Inflationary Fixed Points in Modal Logic

A. Dawar[1,⋆], E. Grädel[2], and S. Kreutzer[2]

[1] University of Cambridge Computer Laboratory, Cambridge CB2 3QG, UK.
anuj.dawar@cl.cam.ac.uk
[2] Mathematische Grundlagen der Informatik, RWTH Aachen, D-52065 Aachen.
{graedel,kreutzer}@informatik.rwth-aachen.de,
WWW: http://www-mgi.informatik.rwth-aachen.de

Abstract. We consider an extension of modal logic with an operator for constructing inflationary fixed points, just as the modal μ-calculus extends basic modal logic with an operator for least fixed points. Least and inflationary fixed point operators have been studied and compared in other contexts, particularly in finite model theory, where it is known that the logics IFP and LFP that result from adding such fixed point operators to first order logic have equal expressive power. As we show, the situation in modal logic is quite different, as the modal iteration calculus (MIC) we introduce has much greater expressive power than the μ-calculus. Greater expressive power comes at a cost: the calculus is algorithmically much less manageable.

1 Introduction

The modal μ-calculus L_μ is an extension of multi-modal logic with an operator for forming least fixed points. This logic has been extensively studied, having acquired importance for a number of reasons. In terms of expressive power, it subsumes a variety of modal and temporal logics used in verification, in particular LTL, CTL, CTL*, PDL and also many logics used in other areas of computer science, for instance description logics. On the other hand, L_μ has a rich theory, and is well-behaved in model-theoretic and algorithmic terms.

The logic L_μ is only one instance of a logic with an explicit operator for forming least fixed points. Indeed, in recent years, a number of fixed point extensions of first order logic have been studied in the context of finite model theory. It may be argued that fixed point logics play a central role in finite model theory, more important than first order logic itself. The best known of these fixed point logics is LFP, which extends first order logic with an operator for forming the least fixed points of positive formulae, defining monotone operators. In this sense, it relates to first order logic in much the same way as L_μ relates to propositional modal logic. However, a number of other fixed point operators have been extensively studied in finite model theory, including inflationary, partial, nondeterministic and alternating fixed points. All of these have in common that they allow the construction of fixed points of operators that are not necessarily monotone.

⋆ Research supported by EPSRC grants GR/L69596 and GR/N23028.

Furthermore, a variety of fragments of the fixed point logics formed have been studied, such as existential and stratified fragments, bounded fixed point logics, transitive closure logic and varieties of Datalog. Thus, there is a rich theory of the structure and expressive power of fixed point logics on finite relational structures and, to a lesser extent, on infinite structures.

In the present paper, we take a first step in the study of extensions of propositional modal logic by operators that allow us to form fixed points of non-monotone formulae. We focus on the simplest of these, that is the inflationary fixed point (also sometimes called the iterative fixed point). Though the inflationary fixed point extension of first order logic (IFP) is often used interchangeably with LFP, as the two have the same expressive power on finite structures, we show that in the context of modal logic, the inflationary fixed point behaves quite differently from the least fixed point.

Least and Inflationary Inductions. We begin by reviewing the known results on the logics LFP and IFP.

(1) On finite structures, LFP and IFP have the same expressive power [12].
(2) It is conjectured that IFP is strictly more expressive than LFP on infinite structures, but only partial results are known. On many interesting infinite structures, for instance in arithmetic $(\omega, +, \cdot)$, LFP and IFP are known to be equally expressive, but the translation of IFP into LFP can make the formulae much more complicated [6].
(3) On ordered finite structures, LFP and IFP express precisely the properties that are decidable in polynomial time.
(4) Simultaneous least or inflationary inductions do not provide more expressive power than simple inductions.
(5) The complexity of evaluating a formula ψ in LFP or IFP on a given finite structure $\mathfrak{A}$ is polynomial in the size of the structure, but exponential in the length of the formula. For formulae with a bounded number k of variables, the evaluation problem is PSPACE-complete [9], even for $k = 2$ and on fixed (and very small) structures. If, in addition to bounding the number of variables one also forbids parameters in fixed point formulae, the evaluation problem for LFP is computationally equivalent to the model checking problem for L_μ [11, 17] which is known to be in NP $\cap$ co-NP, in fact in UP $\cap$ co-UP [14], and hard for PTIME. It is an open problem whether this problem can be solved in polynomial time. The model checking problem for bounded variable IFP does not appear to have been studied previously.

We also note that even though IFP does not provide more expressive power than LFP on finite structures, it is often more convenient to use inflationary inductions in explicit constructions. The advantage of using IFP is that one is not restricted to inductions over positive formulae. A non-trivial case in point is the formula defining an order on the k-variable types in a finite structure, an essential ingredient of the proof of the Abiteboul-Vianu Theorem, saying that least and partial fixed point logics coincide if and only if PTIME = PSPACE (see

[4, 8, 10]). Furthermore, IFP is more robust, in the sense that inflationary fixed points are well-defined, even when other, non-monotone, operators are added to the language (see, for instance, [7]).

Inflationary Inductions in Modal Logic. Given the close relationship between LFP and IFP on finite structures, and the importance of the μ-calculus, it is natural to study also the properties and expressive power of inflationary fixed points in modal logic. In this paper, we undertake a study of an analogue of IFP for modal logic. We define a modal iteration calculus, MIC, by extending basic multi-modal logic with simultaneous inflationary inductions. While deferring formal definitions until Section 2, we begin with an informal explanation.

In L_μ, we can write formulae $\mu X.\varphi$, which are true in state s of a transition system $\mathcal{K}$ if, and only if, s is in the least set X satisfying $X \leftrightarrow \varphi$ in $\mathcal{K}$. We can do this, provided that the variable X appears only positively in φ. This guarantees that φ defines a monotone operator and has a least fixed point. Moreover, the fixed point can be obtained by an iterative process. Starting with the empty set, if we repeatedly apply the operator defined by φ (possibly through a transfinite series of stages), we obtain an increasing sequence of sets, which converges to the desired least fixed point. If, on the other hand, φ is not positive in X, we can still define an increasing sequence of sets, by starting with the empty set, and iteratively taking the union of the current set X with the set of states satisfying $\varphi(X)$, and this sequence must eventually converge to a fixed point (not necessarily of φ, but of the operator that maps X to $X \vee \varphi(X)$). More generally, we allow formulae $\mathbf{ifp}\ X_i : [X_1 \leftarrow \varphi_1, \ldots, X_k \leftarrow \varphi_k]$ that construct sets by a simultaneous inflationary induction. At each stage α, we have a tuple of sets $X_1^\alpha, \ldots, X_k^\alpha$. Substituting these into the formulae $\varphi_1, \ldots, \varphi_k$ we obtain a new tuple of sets, which we *add* to the existing sets $X_1^\alpha, \ldots, X_k^\alpha$, to obtain the next stage.

It is clear that MIC is a modal logic in the sense that it is invariant under bisimulation. In fact, on every class of bounded cardinality, inflationary fixed points can be unwound to obtain equivalent infinitary modal formulae. As a consequence, MIC has the tree model property. It is also clear that MIC is at least as expressive as L_μ. The following natural questions now arise.

(1) Is MIC more expressive than L_μ?
(2) Does MIC have the finite model property?
(3) What are the algorithmic properties of MIC? Is the satisfiability problem decidable? Can model checking be performed efficiently (as efficiently as for L_μ)?
(4) Can we eliminate, as in the μ-calculus and as in IFP, simultaneous inductions without losing expressive power?
(5) What is the relationship of MIC with monadic second-order logic and with finite automata? Or more generally, what are the 'right' automata for MIC?
(6) Is MIC the bisimulation-invariant fragment of any natural logic (as L_μ is the bisimulation-invariant fragment of MSO [13])?

We provide answers to most of these questions. From an algorithmic point of view, most of the answers are negative. From the point of view of expressiveness, we can say that in the context of modal logic, inflationary fixed points provide much more expressive power than least fixed points, and MIC has very different structural properties to L_μ. In particular, we establish the following results:

(1) There exist MIC-definable languages that are not regular. Hence MIC is more expressive than the μ-calculus, and does not translate to monadic second-order logic.
(2) MIC does not have the finite model property.
(3) The satisfiability problem for MIC is undecidable. In fact, it is not even in the arithmetic hierarchy.
(4) The model checking problem for MIC is PSPACE-complete.
(5) Simultaneous inflationary inductions do provide more expressive power than simple inflationary inductions. Nevertheless the algorithmic intractability results for MIC apply also to MIC without simultaneous inductions.
(6) There are bisimulation-invariant polynomial time properties that are not expressible in MIC.
(7) All languages in DTIME$(O(n))$ are MIC-definable.

No doubt, these properties exclude MIC as a candidate logic for hardware verification. On the other hand, the present study is an investigation into the structure of the inflationary fixed point operator and may suggest tractable fragments of the logic MIC, which involve crucial use of an inflationary operator, just as logics like CTL and alternation-free L_μ carve out efficiently tractable fragments of L_μ. In any case, it delineates the differences between inflationary and least fixed point constructs in the context of modal logic

In the rest of this paper, we begin in Section 2 by giving the necessary background on modal logic and fixed points, and giving the definition of MIC, along with an example that illustrates how this calculus has higher expressive power than L_μ. Section 3 establishes that MIC fails to have the finite model property and that the satisfiability problem is highly undecidable. This is established separately for MIC, and 1MIC, its fragment without simultaneous inductions. We also show that MIC is more expressive than 1MIC. In Section 5 we investigate questions of the computational complexity of MIC in the context of finite transitions systems. We show that the model checking problem is PSPACE-complete, that the class of models of any MIC formula is decidable in both polynomial time and linear space, and that there are polynomial time bisimulation-invariant properties that are not expressible in MIC. Finally, Section 6 investigates the expressive power of MIC on finite words, establishing that there are languages definable in MIC that are not context-free, and that every linear time decidable language is expressible in MIC.

Due to space limitations we only sketch the proofs of some results and defer the details to the full version of the paper [5].

2 The Modal Iteration Calculus

Before we define the modal iteration calculus, we briefly recall the definitions of propositional modal logic ML and the μ-calculus L_μ.

2.1 Propositional Modal Logic.

Transition Systems. Modal logics are interpreted on transition systems (also called Kripke structures). Fix a set A of actions and a set $\mathcal{V}$ of atomic propositions. A transition system for A and $\mathcal{V}$ is a structure $\mathcal{K}$ with universe V (whose elements are called states) binary relations $E_a \subseteq V \times V$ for each $a \in A$ and monadic relations $p \subseteq V$ for each atomic proposition $p \in \mathcal{V}$ (we do not distinguish notationally between atomic propositions and their interpretations.)

Syntax of ML. For a set A of actions and a set $\mathcal{V}$ of proposition variables, the formulae of ML are built from *false*, *true* and the variables $p \in \mathcal{V}$ by means of Boolean connectives $\wedge$, $\vee$, $\neg$ and modal operators $\langle a\rangle$ and $[a]$. That is, if ψ is a formula of ML and $a \in A$ is an action, then $\langle a\rangle\psi$ and $[a]\psi$ are also formulae of ML. If there is only one action in A, one simply writes $\Box$ and $\Diamond$ for $[a]$ and $\langle a\rangle$, respectively.

Semantics of ML. The formulae of ML are evaluated on transition systems at a particular state. Given a formula ψ and a transition system $\mathcal{K}$ with state v, we write $\mathcal{K}, v \models \psi$ to denote that the formula ψ holds in $\mathcal{K}$ at state v. We also write $[\![\psi]\!]^{\mathcal{K}}$ to denote the set of states v, such that $\mathcal{K}, v \models \psi$. In the case of atomic propositions, $\psi = p$, we have $[\![p]\!]^{\mathcal{K}} = p$. Boolean connectives are treated in the natural way. Finally for the semantics of the modal operators we put

$$[\![\langle a\rangle\psi]\!]^{\mathcal{K}} := \{v : \text{ there exists a state } w \text{ such that } (v,w) \in E_a \text{ and } w \in [\![\psi]\!]^{\mathcal{K}}\}$$
$$[\![[a]\psi]\!]^{\mathcal{K}} := \{v : \text{ for all } w \text{ such that } (v,w) \in E_a, \text{ we have } w \in [\![\psi]\!]^{\mathcal{K}}\}.$$

Hence $\langle a\rangle$ and $[a]$ can be viewed as existential and universal quantifiers 'along a-transitions'.

2.2 The μ-calculus L_μ.

Syntax of L_μ. The μ-calculus extends propositional modal logic ML by the following rule for building fixed point formulae: if ψ is a formula in L_μ and X is a propositional variable that occurs only positively in ψ, then $\mu X.\psi$ and $\nu X.\psi$ are L_μ formulae.

Semantics of L_μ. A formula $\psi(X)$ with a propositional variable X defines on every transition system $\mathcal{K}$ (with state set V, and with interpretations for free variables other than X occurring in ψ) an operator $\psi^{\mathcal{K}} : \mathcal{P}(V) \to \mathcal{P}(V)$ assigning to every set $X \subseteq V$ the set $\psi^{\mathcal{K}}(X) := [\![\psi]\!]^{\mathcal{K},X} = \{v \in V : (\mathcal{K}, X), v \models \psi\}$.

As X occurs only positively in ψ, the operator $\psi^{\mathcal{K}}$ is *monotone* for every $\mathcal{K}$, and therefore, by a well-known theorem due to Knaster and Tarski, has a least fixed point $\mathbf{lfp}(\psi^{\mathcal{K}})$ and a greatest fixed point $\mathbf{gfp}(\psi^{\mathcal{K}})$. Now we put $[\![\mu X.\psi]\!]^{\mathcal{K}} := \mathbf{lfp}(\psi^{\mathcal{K}})$ and $[\![\nu X.\psi]\!]^{\mathcal{K}} := \mathbf{gfp}(\psi^{\mathcal{K}})$.

Least (and greatest) fixed points can also be constructed inductively. Given a formula $\mu X.\psi(X)$, we define for each ordinal α, the stage X^α of the **lfp**-induction of $\psi^{\mathcal{K}}$ by $X^0 := \emptyset$, $X^{\alpha+1} := [\![\psi]\!]^{(\mathcal{K},X^\alpha)}$, and $X^\alpha := \bigcup_{\beta<\alpha} X^\beta$ if α is a limit ordinal.

By monotonicity, the stages of the **lfp**-induction increase until a fixed point is reached. The first ordinal at which this happens is called the *closure ordinal* of the induction. By ordinal induction, one easily proves that this inductively constructed fixed point coincides with the least fixed point. The cardinality of a closure ordinal cannot be larger than the cardinality of $\mathcal{K}$.

For any formula φ, the formula $\nu X.\varphi$ is equivalent to $\neg\mu X.\neg\varphi(\neg X)$, where $\varphi(\neg X)$ denotes the formula obtained from φ by replacing all occurrences of X with $\neg X$.

Simultaneous Fixed Points. There is a variant of L_μ that admits systems of simultaneous fixed points. Here one associates with any tuple $\overline{\psi} = (\psi_1, \dots, \psi_k)$ of formulae $\psi_i(\overline{X}) = \psi_i(X_1, \dots, X_k)$, in which all occurrences of all X_i are positive, a new formula $\varphi = \mu\overline{X}.\overline{\psi}$. The semantics of φ is induced by the least fixed point of the monotone operator $\psi^{\mathcal{K}}$ mapping $\overline{X}$ to $\overline{X}'$ where $X'_i = \{v \in V : (\mathcal{K}, \overline{X}), v \models \psi_i\}$. More precisely, $\mathcal{K}, v \models \varphi$ iff v is an element of the first component of the least fixed point of the above operator. Although these systems are computationally beneficial and sometimes also allow for more straightforward formalisations, they do not increase the expressive power. It is known that simultaneous least fixed points can be eliminated in favour of nested individual fixed points (see e.g. [1, page 27]). Indeed, $\mu XY.[\psi(X,Y), \varphi(X,Y)]$ is equivalent to $\mu X.\psi(X, \mu Y.\varphi(X,Y))$, and this equivalence generalises to larger systems in the obvious way.

Bisimulations and Tree Model Property. Bisimulation is a notion of behavioural equivalence for transition systems. No reasonable modal logic can distinguish between two systems that are bisimulation equivalent. Formally, given two transition systems $\mathcal{K}$ and $\mathcal{K}'$, 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 $p \in \mathcal{V}$ and each $(u, u') \in R$, $u \in [\![p]\!]^{\mathcal{K}}$ if, and only if, $u' \in [\![p]\!]^{\mathcal{K}'}$; (3) for each $(u, u') \in R$, and each $t \in V$ such that $(u, t) \in E_a$, there is a $t' \in V'$ with $(u', t') \in E'_a$ and $(t, t') \in R$; and (4) for each $(u, u') \in R$, and each $t' \in V'$ such that $(u', t') \in E'_a$, there is a $t \in V$ with $(u, t) \in E_a$ and $(t, t) \in R$.

Bisimulation equivalence corresponds to equivalence in an infinitary modal logic ML^∞ [2]. This logic is the closure of ML under disjunctions and conjunctions taken over arbitrary sets of formulae. Thus, if S is any set (possibly infinite) of formulae, then $\bigwedge S$ and $\bigvee S$ are also formulae of ML. It can be shown that for any transition systems $\mathcal{K}$ and $\mathcal{K}'$, $\mathcal{K}, v \sim \mathcal{K}', v'$ if, and only if, $\mathcal{K}, v$ makes true exactly the same formulae of ML^∞ as $\mathcal{K}', v'$.

A transition system is called a *tree*, if for every state v, there is at most one state u, and at most one action a such that $(u, v) \in E_a$ and there is exactly one state r, called the *root* of the tree, for which there is no state having a transition to r, and if every state is reachable from the root. It is known that for every transition system $\mathcal{K}$, and any state v, there is a tree $\mathcal{T}$ with root r such that $\mathcal{K}, v \sim \mathcal{T}, r$. One consequence of this is that any logic that respects bisimulation has the tree model property. For instance, for any formula φ of L_μ, if φ is satisfiable, then there is a tree $\mathcal{T}$ such that $\mathcal{T}, r \models \varphi$.

2.3 The Modal Iteration Calculus.

We are now ready to introduce MIC. Informally, MIC is propositional modal logic ML, augmented with simultaneous inflationary fixed points.

Definition 2.1 (Syntax and semantics of MIC**).** MIC *extends propositional multi-modal logic by the following rule: if $\varphi_1, \ldots, \varphi_k$ are formulae of* MIC*, and $X_1, \ldots, X_k$ are propositional variables, then*

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

is a system *of rules, and* $(\mathbf{ifp}\ X_i : S)$ *is a formula of* MIC*. If S consists of a single rule $X \leftarrow \varphi$ we simplify the notation and write* $(\mathbf{ifp}\ X \leftarrow \varphi)$ *instead of* $(\mathbf{ifp}\ X : X \leftarrow \varphi)$.

Semantics: On every Kripke structure $\mathcal{K}$, the system S defines, for each ordinal α, a tuple $\overline{X}^\alpha = (X_1^\alpha, \ldots, X_k^\alpha)$ of sets of states, via the following inflationary induction (for $i = 1, \ldots, k$).

$$\begin{aligned} X_i^0 &:= \emptyset, \\ X_i^{\alpha+1} &:= X_i^\alpha \cup [\![\varphi_i]\!]^{(\mathcal{K}, \overline{X}^\alpha)}, \\ X_i^\alpha &:= \bigcup_{\beta<\alpha} X_i^\beta \ \text{ if } \alpha \text{ is a limit ordinal.} \end{aligned}$$

We call $(X_1^\alpha, \ldots, X_k^\alpha)$ the stage α of the inflationary induction of S on $\mathcal{K}$. As the stages are increasing (i.e. $X_i^\alpha \subseteq X_i^\beta$ for any $\alpha < \beta$), this induction reaches a fixed point $(X_1^\infty, \ldots, X_k^\infty)$. Now we put $[\![(\mathbf{ifp}\ X_i : S)]\!]^{\mathcal{K}} := X_i^\infty$.

See Section 2.4 and 3 for examples of such formulae.

Lemma 2.2. $L_\mu \subseteq$ MIC*. Further, on every class of structures of bounded cardinality* MIC $\subseteq$ ML$^\infty$.

Proof. Clearly, if X occurs only positively in ψ, then $\mu X.\psi \equiv \mathbf{ifp}\ X \leftarrow \psi$. Hence $L_\mu \subseteq$ MIC.

Now, let S be a system of rules $X_i \leftarrow \varphi_i(X_1, \dots, X_k)$. It is clear that for each ordinal α there exist formulae $\varphi_1^\alpha, \dots, \varphi_k^\alpha \in \mathrm{ML}^\infty$ defining, over any Kripke structure, the stage α of the induction by S. As closure ordinals are bounded on structures of bounded cardinality, the second claim follows. $\square$

Corollary 2.3. MIC *is invariant under bisimulation and has the tree model property.*

Note that on structures of unbounded cardinality, L_μ and MIC are not contained in ML^∞. For instance, well-foundedness is expressed by the L_μ-formula $\mu X.\Box X$, but is known not to be expressible in ML^∞.

2.4 Non-Regular Languages

We now demonstrate that MIC is strictly more expressive than L_μ. Recall that every formula of L_μ can be translated into a formula of monadic second order logic (MSO). Moreover, it is known [3] that the only sets of finite words that are expressible in MSO are the regular languages. For our purposes, a finite word is a transition system with only one kind of action, which is a finite tree, and where every state has at most one successor.

Proposition 2.4. *There is a language that is expressible in* MIC *but not in* MSO.

Proof. The language $L := \{a^n b^m : n \leq m\}$ is not regular, hence not definable in monadic second-order logic, but it is definable in MIC. To see this, we consider first the formula $\pi(X) = (\mathbf{ifp}\ Y \leftarrow \Diamond(b \wedge \neg X) \vee \Diamond(a \wedge X \wedge Y))$ which (since the rule is positive in Y) is in fact equivalent to a L_μ-formula. On every word $w = w_0 \cdots w_{n-1} \in \{a, b\}^*$ and $X \subseteq \{0, \dots, n-1\}$, the formula is true if w starts with a (possibly empty) a-sequence inside X followed by a b outside X. Now the formula $(\mathbf{ifp}\ X \leftarrow (a \wedge \pi(X)) \vee (b \wedge \Box X))$ defines (inside a^*b^*) the language L. Note that the language a^*b^* is definable in L_μ, so we can conjoin this definition to the above formula to obtain a definition of L which works on all words in $\{a, b\}^*$ $\square$

The observation in Proposition 2.4 was pointed out to us in discussion by Martin Otto, and was the starting point of the investigation reported here.

3 Interpreting Arithmetic in MIC

In this section we prove that the satisfiability problem of MIC is undecidable in a very strong sense. Given that MIC is invariant under bisimulation, we can restrict attention to trees. In fact we will only consider well-founded trees (i.e. trees satisfying the formula $\mathbf{ifp}\ X \leftarrow \Box X$). The *height* $h(v)$ of a node v in a well-founded tree $\mathcal{T}$ is an ordinal, namely the least upper bound of the heights of its children. For any node v in a tree $\mathcal{T}$, we write $\mathcal{T}(v)$ for the subtree of $\mathcal{T}$ with root v. We first show that the nodes of finite height and the nodes of height ω are definable in MIC.

Lemma 3.1. *Let S be the system*

$$\begin{aligned} X &\leftarrow \Box \mathit{false} \vee (\Box X \wedge \Diamond \neg Y) \\ Y &\leftarrow X. \end{aligned}$$

Then, on every tree $\mathcal{T}$, $[\![\mathbf{ifp}\ X : S]\!]^{\mathcal{T}} = [\![\mathbf{ifp}\ Y : S]\!]^{\mathcal{T}} = \{v : h(v) < \omega\}$.

Proof. By induction we see that for each $i < \omega$, $X^i = \{v : h(v) < i\}$ and $Y^i = X^{i-1} = \{v : h(v) < i-1\}$. As a consequence $X^\omega = Y^\omega = \{v : h(v) < \omega\}$. One further iteration shows that $X^{\omega+1} = Y^{\omega+1} = X^\omega$. $\Box$

With the system S exhibited in Lemma 3.1 we obtain the formulae finite-height $:= (\mathbf{ifp}\ X : S)$ and ω-height $:= \neg(\mathbf{ifp}\ X : S) \wedge \Box(\mathbf{ifp}\ X : S)$ which define, respectively, the nodes of finite height and the nodes of height ω. Note that ω-height is a satisfiable formula all of whose models are infinite.

Proposition 3.2. MIC *does not have the finite model property.*

We show that the satisfiability problem of MIC is undecidable. In fact MIC interprets full arithmetic on the heights of nodes. To prove this we first define some auxiliary formulae that will be used frequently throughout the paper. We always assume that the underlying structure is a well-founded tree.

- The formula nonempty$(\varphi) := (\mathbf{ifp}\ X \leftarrow \varphi \vee \Diamond X)$ expresses that φ holds somewhere in the subtree of the current node: $\mathcal{T}, v \models$ nonempty(φ) iff $[\![\varphi]\!]^{\mathcal{T}} \cap \mathcal{T}(v) \neq \emptyset$.
- Dually all$(\varphi) := (\mathbf{ifp}\ X \leftarrow \varphi \wedge \Box X)$ says that φ holds at all nodes of the subtree $\mathcal{T}(v)$.
- We say that a set X (in a tree $\mathcal{T}$) *encodes* the ordinal α if $X = \{v : h(v) < \alpha\}$. Let ordinal$(X)$ be the conjunction of the formula all$(X \to \Box X)$ with

  $$\begin{aligned} \neg(\mathbf{ifp}\ Z : Y &\leftarrow \Box Y \\ Z &\leftarrow \text{nonempty}(\neg Y \wedge \Box Y \wedge X) \wedge \text{nonempty}(\neg Y \wedge \Box Y \wedge \neg X)). \end{aligned}$$

  It expresses that X encodes some ordinal. Indeed all$(X \to \Box X)$ says that with each node $v \in X$, the entire subtree rooted at v is contained in X. The second conjunct performs an inflationary induction incorporating into Y at each stage $\beta + 1$ all nodes of height β (which satisfy $\neg Y \wedge \Box Y$) and incorporates the root of the tree into Z if both X and its complement contain nodes of height β. Hence, at the end of the induction the root of the tree will *not* be contained in Z if, and only if, X does not distinguish between nodes of the same height. Together the two conjuncts imply that X contains all nodes up to some height.
- The formula number$(X) =$ ordinal$(X) \wedge$ nonempty(finite-height $\wedge \neg X$) says that X encodes a natural number n (inside a tree of height $> n$).

Lemma 3.3. *Let $\mathcal{T}$ be a well-founded tree of height ω. There exist formulae* plus(S,T) *and* times(S,T) *of* MIC *such that, whenever the sets S and T encode in the tree $\mathcal{T}$ the natural numbers s and t, then* $[\![\text{plus}(S,T)]\!]^{\mathcal{T}}$ *encodes $s+t$, and* $[\![\text{times}(S,T)]\!]^{\mathcal{T}}$ *encodes st.*

Proof. Let

$$\begin{aligned}\text{plus}(S,T) := \textbf{ifp}\ Y : X &\leftarrow \Box X\\ Y &\leftarrow S \vee (\Box Y \wedge \text{nonempty}(X) \wedge \text{all}(X \rightarrow T)).\end{aligned}$$

Obviously at each stage n, we have $X^n = \{v : h(v) < n\}$. We claim that for each n, $Y^{n+1} = \{v : h(v) < s + \min(n,t)\}$. For $n = 0$ this is clear (note that for the case $s = 0$ this is true because the conjunct nonempty(X) prevents the Y-rule from being active at stage 1). For $n > 0$ the inclusion $X^n \subseteq T$ is true iff $n \leq t$. Hence we have $Y^{n+1} = \{v : h(v) < s + n\}$ in the case that $n \leq t$ and $Y^{n+1} = Y^n = \cdots = Y^t$ otherwise. To express multiplication we define

$$\begin{aligned}\text{times}(S,T) := \textbf{ifp}\ Y : X &\leftarrow \Box X\\ Y &\leftarrow \text{plus}(Y,S) \wedge \text{all}(\Box X \rightarrow T).\end{aligned}$$

We claim that $Y^n = \{v : h(v) < s \cdot \min(n,t)\}$. This is trivially true for $n = 0$. If it is true for $n < t$, then $Y^{n+1} = \{v : h(v) < sn + s\} = \{v : h(v) < s(n+1)\}$. Finally for $n \geq t$, the extension of $\Box X^n$ is $\{v : h(v) < n + 1\}$ which is not contained in $T = \{v : h(v) < t\}$, hence $Y^{n+1} = Y^n = \cdots = Y^t$. □

Corollary 3.4. *For every polynomial $f(x_1, \ldots, x_r)$ with coefficients in the natural numbers there exists a formula $\psi_f(X_1, \ldots, X_r) \in$ MIC such that for every tree $\mathcal{T}$ of height ω and all sets $S_1, \ldots, S_r$ encoding numbers $s_1, \ldots, s_r \in \omega$*

$$[\![\psi_f(S_1, \ldots, S_r)]\!]^{\mathcal{T}} = \{v : h(v) < f(s_1, \ldots, s_r)\}.$$

Proof. By induction on f.

- $\psi_0 := \mathit{false}$.
- $\psi_1 := \Box \mathit{false}$.
- $\psi_X := X$.
- $\psi_{f+g} := \text{plus}[S/\psi_f, T/\psi_g]$, i.e. the formula obtained by replacing in plus(S,T) the variables S and T by, respectively, ψ_f and ψ_g.
- $\psi_{f \cdot g} := \text{times}[S/\psi_f, T/\psi_g]$.

□

Theorem 3.5. *For every first order sentence ψ in the vocabulary $\{+, \cdot, 0, 1\}$ of arithmetic, there exists a formula $\psi^* \in$ MIC such that ψ is true in the standard model $(\mathbb{N}, +, \cdot, 0, 1)$ of arithmetic if and only if ψ^* is satisfiable.*

Proof. We have already seen that there exists a MIC-axiom ω-height axiomatising the models that are bisimilar to a tree of height ω. Further, we can express set equalities $X = Y$ by all$(X \leftrightarrow Y)$ and we know how to represent polynomials by MIC-formulae. What remains is to translate quantifiers.

More precisely, we need to show that for each first order formula $\psi(y_1, \ldots, y_r)$ in the language of arithmetic there exists a MIC-formula $\psi^*(Y_1, \ldots, Y_r)$ such that on rooted trees $\mathcal{T}, w$ of height ω and all sets $S_1, \ldots, S_r$ that encode numbers $s_1, \ldots, s_r$ on $\mathcal{T}$ we have that $(\mathbb{N}, +, \cdot, 0, 1) \models \psi(s_1, \ldots, s_r)$ iff $\mathcal{T}, w \models \psi^*(S_1, \ldots, S_r)$.

Only the case of formula of the form $\psi(\overline{y}) := \exists x \varphi(x, \overline{y})$ remains to be considered. By induction hypothesis, we assume that for $\varphi(x, \overline{y})$ the corresponding MIC-formula $\varphi^*(X, \overline{Y})$ has already been constructed. Now let

$$\begin{aligned}\psi^*(\overline{Y}) := \mathbf{ifp}\ Z :\ & X \leftarrow \Box X \\ & Z \leftarrow \varphi^*(X, \overline{Y}) \wedge \mathrm{number}(X).\end{aligned}$$

□

Corollary 3.6. *The satisfiability problem for* MIC *is undecidable. In fact, it is not even in the arithmetical hierarchy.*

The proof given above appears to rely crucially on the use of simultaneous inductions. Indeed, one can show that formulae of MIC involving simultaneous inductions, in particular the formula constructed in the proof of Lemma 3.1, cannot be expressed without simultaneous inductions (see Theorem 4.2). However, it is still the case that first order arithmetic can be reduced to the satisfiability problem for MIC without simultaneous inductions. (See [5] for details.)

4 Simultaneous vs. Non-Simultaneous Inductions

It is easy to see that the equivalence $\mu XY.(\psi, \varphi) \equiv \mu X.\psi(X, \mu Y.\varphi(X,Y))$ (sometimes called the Bekic-principle [1]) fails in both directions when we take inflationary instead of least fixed points. However, it still is conceivable that simultaneous inductions could be eliminated by more complicated techniques. It follows from the results below, that this is not the case, i.e. simultaneous inflationary inductions provide more expressive power than simple ones. Let 1MIC denote the fragment of MIC that does not involve simultaneous inductions.

For any ordinal α, let $\mathcal{T}_\alpha$ denote the tree with a root v_α that has a set $\{v_\beta \mid \beta < \alpha\}$ of children indexed by ordinals less than α, where each v_β is the root of a subtree isomorphic to $\mathcal{T}_\beta$.

Lemma 4.1. *Let $\varphi \in$ 1MIC be a formula. If $X_1, \ldots, X_k$ are atomic propositions on $\mathcal{T}_\omega$, closed under bisimulations, such that $v_\omega \notin X_i$ (where v_ω is the root of $\mathcal{T}_\omega$) and $\mathcal{T}_\omega, v_\omega \models \varphi(X_1, \ldots, X_k)$, then there is a finite N such that for all $n > N$ and all nodes v_n of height n, $\mathcal{T}_\omega, v_n \models \varphi(X_1 - \{v_n\}, \ldots, X_k - \{v_n\})$.*

It is a straightforward consequence of this lemma that the formula ω-height defined in Sect. 3 is not equivalent to any formula of 1MIC. We hence have established the following separation result.

Theorem 4.2. MIC *is strictly more powerful than* 1MIC.

5 The Model Checking Problem for MIC

Recall that the model checking problem for the μ-calculus is in UP $\cap$ co-UP, and is conjectured by some to be solvable in polynomial time. We now show that MIC is algorithmically more complicated (unless PSPACE = NP).

We first observe that the naive bottom-up evaluation algorithm for MIC-formulae uses polynomial time with respect to the size of the input structure, and polynomial space (and exponential time) with respect to the length of the formula. Let $\mathcal{K}$ be a transition system with n nodes and m edges. The size $||\mathcal{K}||$ of appropriate encodings of $\mathcal{K}$ as an input for a model checking algorithm is $O(n+m)$. It is well known that the extension $[\![\varphi]\!]^{\mathcal{K}}$ of a basic modal formula φ (without fixed points) on a finite transition system $\mathcal{K}$ can be computed in time $O(|\varphi| \cdot ||\mathcal{K}||)$. Further, any inflationary induction $\mathbf{ifp}\ X_i : [X_1 \leftarrow \varphi_1, \ldots, X_k \leftarrow \varphi_k]$ reaches a fixed point on $\mathcal{K}$ after at most kn iterations. Hence, the bottom-up evaluation of a MIC-formula ψ with d nested simultaneous inflationary fixed points, each of width k, on $\mathcal{K}$ needs at most $O((kn)^d)$ basic evaluation steps. For each fixed point variable occurring in the formula, $2n$ bits of workspace are needed to record the current value and the last value of the induction. This gives the following complexity results.

Proposition 5.1. *Any* MIC *formula ψ of nesting depth d and simultaneous inductions of width at most k on a transition system $\mathcal{K}$ with n nodes can be evaluated in time $O((kn)^d|\psi| \cdot ||\mathcal{K}||)$ and space $O(|\psi| \cdot n)$.*

In terms of common complexity classes the results can be stated as follows.

Theorem 5.2. *(1) The combined complexity of the model checking problem for* MIC *on finite structures is in* PSPACE.

(2) For any fixed formula $\psi \in$ MIC*, the model checking problem for ψ on finite structures is solvable in polynomial time and linear space.*

We now show that, contrary to the case of the μ-calculus, the complexity results obtained by this naive algorithm cannot be improved essentially.

Theorem 5.3. *There exist transitions systems $\mathcal{K}$, such that the model checking problem for* MIC *on $\mathcal{K}$ is* PSPACE*-complete (even for* 1MIC*).*

The proof is by reduction from QBF (the evaluation problem for quantified Boolean formulae.) We only sketch the argument here.

Let $\mathcal{K}$ be the Kripke-structure consisting of two points $0, 1$, the atomic proposition $p = \{1\}$, and the complete transition relation $\{0,1\} \times \{0,1\}$. Let $\alpha(X) := \neg X \land (p \to \Diamond X)$. Further, let $\varphi[X/\alpha(X)]$ denote the formula obtained from φ by replacing every free occurrence of X by $\alpha(X)$.

We inductively associate with every quantified Boolean formula ψ a MIC-formula ψ^* as follows. For $\psi := X$ we set $\psi^* := (p \land X) \lor (\neg p \land \Diamond X)$. Further, $(\neg\psi)^* := \neg\psi^*$ and $(\psi \circ \varphi)^* := \psi^* \circ \varphi^*$ for $\circ \in \{\land, \lor\}$. Finally, for $\psi := \forall X \varphi$ we put $\psi^* := \Box(\mathbf{ifp}\ X \leftarrow \alpha(X) \land \varphi^*[X/\alpha(X)])$.

It can be shown that for any closed QBF-formula ψ, we have $[\![\psi^*]\!]^{\mathcal{K}} = \{0,1\}$ if ψ is true and $[\![\psi^*]\!]^{\mathcal{K}} = \emptyset$ otherwise. The theorem now follows immediately.

In [15], Otto introduced a higher-dimensional μ-calculus, denoted L_μ^ω which extends basic multi-modal logic with an operator for forming least fixed points of arbitrary arity, rather than just sets. He showed that L_μ^ω can express every bisimulation-invariant, polynomial time decidable property of finite structures. Since we know that any collection of finite structures definable in MIC is both bisimulation-invariant and polynomial time decidable, it follows that every formula of MIC can be translated to a formula of L_μ^ω that is equivalent to it on finite structures. We now show that the converse fails. In particular, there are properties of finite trees that are bisimulation-invariant and polynomial time decidable but cannot be expressed in MIC.

Theorem 5.4. *There is a collection $\mathcal{F}$ of finite trees in* PTIME*, closed under bisimulation, which is not expressible in* MIC*.*

We sketch the proof. Define $\mathcal{F}$ to be the collection of all finite trees $\mathcal{T}$ such that all children of the root of $\mathcal{T}$ are bisimilar. As bisimuation equivalence is decidable in polynomial time, it follows that $\mathcal{F}$ is in PTIME. It is also obvious that $\mathcal{F}$ is closed under bisimulation.

Assume, towards a contradiction, that there is a formula $\varphi \in$ MIC that defines $\mathcal{F}$. We use φ to define an equivalence relation on trees. Informally $\mathcal{T}_1 \sim_\varphi \mathcal{T}_2$ if at all stages of all the **ifp**-inductions in φ, the same subformulae of φ become true in $\mathcal{T}_1$ as in $\mathcal{T}_2$. Now, it can be shown that the index of $\sim_\varphi$ on trees of height n is bounded by $2^{p(n)}$ (for some polynomial $p(n)$ depending only on φ), whereas the bisimulation-index on trees of height n is not bounded by any elementary function. Hence there exist $\mathcal{T}_1 \sim_\varphi \mathcal{T}_2$ with $\mathcal{T}_1 \not\sim \mathcal{T}_2$. It is easy to see that φ cannot distinguish between those trees where every child of the root is the root of a copy of $\mathcal{T}_1$ and those trees where one of these copies is replaced by $\mathcal{T}_2$. But in the first case the tree is in $\mathcal{F}$, and in the second it is not, yielding a contradiction.

6 Languages

In this section we investigate the expressive power of MIC on finite strings. In other words we attempt to determine what languages are definable by formulae of MIC. For our purposes, a word w of length n, in an alphabet Σ is a transition system with n states $v_1, \ldots, v_n$, a single action such that $(v_i, v_j) \in E$ if, and only if, $j = i + 1$ and an atomic proposition s for each $s \in \Sigma$, such that for each v_i, there is a unique s with $v_i \in s$.

We have already seen in Proposition 2.4, that there are non-regular languages that are definable in MIC. We begin this section by strengthening this result and showing that there are languages definable in MIC that are not even context-free.

6.1 Non-CFLs in MIC

Theorem 6.1. *There is a language definable in* MIC *that is not context-free*

Proof. Consider the language $L := \{cwdw \mid w \in \{a,b\}^*\}$ over the alphabet $\{a,b,c,d\}$. It is easily verified that L is not a context-free language. To see that it is definable in MIC, first note that the formula

$$\alpha := c \wedge \Box\text{empty}(c) \wedge \text{nonempty}(d) \wedge \text{all}(d \to \Box\text{empty}(d))$$

defines the set of strings $\{cxdy \mid x, y \in \{a,b\}^*\}$. Now, the desired formula is the conjunction of α with the negation of the formula

$$\varphi := \textbf{ifp } X \leftarrow [\neg c \wedge (\Box X \vee \Box d)] \vee [c \wedge \text{nonempty}(\psi)]$$

where, ψ is the formula

$$\begin{aligned}\neg X \wedge \Box X \wedge [(b \wedge \text{nonempty}(a \wedge \Box X \wedge \neg X)) \vee \\ (a \wedge \text{nonempty}(b \wedge \Box X \wedge \neg X))]\end{aligned}$$

$\Box$

We can also add the observation that the formula constructed in the proof of Theorem 6.1 above does not involve any simultaneous inductions, and therefore there are non-context-free languages definable in 1MIC.

Another measure of the complexity of a language, considered in [16] is *automaticity*. Briefly, the automaticity of a language L is the function A_L which gives for each n the number of states in the smallest deterministic automaton which accepts a language that agrees with L on all strings of length at most n. Clearly, every regular language has constant automaticity. Here, we note that it can be shown that the language used in the proof of Theorem 6.1 has exponential automaticity, which is worst possible.

Finally, to place the expressive power of MIC in the Chomsky hierarchy, we note that every language definable in MIC can be defined by a context-sensitive grammar. This follows from the observation made in Section 5 that any class of finite structures defined by a formula of MIC is decidable in linear space, and the result that all languages decidable by nondeterministic linear space machines are definable by context-sensitive grammars.

6.2 Capturing Linear Time Languages

We have seen in Section 5 that the data complexity of evaluating MIC-formulae is in polynomial time and linear space. It is also clear that MIC can express PTIME-complete properties, as this is already the case for the μ-calculus.

On words the situation is somewhat different. The μ-calculus defines precisely the regular languages and hence is very far away from expressing PTIME-complete properties. On the other side we have already seen that there exist MIC-definable languages that are not even context-free. We will now show that MIC can in fact define all languages that are decidable in linear time (by a Turing machine).

An observation that we will use in the proof, but which may well be of independent interest, is that cardinality comparisons and addition of cardinalities are expressible in MIC on words (recall that none of these are MSO-definable).

Lemma 6.2. *There exists a formula $\varphi(X,Y)$ of* MIC *such that on every word w, we have $w,X,Y \models \varphi$ if and only if $|X| = |Y|$. Similarly for $|X| < |Y|$ and $|X| + |Y| = |Z|$.*

Theorem 6.3. *Every language $L \in \mathrm{DTIME}(O(n))$ is* MIC*-definable.*

Note that we cannot expect to extend the result for linear time to quadratic time or higher. This is because, as we have seen, every language definable in MIC is decidable in linear space, and it is not expected that quadratic time is included in linear space.

References

1. A. Arnold and D. Niwinski. *Rudiments of μ-calculus*, North-Holland, 2001.
2. J. van Benthem. *Modal Logic and Classical Logic.* Bibliopolis, Napoli, 1983.
3. J. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.
4. A. Dawar. *Feasible Computation Through Model Theory.* PhD thesis, University of Pennsylvania, 1993.
5. A. Dawar, E. Grädel, and S. Kreutzer. *Inflationary Fixed Points in Modal Logics.* See "http://www.cl.cam.ac.uk/users/ad260/pubs.html" or "http://www-mgi.informatik.rwth-aachen.de/Publications/".
6. A. Dawar and Y. Gurevich. Fixed point logics. In preparation.
7. A. Dawar and L. Hella. The expressive power of finitely many generalized quantifiers. *Information and Computation*, 123:172–184, 1995.
8. A. Dawar, S. Lindell, and S. Weinstein. Infinitary logic and inductive definability over finite structures. *Information and Computation*, 119:160–175, 1994.
9. S. Dziembowski. Bounded-variable fixpoint queries are PSPACE-complete. In *10th Annual Conference on Computer Science Logic CSL 96. Selected papers*, volume 1258 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1996.
10. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 2nd edition, 1999.
11. E. Grädel and M. Otto. On logics with two variables. *Theoretical Computer Science*, 224:73–113, 1999.
12. Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.
13. 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*, number 1119 in Lecture Notes in Computer Science, pages 263–277. Springer-Verlag, 1996.
14. M. Jurdzinski. Deciding the winner in parity games is in UP $\cap$ Co-UP. *Information Processing Letters*, 68:119–124, 1998.
15. M. Otto. Bisimulation-invariant Ptime and higher-dimensional mu-calculus. *Theoretical Computer Science*, 224:237–265, 1999.
16. J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. *Journal of Computer and System Sciences*, 53:10–25, 1996.
17. M. Vardi. On the complexity of bounded-variable queries. In *Proc. 14th ACM Symp. on Principles of Database Systems*, pages 266–267, 1995.