L}$ 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. $\square$

Thus, the μ-calculus has a polynomial labelling index in the sense that every L_μ definable property has polynomial labelling index and there are L_μ-definable properties with a linear lower bound on the labelling index.

This also shows that various ML extensions like LTL, CTL, or CTL* have non-constant labelling index, as they can express reachability.

6 Labelling Index and Complexity

We begin by contrasting labelling index with the usual notion of computational complexity in terms of machine time 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 [3], 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 6.1. *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 6.2. *There are* NP*-complete classes with exponential labelling index.*

It is an open question whether the exponential bound in Theorem 6.2 can be lowered.

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

Theorem 6.3. *(i) On acyclic structures, unary trace equivalence has exponential labelling index.*
(ii) On acyclic structures, binary trace equivalence has double exponential labelling index.
(iii) On arbitrary structures, unary trace equivalence has double exponential labelling index.
(iv) On arbitrary structures, binary trace equivalence has a treble exponential labelling index.

Proof. The proofs of the four statements are fairly similar, and we establish them all at once. In each case, we identify with each node v in a structure a *discriminating set* D_v which is a finite set of traces available from v such that if u and v are not trace equivalent then $D_u \neq D_v$. It is easy to see 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. It can also be shown that in an arbitrary structure of rank n, two inequivalent nodes are distinguished by a trace of length at most 2^n. Thus, if $\mathcal{D}$ is the set of all discriminating sets, $\mathcal{D}$ is exponential in n in case 1; double exponential in cases 2 and 3; and treble exponential in case 4. This allows us to construct the labelling systems establishing the upper bounds.

For the lower bounds, note that, in the case of acyclic structures, for every set $D \in \mathcal{D}$ we can easily construct a rooted structure $\mathcal{K}_D, v$ of rank n such that its discriminating set is exactly D. For the case of structures with cycles, the construction is not as straightforward but, it can be shown that there is a polynomial p such that there is a collection of 2^{2^n} unary structures (and $2^{2^{2^n}}$ binary structures) of rank $p(n)$ with pairwise distinct discriminating sets. $\Box$

It follows that none of these 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, E. Grädel, and S. Kreutzer. Inflationary fixed points in modal logic. In *Proc. of the 10th Conf. on Computer Science Logic (CSL)*, volume 2142 of *LNCS*, pages 277–291. Springer, 2001. Full version at `http://www.cl.cam.ac.uk/~ad260/papers/mic.ps`.
4. 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.
5. 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.
6. M. Otto. Bisimulation-invariant Ptime and higher-dimensional mu-calculus. *Theoretical Computer Science*, 224:237–265, 1999.
7. J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. *Journal of Computer and System Sciences*, 53:10–25, 1996.
8. C. Stirling. *Modal and Temporal Properties of Processes*. Springer, 2001.
9. 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.
10. 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.
11. W. Thomas. Languages, automata and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.