The expressive power of two-variable least fixed-point logics

Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt

Institut für Informatik, Humboldt-Universität, Berlin
{grohe,kreutzer,schweika}@informatik.hu-berlin.de

Abstract. The present paper gives a classification of the expressive power of two-variable least fixed-point logics. The main results are:
1. The two-variable fragment of monadic least fixed-point logic with parameters is as expressive as full monadic least fixed-point logic (on binary structures).
2. The two-variable fragment of *monadic* least fixed-point logic without parameters is as expressive as the two-variable fragment of *binary* least fixed-point logic without parameters.
3. The two-variable fragment of binary least fixed-point logic with parameters is strictly more expressive than the two-variable fragment of monadic least fixed-point logic with parameters (even on finite strings).

1. Introduction

In the fields of mathematical logic and finite model theory it has always been an important issue to compare the expressive power of different logics. Among the logics that received particular attention in theoretical computer science, extensions of first-order logic by mechanisms that allow to define relations *by induction* play a prominent role. Formalising such inductive definitions in a logical language usually involves some kind of fixed-point construction. In particular, *least fixed-point logic*, LFP, is the extension of first-order logic by least fixed-point operators, whereas M-LFP is the fragment of LFP where fixed-point operators are *monadic*, i.e., have arity at most 1.

From a well-known theorem due to Immerman and Vardi [13, 19] it is known that on ordered finite structures the logic LFP precisely characterises the complexity class PTIME. Apart from describing complexity classes, logics – and in particular fixed-point logics – are used, e.g., as languages for hardware and process specification and verification, and as query languages for expressing queries against databases. As observed, e.g., in [2, 18], the size of intermediate results that occur while evaluating a query (i.e., a logical formula) over a database (i.e., a finite structure) crucially depends on the number of first-order variables that occur in the formula. If the number of such variables is bounded by a constant k, the size of intermediate results remains polynomial in the size of the input structure. Therefore, the combined complexity of the *model checking problem* apparently is much smaller when considering the *k-variable fragment* of a logic instead of the entire logic.

The research community's considerable interest in bounded variable logics (cf., e.g., [3, 12, 18, 4, 6, 14, 16, 10, 9, 8]) can partly be explained by the comparably low combined complexity of the model checking problem for these logics. Further motivations for studying, in particular, *two*-variable logics are the decidability of FO^2 and the fact that *modal logics* can be embedded into two-variable logics. For example, plain modal logic ML can be embedded into the two-variable fragment of first-order logic, FO^2, the modal iteration calculus MIC [5] can be embedded into the two-variable fragment of monadic inflationary fixed-point logic, and the modal μ-calculus L_μ can be embedded into the two-variable fragment of monadic least fixed-point logic, $\mathrm{M\text{-}LFP}^2$, which, in turn, can be embedded into two-variable infinitary logic $L^2_{\infty\omega}$. In particular the logics FO^2 and $\mathrm{M\text{-}LFP}^2$ have received a lot of attention in the past (cf., [9, 8, 16]) in an attempt to explain the nice model-theoretical and computational properties of modal logics such as ML and L_μ. An overview of what is known about bounded variable logics and, in particular, *two*-variable logics, can be found in [8, 16].

The present paper's aim is to study the expressive power of two-variable fragments of least fixed-point logic LFP. As observed in [6, 9], defining these fragments requires some care, because allowing or forbidding the use of parameters (i.e., free first-order variables) in least fixed-point operators crucially changes the expressive power of the logic under consideration. In the literature, LFP^k usually refers to the parameter-free fragment of LFP where k first-order variables and second-order variables of arity at most k are available, cf. [8].
The logics we consider are

- $\mathrm{M\text{-}LFP}^2_{param}$ and LFP^2_{param}, the two-variable fragments of monadic and binary least fixed-point logic, respectively, where the use of parameters is allowed in fixed-point operators, and
- $\mathrm{M\text{-}LFP}^2$ and LFP^2, the two-variable fragments of monadic and binary least fixed-point logic, respectively, where least fixed-point operators are *not* allowed to have parameters.

We only consider fixed-point operators of arity at most two, since fixed-point definitions of higher arity already syntactically involve more than just two first-order variables. The presence of only two first-order variables furthermore renders it reasonable to restrict attention to *binary structures*, i.e., structures over a signature that consists of constant symbols and of relation symbols of arity at most two.

The logics M-LFP, $\mathrm{M\text{-}LFP}^2$ and $\mathrm{M\text{-}LFP}^2_{param}$, in particular, have explicitly been considered before [6, 8, 9, 17, 11]. E.g., $\mathrm{M\text{-}LFP}^2$ coincides with the logic called FP^2 in [8, 9] and $\widehat{\mathrm{FP}}^2$ in [6], whereas $\mathrm{M\text{-}LFP}^2_{param}$ coincides with the logic called FP^2 in [6].

It is known that on finite structures (or, more generally, on classes of structures of bounded cardinality), LFP^2 can be embedded into infinitary logic $L^2_{\infty\omega}$ [9, 16]. Furthermore, it has been observed in various places (cf., e.g., [16, 7]) that $\mathrm{M\text{-}LFP}^2_{param}$ can express the transitive closure of a binary relation and therefore cannot be embedded into $L^2_{\infty\omega}$. Consequently, one obtains that already on the class of finite graphs, $\mathrm{M\text{-}LFP}^2 \lneq \mathrm{M\text{-}LFP}^2_{param}$ and $\mathrm{M\text{-}LFP}^2_{param} \not\leq \mathrm{LFP}^2$. (Here we write $L \lneq L'$ to denote that a logic L is strictly less expressive than a logic L', and we write $L \not\leq L'$ to denote that there are problems that can be expressed in L, but not in L'.)

From [6] it is known that the combined complexity of the model checking problem for M-LFP^2_{param} and LFP^2_{param} is PSPACE-complete, whereas the combined complexity of the model checking problem for LFP^2 is closely related to that of the modal μ-calculus and therefore belongs to $\text{NP} \cap \text{Co-NP}$ and is PTIME-hard.

When restricting attention to the class of finite strings, one obtains an entirely different picture. Due to Büchi's theorem (cf., e.g., [7]), monadic second-order logic MSO can describe exactly the *regular* string languages which, in turn, can already be described by the modal μ-calculus L_μ. Consequently, on finite strings the logics MSO, M-LFP, M-LFP^2_{param}, and L_μ all have the same expressive power. Furthermore, it is known (cf., [9, 7]) that the two-variable fragment of monadic *inflationary* fixed-point logic, M-IFP^2, is capable of describing non-regular string-languages, and therefore $\text{M-LFP}^2 \lneq \text{M-IFP}^2$.

The present paper's contribution is to complete the picture of the expressive power of the two-variable fragments of least fixed-point logics. Our main results are

1. $\text{M-LFP}^2_{param} = \text{M-LFP}_{param}$ on binary structures. I.e. the two-variable fragment of monadic least fixed-point logic with parameters is as expressive as full monadic least fixed-point logic with parameters. Here, of course, the restriction to binary structures is crucial, as M-LFP contains full first-order logic.
2. $\text{LFP}^2 = \text{M-LFP}^2$, i.e., parameter-free two-variable *binary* least fixed-point logic has the same expressive power as parameter-free two-variable *monadic* least fixed-point logic.
3. $\text{M-LFP}^2_{param} \lneq \text{LFP}^2_{param}$, and the inclusion is strict already on the class of finite strings. We prove this result by showing that the non-regular string-language $\{a^n b^n \mid n \in \mathbb{N}\}$ is expressible in LFP^2_{param}.

Altogether this leads to the following inclusion structure of the expressive power of the two-variable fragments of least fixed-point logic:

$$\text{M-LFP}^2 \;=\; \text{LFP}^2 \;\lneq\; \text{M-LFP}^2_{param} \;=\; \text{M-LFP}_{param} \;\lneq\; \text{LFP}^2_{param}$$

on the class of binary structures,

and

$$\text{M-LFP}^2 \;=\; \text{LFP}^2 \;=\; \text{M-LFP}^2_{param} \;=\; \text{M-LFP}_{param} \;\lneq\; \text{LFP}^2_{param}$$

on the class of finite strings.

Fig. 1. Expressive power of the two-variable fragments of least fixed-point logic

The paper is organised as follows: After fixing some basic notation in section 2, we formally introduce the two-variable fragments of least fixed-point logic in section 3. The equivalence of M-LFP$^2_{param}$ and M-LFP$_{param}$ is proved in section 4. Afterwards, in section 5 we show that LFP^2 is equivalent to M-LFP^2 and that LFP$^2_{param}$ can express non-regular string-languages and therefore is strictly more expressive than M-LFP$^2_{param}$. Detailed proofs of the results presented here can be found in the full version of the paper.

2. Preliminaries

As usual, we write Ord for the class of ordinals and ω for the set of finite ordinals (i.e., non-negative integers). By Pow(S) we denote the power set of a set S. A *signature* is a finite set of relation and constant symbols. We call a signature τ *binary* if the arity of every relation symbol occurring in τ is at most two. Thus, structures of a binary signature are essentially coloured graphs.

In this paper we deal primarily with two-variable logics – logics that only allow for two distinct first-order variables. As with only two variables we cannot take advantage of relations of higher arity, we will only consider binary signatures throughout this paper. In most cases, this restriction has no impact on the validity of our statements. In the few places where it does, we will state this explicitly.

We use German letters $\mathfrak{A}, \mathfrak{B}, \dots$ to denote structures and the corresponding Roman letters $A, B, \dots$ to denote their universes.

We assume that the reader is familiar with *first-order logic* (FO). We use FO(τ) to denote the class of all first-order formulae of signature τ. Besides first-order variables we also allow free second-order variables in the formulae (but no second-order quantification). We write $\varphi(R_1, \dots, R_k, x_1, \dots, x_n)$ to indicate that the free first-order variables of the formula φ are $x_1, \dots, x_n$ and the free relation variables are $R_1, \dots, R_k$. We use $\bar{x}$ and $\bar{R}$ as abbreviations for sequences $x_1, \dots, x_n$ and $R_1, \dots, R_k$ of variables. Finally, we write $\varphi(R_1, \dots, R_k, \bar{x}_1, \dots, \bar{x}_k, \bar{z})$ to indicate that the free first-order variables of φ are the variables in the tuples $\bar{x}_i$ and $\bar{z}$ and that the arity of a tuple $\bar{x}_i$ is the same as the arity of the relation variable R_i.

3. Finite variable fragments of least fixed-point logic

In this section we give a brief introduction to least fixed-point logic and its two-variable fragments. For a detailed exposition see [7].

Least and greatest fixed points of monotone operators. Let τ be a signature and let $\varphi(R, \bar{x})$ be a formula of signature τ which is *positive* in the k-ary relation variable R, i.e. every atom of the form $R\bar{t}$ in φ occurs within the scope of an *even* number of negation symbols. φ defines for every τ-structure $\mathfrak{A}$ a monotone operator[1] $F_{\mathfrak{A},\varphi} : \mathrm{Pow}(A^k) \to \mathrm{Pow}(A^k)$ via $F_{\mathfrak{A},\varphi}(P) := \{\bar{a} \in A^k : (\mathfrak{A}, P) \models \varphi[\bar{a}]\}$, for every $P \subseteq A^k$. In cases where $\mathfrak{A}$ is understood from the context, we simply write F_φ for $F_{\mathfrak{A},\varphi}$.

A set P is called a *fixed point* (a *pre fixed point*) of φ in $\mathfrak{A}$ if, and only if, $F_{\mathfrak{A},\varphi}(P) = P$ ($F_{\mathfrak{A},\varphi}(P) \subseteq P$). P is called the *least fixed point* of φ if P is a fixed point of φ and $P \subseteq Q$ for every fixed point Q of F_φ. We write $\mathbf{lfp}(F_{\mathfrak{A},\varphi})$ for the least fixed point of $F_{\mathfrak{A},\varphi}$. Further, as the intersection of all pre fixed points is itself a fixed point, we get the following characterisation of least fixed points:

$$\mathbf{lfp}(F_{\mathfrak{A},\varphi}) \quad = \quad \bigcap\{Q : F_{\mathfrak{A},\varphi}(Q) = Q\} \quad = \quad \bigcap\{Q : F_{\mathfrak{A},\varphi}(Q) \subseteq Q\} \qquad (1)$$

There is also the corresponding notion of a *greatest fixed point* of φ which is the fixed point that contains all other fixed points. Least and greatest fixed points are dual to each other, in the sense that for every monotone operator $F : \mathrm{Pow}(M) \to \mathrm{Pow}(M)$ we have $\mathbf{lfp}(F) = \overline{\mathbf{gfp}(\overline{F})}$, where $\overline{F}$ is defined as $\overline{F}(U) := (F(U^c))^c$ (where U^c denotes the complement of U).

Least (and also greatest) fixed points of monotone operators can also be built up inductively. For this we define for all ordinals α sets $R^\alpha_{\mathfrak{A},\varphi} \subseteq A^k$ inductively as $R^0_{\mathfrak{A},\varphi} := \varnothing$, $R^{\alpha+1}_{\mathfrak{A},\varphi} := F_{\mathfrak{A},\varphi}(R^\alpha_{\mathfrak{A},\varphi})$, and $R^\lambda_{\mathfrak{A},\varphi} := \bigcup_{\gamma<\lambda} R^\gamma_{\mathfrak{A},\varphi}$ for infinite limit ordinals λ. In cases where $\mathfrak{A}$ and φ are understood, we simply write R^α. Since $F_{\mathfrak{A},\varphi}$ is monotone we have $R^\alpha \subseteq R^{\alpha+1}$ for all α. Hence the sequence $(R^\alpha)_{\alpha\in\mathrm{Ord}}$ eventually reaches a fixed point, i.e. there is an ordinal α such that $R^\alpha = R^{\alpha+1} = R^\gamma$ for all $\gamma > \alpha$. We refer to this fixed point as R^∞. It is easily seen that if the structure $\mathfrak{A}$ is finite then α is finite too. A theorem due to Knaster and Tarski establishes the equivalence $R^\infty_{\mathfrak{A},\varphi} = \mathbf{lfp}(F_{\mathfrak{A},\varphi})$ for all structures $\mathfrak{A}$ and formulae φ positive in the variable R. Thus, the sequence $(R^\alpha_{\mathfrak{A},\varphi})_{\alpha\in\mathrm{Ord}}$ approximates the least fixed point of $F_{\mathfrak{A},\varphi}$ from below. The sets $R^\alpha_{\mathfrak{A},\varphi}$ are called the *stages* of the least fixed-point induction on φ in $\mathfrak{A}$. A similar induction can be used to define greatest fixed points, where we start with $R^0 := A^k$ and take intersections instead of unions to define the higher stages.

Least fixed-point logic. The logic $\mathrm{LFP}(\tau)$ is the extension of $\mathrm{FO}(\tau)$ by least fixed-point operators. Precisely: $\mathrm{LFP}(\tau)$ contains $\mathrm{FO}(\tau)$ and is closed under Boolean connectives and first-order quantification; and if $\varphi(R, \bar{x}, \bar{z}, \bar{Q})$ is an $\mathrm{LFP}(\tau)$-formula which is positive in the k-ary relation variable R then for every k-tuple $\bar{t}$ of terms $[\mathbf{lfp}_{R,\bar{x}}\ \varphi](\bar{t})$ is an $\mathrm{LFP}(\tau)$-formula such that for every $\big(\tau \,\dot\cup\, \{\bar{z}, \bar{Q}\}\big)$-structure $\mathfrak{A}$ and every tuple $\bar{a} \in A^k$ we have $\mathfrak{A} \models [\mathbf{lfp}_{R,\bar{x}}\ \varphi](\bar{a})$ if, and only if, $\bar{a} \in \mathbf{lfp}(F_{\mathfrak{A},\varphi})$. Similarly, we allow formulae $[\mathbf{gfp}_{R,\bar{x}}\ \varphi](\bar{t})$ defining the greatest fixed point of $F_{\mathfrak{A},\varphi}$. The variables in $\bar{z}$ that are not contained in $\bar{x}$ are called the *parameters* of the fixed-point induction. They will play an important role in later sections.

Due to the above mentioned duality of least and greatest fixed points, **gfp**-operators can easily be replaced by **lfp**-operators with additional negation symbols. On the other hand, the use of **lfp**- *and* **gfp**-operators allows to transform every formula into a formula in *negation normal form*, i.e., a formula where negation symbols only occur directly in front of *atomic* sub-formulae.

We continue with the definition of some important fragments of least fixed-point logic. The *monadic least fixed-point logic* (M-LFP) is defined as the fragment of LFP where all fixed-point variables are unary, i.e. of arity ≤ 1. Analogously we define *binary least fixed-point logic* (Bin-LFP) as the fragment of LFP where all fixed-point variables are of arity ≤ 2.

[1] An operator $F : \mathrm{Pow}(M) \to \mathrm{Pow}(M)$ is *monotone* iff $F(A) \subseteq F(B)$ for all $A \subseteq B \subseteq M$.

We are primarily interested in fragments of M-LFP and Bin-LFP where the number of available first-order variables is restricted to two. Recall from above that the variables $\bar{z}$ occurring free in $\varphi(R, \bar{x}, \bar{z})$ other than $\bar{x}$ are called parameters of the fixed-point formula $[\mathbf{lfp}_{R,\bar{x}}\ \varphi](\bar{t})$. It is well know in finite model theory that parameters can be eliminated by increasing the arity of the fixed-point variables, i.e. for every LFP-formula $[\mathbf{lfp}_{R,\bar{x}}\ \varphi(R, \bar{x}, \bar{z})](\bar{t})$ there is an equivalent LFP-formula $[\mathbf{lfp}_{R',\bar{x}'}\ \varphi'(R', \bar{x}')](\bar{t}')$ which is parameter-free. However, this translation does not only require fixed-point variables of higher arity, it also requires the introduction of fresh first-order variables. Thus the standard translation of formulae with parameters into formulae without parameters does not apply to the two-variable fragments defined above. And indeed, as we will see later on, in this restricted setting, parameters increase the expressive power of the logics. We therefore introduce a separate notation for logics with and without parameters.

The logics M-LFP^2 and LFP^2 are defined as the parameter-free fragment of M-LFP and Bin-LFP resp. where only two distinct first-order variables may be used in the formulae. Analogously, the logics M-LFP$^2_{param}$ and LFP$^2_{param}$ are defined as the fragment of M-LFP and Bin-LFP resp. where only two distinct first-order variables may be used in the formulae but where the fixed-point operators may have parameters.

Simultaneous fixed-point inductions. Simultaneous inductions can simplify the formalisation of properties significantly, but as we will see below, they do not add to the expressive power of the logics.

Definition 1 (Simultaneous least fixed-point logic). Let $R_1, \dots, R_k$ be relation symbols of arity $r_1, \dots, r_k$, respectively. Simultaneous formulae are formulae of the form $\psi(\bar{x}) := [\mathbf{lfp}\ \ R_i : S](\bar{x})$, where

$$S := \begin{cases} R_1\bar{x}_1 \leftarrow \varphi_1(R_1, \dots, R_k, \bar{x}_1) \\ \qquad\vdots \\ R_k\bar{x}_k \leftarrow \varphi_k(R_1, \dots, R_k, \bar{x}_k) \end{cases}$$

is a system of LFP-formulae φ_i which are positive in all variables $R_1, \dots, R_k$. On any structure $\mathfrak{A}$, the system S induces an operator

$$F_S : \mathrm{Pow}(A^{r_1}) \times \cdots \times \mathrm{Pow}(A^{r_k}) \to \mathrm{Pow}(A^{r_1}) \times \cdots \times \mathrm{Pow}(A^{r_k})$$

defined as $F_S(P_1, \dots, P_k) = (F_{\varphi_1}(\bar{P}), \dots, F_{\varphi_k}(\bar{P}))$, where F_{φ_i} is the operator induced by φ_i in S defined as

$$\begin{aligned} F_{\varphi_i} :\ &\mathrm{Pow}(A^{r_1}) \times \cdots \times \mathrm{Pow}(A^{r_k}) \longrightarrow \mathrm{Pow}(A^{r_i}) \\ &(R_1, \dots, R_k) \qquad\qquad\ \longmapsto \{\bar{a} : (\mathfrak{A}, R_1, \dots, R_k) \models \varphi_i[\bar{a}]\}. \end{aligned}$$

The stages S^α of an induction on such a system S of formulae are k-tuples of sets $(R_1^\alpha, \dots, R_k^\alpha)$ defined as $R_i^0 := \varnothing$, $R_i^{\alpha+1} := F_{\varphi_i}(R_1^\alpha, \dots, R_k^\alpha)$, and $R_i^\lambda := \bigcup_{\xi<\lambda} R^\xi$ for infinite limit ordinals λ.

For every structure $\mathfrak{A}$ and any tuple $\bar{a}$ from A, $\mathfrak{A} \models \psi[\bar{a}]$ if, and only if, $\bar{a} \in R_i^\infty$, where R_i^∞ denotes the i-th component of the simultaneous least fixed point of S.

Let S-LFP denote the class of LFP-formulae with simultaneous inductions.

We show next that allowing simultaneous fixed points does not increase the expressive power of LFP, i.e. S-LFP = LFP.

Theorem 2. *For any parameter-free system S of formulae in* LFP*, positive in their free fixed-point variables, $\varphi := [\mathbf{lfp}\ R_i : S](\bar{t})$ is equivalent to a formula φ^* in* LFP *(without simultaneous inductions). Further, φ and φ^* use the same set of first and second-order variables. In particular, the arity of the involved fixed-point operators does not increase.*

The theorem follows immediately from the following lemma – sometimes called the *Bekič-principle* – which is part of the folklore of the community. (See e.g. [1, Lemma 1.4.2], [15, Lemma 10.9].)

Lemma 3. *Let*

$$S := \begin{cases} R_1\bar{x}_1 & \leftarrow \varphi_1(R_1, \ldots, R_k, \bar{x}_1) \\ & \vdots \\ R_{k-1}\bar{x}_{k-1} & \leftarrow \varphi_{k-1}(R_1, \ldots, R_k, \bar{x}_{k-1}) \\ R_k\bar{x}_k & \leftarrow \varphi_k(R_1, \ldots, R_k, \bar{x}_k) \end{cases}$$

be a system of formulae in LFP *such that $[\mathbf{lfp}\ R_1 : S](\bar{x}_1)$ is parameter-free. Then $[\mathbf{lfp}\ R_1 : S]$ is equivalent to the parameter-free formula $[\mathbf{lfp}\ R_1 : T]$, where*

$$T := \begin{cases} R_1\bar{x}_1 & \leftarrow \varphi'_1(R_1, \ldots, R_{k-1}, \bar{x}_1) \\ & \vdots \\ R_{k-1}\bar{x}_{k-1} & \leftarrow \varphi'_{k-1}(R_1, \ldots, R_{k-1}, \bar{x}_k). \end{cases}$$

Here $\varphi'_i := \varphi_i(R_1, \ldots, R_{k-1}, R_k\bar{u}/[\mathbf{lfp}_{R_k,\bar{x}_k}\ \varphi_k](\bar{u}), \bar{x}_1)$ is obtained from φ_i by replacing every occurrence of an atom $R_k\bar{u}$ by the formula $[\mathbf{lfp}_{R_k,\bar{x}_k}\ \varphi_k](\bar{u})$.

Note that this lemma cannot be applied in cases where parameters are allowed.

4. Monadic two-variable fixed-point logic

As already mentioned in section 1, it is known that M-LFP^2 is strictly less expressive than M-LFP$^2_{param}$ on the class of finite graphs. In fact, M-LFP^2 can be embedded into two-variable infinitary logic $L^2_{\infty\omega}$, whereas M-LFP$^2_{param}$ can not. Due to this, it has been claimed by several authors that allowing parameters in a two-variable fixed-point logic does not yield a logic that behaves as a "proper two-variable logic" (cf., [9, 8, 6]) . The next theorem gives additional backup to this claim by showing that – subject to the obvious restriction to binary structures – the two-variable fragment of monadic least fixed-point logic with parameters is as expressive as full monadic least fixed-point logic.

Theorem 4. M-LFP$^2_{param}$ = M-LFP$_{param}$ *on binary structures. That is, for every binary signature τ the following is true: For every* M-LFP(τ)*-sentence φ there is an* M-LFP$^2_{param}(\tau)$*-sentence φ' that is equivalent to φ on all τ-structures.*

The proof is based on the simple idea of replacing every first-order quantification by a new monadic second-order variable and a fixed point construction.

5. Binary two-variable fixed-point logic

In this section we concentrate on the expressive power of two-variable *binary* least fixed-point logic with and without parameters, respectively. First we show that parameter-free two-variable binary least fixed-point logic is no more expressive than parameter-free two-variable *monadic* least fixed-point logic. Afterwards, we prove that (already on the class of finite strings) two-variable binary least fixed-point logic *with* parameters is strictly more expressive than two-variable *monadic* least fixed-point logic with parameters.

Parameter-free two-variable binary least fixed-point logic.

Theorem 5. $\mathrm{LFP}^2 = \mathrm{M\text{-}LFP}^2$ *on binary structures. That is, for every binary signature* τ *the following is true: For every* $\mathrm{LFP}^2(\tau)$*-formula* φ *there is an* $\mathrm{M\text{-}LFP}^2(\tau)$*-formula* φ' *that is equivalent to* φ *on all* τ*-structures.*

Proof. By definition, every $\mathrm{M\text{-}LFP}^2$-formula is also a valid formula in LFP^2. Hence, $\mathrm{M\text{-}LFP}^2 \leq \mathrm{LFP}^2$. Towards the converse, we show by induction on the number n of binary fixed-point operators that every formula in LFP^2 is equivalent to a formula in $\mathrm{M\text{-}LFP}^2$. For $n = 0$ this is trivial. Let λ' be a formula with $n > 0$ binary fixed-point operators and let $\lambda(x, y)$ be a sub-formula of λ' of the form $\lambda(x, y) := [\mathbf{lfp}_{R,x,y}\, \varphi](t_1, t_2)$ such that φ is in $\mathrm{M\text{-}LFP}^2$. φ can be decomposed into a positive Boolean combination of the following formulae:

- A quantifier-free formula $\theta(x, y)$ with free variables x and y. Here, by "quantifier-free" we mean absence of fixed-point operators too.
- Formulae $\psi_1(x), \ldots, \psi_s(x)$ where x and only x occurs as a free variable.
- Formulae $\chi_1(y), \ldots, \chi_r(y)$ where y and only y occurs as a free variable.
- A formula ϑ without any free variables

The crucial observation is that as φ is in $\mathrm{M\text{-}LFP}^2$ and we do not allow parameters to the fixed-point operators, the only sub-formulae with two free variables are atoms or negated atoms.

The formula φ is a positive Boolean combination of the sub-formulae described above and all sub-formulae are positive in the fixed-point variable R. Hence, the system

$$S := \begin{cases} Rxy \leftarrow \hat{\varphi}(R, x, y, \bar{X}, \bar{Y}, T) & \\ X_i x \leftarrow \psi_i(x) & \text{for all } i \in \{1, \ldots, s\} \\ Y_i y \leftarrow \chi_i(y) & \text{for all } i \in \{1, \ldots, r\} \\ T \leftarrow \vartheta & \end{cases}$$

is positive in all fixed-point variables. Here $\hat{\varphi}$ is obtained from φ by replacing the sub-formulae $\psi_i(x)$ by $X_i x$, the $\chi_i(y)$ by $Y_i y$ and ϑ by T. Note that T is a nullary second-order variable, i.e. it can only take the values $\varnothing$ or $\{()\}$. A simple induction on the stages of the fixed-point induction establishes the next lemma.

Lemma 6. $\lambda(x, y) \equiv [\mathbf{lfp}\ \ R : S](x, y)$. □

Now we can treat $[\mathbf{lfp}_{R,x,y}\, \hat{\varphi}](x, y)$ as a fixed-point formula over the extended signature $\tau \mathbin{\dot\cup} \{\bar{X}, \bar{Y}, T\}$ and consider the formulae $\hat{\varphi}^\alpha$ of the unravelling of $\hat{\varphi}$ defined as $\hat{\varphi}^0(x) := \neg x = x$ and $\hat{\varphi}^{n+1}(x) := \hat{\varphi}(Rt_1t_2/\hat{\varphi}^n(t_1,t_2))$. Here, $\hat{\varphi}(Rt_1t_2/\hat{\varphi}^n(t_1,t_2))$ is the formula obtained from $\hat{\varphi}$ by replacing every occurrence of an atom Rt_1t_2 by the result of substituting in $\hat{\varphi}^n$ x by t_1 and y by t_2. As $\hat{\varphi}$ is quantifier-free the formulae $\hat{\varphi}^n$ are quantifier-free as well. Further, there are (up to equivalence) only finitely many quantifier-free formulae for a fixed (and finite) relational signature. Thus, there is an $n < \omega$ which only depends on the signature and not on a particular interpretation of the relation variables X_i, Y_i, and T such that $\hat{\varphi}^n \equiv \hat{\varphi}^{n+1}$. (Precisely, there are $n, q < \omega$ such that $\hat{\varphi}^n \equiv \hat{\varphi}^{n+q}$. But as $\hat{\varphi}$ is monotone in R, this implies $\hat{\varphi}^n \equiv \hat{\varphi}^{n+1}$.) Consequently, for $\hat{\hat{\varphi}} := \hat{\varphi}^n$,

$$[\mathbf{lfp}_{R,x,y}\,\hat{\varphi}](x,y) \qquad \equiv \qquad \hat{\hat{\varphi}} \tag{$*$}$$

on all structures over the signature $\tau \,\dot\cup\, \{X_i, Y_i, T\}$. Note that in $\hat{\hat{\varphi}}$ the variable R does no longer occur.

The next step is to (a) eliminate in S the rule $Rxy \leftarrow \hat{\varphi}$ by applying the construction of Lemma 3 and (b) to replace every occurrence of $[\mathbf{lfp}_{R,x,y}\,\hat{\varphi}](t_1,t_2)$ by $\hat{\hat{\varphi}}(t_1,t_2)$. This construction yields the system

$$S' := \begin{cases} X_i x \leftarrow \psi_i(Rt_1t_2/\hat{\hat{\varphi}}(t_1,t_2)) \text{ for all } i \in \{1,\dots,s\} \\ \;Y_i x \leftarrow \chi_i(Rt_1t_2/\hat{\hat{\varphi}}(t_1,t_2)) \text{ for all } i \in \{1,\dots,r\} \\ \quad T \leftarrow \vartheta(Rt_1t_2/\hat{\hat{\varphi}}(t_1,t_2)) \end{cases}$$

where $\psi_i(Rt_1t_2/\hat{\hat{\varphi}}(t_1,t_2))$ denotes the formula obtained from ψ_i by replacing every occurrence of an atom Rt_1t_2 by the formula $\hat{\hat{\varphi}}(t_1,t_2)$ – the result of substituting in $\hat{\hat{\varphi}}$ t_1 for x and t_2 for y. By Lemma 3 and the equivalence $(*)$, the systems S and S' are equivalent in the sense that for all $i \in \{1,\dots,s\}$ we have

$$[\mathbf{lfp}\;\; X_i : S](x) \qquad \equiv \qquad [\mathbf{lfp}\;\; X_i : S'](x) \tag{$**$}$$

and likewise for T and all Y_i. Let $(R^\infty, X_i^\infty, Y_i^\infty, T^\infty)$ be the simultaneous least fixed point of F_S. Further, $(**)$ implies that $(X_i^\infty, Y_i^\infty, T^\infty)$ is also the simultaneous least fixed point of $F_{S'}$. By definition, $R^\infty = \{(a,b) : (\mathfrak{A}, R^\infty, X_i^\infty, Y_i^\infty, T^\infty) \models \hat{\varphi}(a,b)\}$. We claim that

$$R^\infty = \{(a,b) : (\mathfrak{A}, X_i^\infty, Y_i^\infty, T^\infty) \models [\mathbf{lfp}_{R,x,y}\,\hat{\varphi}](a,b)\}. \tag{$***$}$$

We let $R'^\infty := \{(a,b) : (\mathfrak{A}, X_i^\infty, Y_i^\infty, T^\infty) \models [\mathbf{lfp}_{R,x,y}\,\hat{\varphi}](a,b)\}$. Clearly, $R'^\infty \subseteq R^\infty$, as R^∞ is a fixed point of $\hat{\varphi}$ (with the given interpretation $X_i^\infty, Y_i^\infty, T^\infty$ of the other free variables) and R'^∞ is its least fixed point. Conversely, the sequence $(R'^\infty, X_i^\infty, Y_i^\infty, T^\infty)$ is a fixed point of F_S and thus $R^\infty \subseteq R'^\infty$.

Now we can put the various parts together to obtain the following chain of equalities:

$$\begin{aligned} R^\infty &= \{(a,b) : (\mathfrak{A}, X_i^\infty, Y_i^\infty, T^\infty) \models [\mathbf{lfp}_{R,x,y}\,\hat{\varphi}](a,b)\} && (\text{ by } (***) \text{ }) \\ &= \{(a,b) : (\mathfrak{A}, X_i^\infty, Y_i^\infty, T^\infty) \models \hat{\hat{\varphi}}(a,b)\} && (\text{ by } (*) \text{ }) \\ &= \{(a,b) : \mathfrak{A} \models \varphi^*(a,b)\}, \end{aligned}$$

where φ^* is the formula derived from $\hat{\hat{\varphi}}$ by replacing every occurrence of an atom $X_i t$ by $[\mathbf{lfp}\ X_i : S'](t)$ and likewise for the relations Y_i and T. By construction, the formula φ^* only contains monadic fixed-point operators and is equivalent to the formula $\lambda(x, y)$ from the beginning of the proof. Thus, we can replace the occurrence of λ in λ' by φ^*. The resulting formula has fewer binary fixed-point operators as λ' and, by induction hypothesis, is therefore equivalent to a formula without any binary fixed-point operators. This concludes the proof of the theorem. $\square$

Remark. The theorem naturally extends to the k-variable fragment of LFP, that is, every parameter-free formula of LFP with at most k distinct first-order variables and k-ary fixed-point operators is equivalent to a parameter-free k-variable formula of LFP with fixed-point relations of arity at most $k-1$.

Two-variable binary least fixed-point logic with parameters. We show next that LFP^2_{param} is strictly more expressive than $\mathrm{M\text{-}LFP}^2_{param}$. We prove this by showing that the non-regular string-language $\{a^n b^n \mid n \in \mathbb{N}\}$ can be defined by an LFP^2_{param}-sentence. In order to give a detailed proof, we need some additional notation: A non-empty string w over an alphabet Σ is represented by a structure $\mathfrak{W}$ over the binary signature $\tau_\Sigma := \{min, max, succ, <\} \cup \{Q_\sigma \mid \sigma \in \Sigma\}$ in the usual way: If $w = w_1 \cdots w_n$ with $w_i \in \Sigma$, then $\mathfrak{W}$ is the τ_Σ-structure with universe $W = \{1, .., n\}$, $min^{\mathfrak{W}} = 1$, $max^{\mathfrak{W}} = n$, $succ^{\mathfrak{W}} = \{(i, i{+}1) \mid i < n\}$, $<^{\mathfrak{W}}$ is the natural linear order on $\{1, .., n\}$, and $Q_\sigma^{\mathfrak{W}} = \{i \leq n \mid w_i = \sigma\}$. We say that a string-language $L \subseteq \Sigma^*$ *is expressible* in a logic $\mathcal{L}$, if there is an $\mathcal{L}(\tau_\Sigma)$-sentence φ_L such that for all non-empty strings $w \in \Sigma^*$ we have $w \in L \iff \mathfrak{W} \models \varphi_L$.

Lemma 7. *(i)* $\{a^n b^n \mid n \in \mathbb{N}\}$ *is expressible in* LFP^2_{param}.

(ii) $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ *is expressible in* LFP^2_{param}. *In particular,* LFP^2_{param} *is capable of defining a non-context-free string-language.*

Proof. We use x and y to denote the two first-order variables available in the logic LFP^2_{param}.

A binary relation E over $\{1, .., n\}$ is called a *pairing* iff the following is true for all $(i, j), (i', j') \in E$: (1) $i < j$, (2) $i = i' \iff j = j'$, and (3) $i < i' \iff j' < j$.

Let $\varphi_y(x, y, R)$ be the following $\mathrm{M\text{-}LFP}^2_{param}$-formula

$$\begin{aligned}\varphi_y(x, y, R) \ := \ \big[\,\mathbf{lfp}_{X,x}\ & succ(y, x) \ \vee\ \exists y\,\big(Xy \wedge R(x, y)\big) \ \vee \\ & \exists y\,\big(Xy \wedge succ(y, x) \wedge \exists x\,(Xx \wedge R(y, x))\,\big)\,\big](x)\,.\end{aligned}$$

Claim 1: *For every $n \in \mathbb{N}$, every pairing $E \subseteq \{1, .., n\}^2$, and all $i, j \in \{1, .., n\}$ the following is true: $\langle\{1, .., n\}, succ\rangle \models \varphi_y(i, j, E)$ if, and only if,* $i \in \{j{+}1,\ i',\ i'{+}1 \mid (i', j{+}1) \in E\}$. (An illustration is given in Figure 2.)

Due to space limitation, we have to omit the proof of this and the following two claims.

Analogously to the formula $\varphi_y(x, y, R)$ we define an $\mathrm{M\text{-}LFP}^2_{param}$-formula $\varphi_x(x, y, R)$ as follows:

$$\begin{aligned}\varphi_x(x, y, R) \ := \ \big[\,\mathbf{lfp}_{Y,y}\ & succ(y, x) \ \vee\ \exists x\,\big(Yx \wedge R(x, y)\big) \ \vee \\ & \exists x\,\big(Yx \wedge succ(y, x) \wedge \exists y\,(Yy \wedge R(y, x))\,\big)\,\big](y)\,.\end{aligned}$$

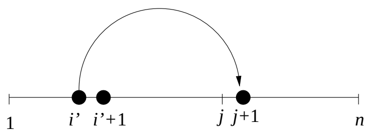

Fig. 2. The unary least fixed-point defined by the formula φ_y in case that y is interpreted by some j for which there exists an i' such that $(i', j{+}1) \in E$. The nodes that belong to this fixed-point are marked by black circles.

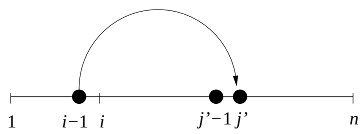

Fig. 3. The unary least fixed-point defined by the formula φ_x in case that x is interpreted by some i for which there exists a j' such that $(i{-}1, j') \in E$. The nodes that belong to this fixed-point are marked by black circles.

Claim 2: *For every $n \in \mathbb{N}$, every pairing $E \subseteq \{1,..,n\}^2$, and all $i, j \in \{1,..,n\}$ the following is true: $\langle \{1,..,n\}, succ \rangle \models \varphi_x(i,j,E)$ if, and only if, $j \in \{i{-}1,\ j',\ j'{-}1 \mid (i{-}1, j') \in E\}$.* (An illustration is given in Figure 3.)

Finally, we define the LFP$^2_{param}$-formulae $\chi(x,y,R) := x < y \ \wedge\ \big(\big(x = min \wedge y = max\big) \vee \big(\,\varphi_x(x,y,R) \wedge \varphi_y(x,y,R)\,\big)\big)$ and $\psi(x,y) := \big[\ \mathbf{lfp}_{R,xy}\, \chi(x,y,R)\,\big]\,(x,y)\,.$

Claim 3: *For every $n \in \mathbb{N}$ and all $i, j \in \{1,..,n\}$, the following is true: $\langle \{1,..,n\}, min, max, succ, < \rangle \models \psi(i,j) \iff i < j$ and $j = n - i + 1\,.$*

We are now ready to present the LFP$^2_{param}$-sentence $\varphi_{a^nb^n}$ that defines the string-language $\{a^nb^n \mid n \in \mathbb{N}\}$ via

$$\begin{aligned}\varphi_{a^nb^n} \ :=\ \exists x\, \exists y\ \ & \psi(x,y) \ \wedge\ succ(x,y) \ \wedge\ Q_a(x) \ \wedge\ Q_b(y) \wedge \\ & \forall y\, \big(y < x \rightarrow Q_a(y)\big) \ \wedge\ \forall x\, \big(y < x \rightarrow Q_b(x)\big)\,.\end{aligned}$$

Using Claim 3 it is straightforward to see that $\varphi_{a^nb^n}$ indeed defines the language $\{a^nb^n \mid n \in \mathbb{N}\}$. Thus, the proof of part *(i)* of Lemma 7 is complete.

The proof of part *(ii)* of Lemma 7 uses a similar construction. Now, however, the formula $\varphi_{a^nb^nc^n}$ defining the string-language $\{a^nb^nc^n \mid n \in \mathbb{N}\}$ is given via $\varphi_{a^nb^nc^n} \ :=\ \varphi_{a^*b^*c^*} \ \wedge\ \varphi_{a^nb^nc^*} \ \wedge\ \varphi_{a^*b^nc^n}$ where

- $\varphi_{a^*b^*c^*}$ is an FO^2-sentence expressing that the underlying string belongs to the language defined by the regular expression $a^*b^*c^*$.
- $\varphi_{a^nb^nc^*}$ is a LFP$^2_{param}$-sentence which, for an underlying string of the form $a^*b^*c^*$ expresses that the number of as is equal to the number of bs. This sentence can be obtained in a similar way as the sentence $\varphi_{a^nb^n}$ from the proof of part *(i)* of Lemma 7.
- $\varphi_{a^*b^nc^n}$ is a LFP$^2_{param}$-sentence which, for an underlying string of the form $a^*b^*c^*$ expresses that the number of bs is equal to the number of cs. Again, this sentence can be obtained in a similar way as the sentence $\varphi_{a^nb^n}$ from the proof of part *(i)* of Lemma 7. □

Using Lemma 7, one easily obtains

Theorem 8. M-LFP$^2_{param}$ $\lneq$ LFP$^2_{param}$ *on finite strings. That is, already on the class of finite strings, the two-variable fragment of binary least fixed-point logic where parameters are allowed is strictly more expressive than the two-variable fragment of monadic least fixed-point logic where parameters are allowed.*

Proof. It is well-known that the string-language $\{a^n b^n \mid n \in \mathbb{N}\}$ is not regular, i.e., due to Büchi's theorem, not expressible in monadic second-order logic MSO. As M-LFP$^2_{param}$ $\leq$ M-LFP $\leq$ MSO, we therefore obtain that $\{a^n b^n \mid n \in \mathbb{N}\}$ is not expressible in M-LFP$^2_{param}$. From Lemma 7 we obtain that $\{a^n b^n \mid n \in \mathbb{N}\}$ is expressible in LFP$^2_{param}$. $\square$

References

1. A. Arnold and D. Niwiński. *Rudiments of μ-calculus.* North Holland, 2001.
2. S. S. Cosmadakis. The complexity of evaluating relational queries. *Information and Control*, 58:101–112, 1983.
3. A. Dawar. *Feasible computation through model theory.* PhD thesis, Univ. of Pennsylvania, 1993.
4. A. Dawar, S. Lindell, and S. Weinstein. Infinitary logic and inductive definability over finite structures. *Information and Computation*, 119:160 – 175, 1995.
5. A. Dawar, E. Grädel, and S. Kreutzer. Inflationary fixed points in modal logic. *ACM Transactions on Computational Logic*, 5:282–315, 2004.
6. S. Dziembowski. Bounded-variable fixpoint queries are pspace-complete. In *Proc. of CSL'96*, volume 1258 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1997.
7. H.-D. Ebbinghaus and J. Flum. *Finite model theory.* Springer, New York, second edition, 1999.
8. E. Grädel and M. Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224:73–113, 1999.
9. E. Grädel, M. Otto, and E. Rosen. Undecidability Results for Two-Variable Logics. *Archive for Mathematical Logic*, 38:313–354, 1999. Journal version of STACS'97 paper.
10. M. Grohe. Finite variable logics in descriptive complexity theory. *Bulletin of Symbolic Logic*, 4:345–398, 1998.
11. M. Grohe and N. Schweikardt. Comparing the succinctness of monadic query languages over finite trees. *RAIRO - Theoretical Informatics and Applications (ITA)*, 38:343–373, 2004. Journal version of CSL'03 paper.
12. I. Hodkinson. Finite variable logics. *Bull. Europ. Assoc. Theor. Comp. Sci.*, 51:111–140, 1993.
13. N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.
14. Ph. Kolaitis and M. Vardi. On the expressive power of variable-confined logics. In *Proc. of LICS'96*, pages 348–359, 1996.
15. L. Libkin. *Elements of Finite Model Theory.* Springer, 2004.
16. M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9 of *Lecture Notes in Logic*. Springer-Verlag, 1997. IX+183 pages.
17. N. Schweikardt. On the expressive power of monadic least fixed point logic. In *Proc. of ICALP'04*, Lecture Notes in Computer Science, pages 1123–1135. Springer, 2004.
18. M. Y. Vardi. On the complexity of bounded-variable queries. In *PODS'95: 14th ACM Symposium on Principles of Database Systems*, pages 266–276, 1995.
19. M. Y. Vardi. The complexity of relational query languages. In *STOC'82: 14th Annual ACM Symposium on the Theory of Computing*, pages 137–146, 1982.