Theoretical Informatics and Applications Will be set by the publisher
Informatique Théorique et Applications

THREE NOTES ON THE COMPLEXITY OF MODEL CHECKING FIXPOINT LOGIC WITH CHOP

Martin Lange[1]

Abstract. This paper analyses the complexity of model checking Fixpoint Logic with Chop – an extension of the modal μ-calculus with a sequential composition operator. It uses two known game-based characterisations to derive the following results: the combined model checking complexity as well as the data complexity of FLC are EXPTIME-complete. This is already the case for its alternation-free fragment. The expression complexity of FLC is trivially P-hard and limited from above by the complexity of solving a parity game, i.e. in UP∩co-UP. For any fragment of fixed alternation depth, in particular alternation-free formulas it is P-complete.

1991 Mathematics Subject Classification. 68Q60,68Q17,03B44.

Introduction

Fixpoint Logic with Chop (FLC), as introduced by Müller-Olm [17], extends Kozen's modal μ-calculus [12] with a sequential composition operator. This vastly increases its expressive power. Since the modal μ-calculus is equi-expressive to the bisimulation-invariant fragment of Monadic Second Order Logic over all graphs [9] and to finite Rabin tree automata [6], properties expressed in this logic are inherently regular.

On infinite words, e.g., FLC is equi-expressive to alternating context-free grammars with leftmost derivations [13]. Hence, FLC is capable of specifying certain non-regular properties including all context-free and some context-sensitive ones. Moreover, FLC formulas can also describe the bisimulation-invariance class of transition systems specified in Basic Process Algebra (BPA) [17]. Consequently,

Keywords and phrases: model checking, games, EXPTIME-complete

[1] Department of Computer Science, University of Munich, Oettingenstr. 67, D-80538 München, Germany, email: `Martin.Lange@ifi.lmu.de`

FLC does not retain the finite model property and its satisfiability problem is undecidable.

Because of its ability to specify non-regular properties like uniform inevitability, unlimited counting, etc. FLC is interesting for program verification purposes via model checking. Unfortunately, FLC model checking is already undecidable for normed deterministic BPA, a very small class of infinite state systems [16]. However, it is decidable for finite-state processes for trivial reasons as noted by Müller-Olm [17]: the semantics of a formula is one of finitely many candidates only. Note that checking non-regular properties of finite systems is not an oxymoron but can be useful because of succinctness, or e.g. if the exact size of the underlying system is unknown.

Stirling and the author of this paper [16] presented a tableaux calculus which shows that FLC's model checking problem is in fact decidable in deterministic exponential time. They also show that it is hard for polynomial space. A simpler proof is due to Müller-Olm — featured in the same publication — which even shows PSPACE-hardness for the *data complexity* of FLC. This measures the model checking problem when only the transition system is considered to be the input while the formula is fixed. Dually, the *expression complexity* of a logic is the complexity of its model checking problem for a fixed transition system but varying formula. In this setting, we also speak of the *combined model checking complexity* when both the transition system and the formula are regarded as input.

Stirling and the author of this paper also claimed that the model checking problem for FLC is in PSPACE for formulas of fixed alternation depth. This, however, is based on the false observation that on a transition system of size n, each fixpoint formula has to be unfolded at most n times. In order to deal with sequential composition, Müller-Olm has lifted the semantics of a modal μ-calculus formula as a predicate in the subset lattice of a state space to a predicate transformer in the lattice of monotone functions from such subsets to subsets. If n is the size of the underlying transition system, then the height of this lattice is $n \cdot 2^n$. This only gives an exponential upper bound on the number of iterations, viz unfoldings needed for one fixpoint formula.

This paper provides three notes on the model checking problem for FLC. The first one in Section 2 recalls a game-theoretic characterisation of FLC's model checking which was used to prove inclusion of the latter in EXPTIME [16]. It then simply observes that these games are parity games whose size is linear in the size of the FLC formula. Hence, the expression complexity of FLC is that of solving parity games, i.e. in NP∩co-NP [7] and even in UP∩co-UP [10].

The second note in Section 3 targets the optimality of the exponential time upper bound. We present a family of transition systems of growing size n and an FLC formula with a least fixpoint quantifier s.t. the number of unfoldings needed to show satisfaction of the least fixpoint formula is exponential in n. This exposes a mistake in the proof of the earlier statement about FLC model checking for formulas of fixed alternation depth is in PSPACE.

The third note in Section 5 provides a lower bound for FLC's model checking. It uses another game-theoretic characterisation of FLC's model checking problem [14] which enables a simple reduction from the pushdown game problem defined in Section 4 and shown to be EXPTIME-complete by Walukiewicz [21].

Finally, Section 6 summarises these findings, discusses further work and presents – as a consequence – a separation result between FLC and another modal fixpoint logic.

1. Fixpoint Logic with Chop

Let $\mathcal{P} = \{p, \overline{p}, q, \overline{q}, \ldots\}$ be a set of propositional constants with $\overline{\overline{q}} = q$ for any q, $\mathcal{V} = \{X, Y, \ldots\}$ a set of propositional variables, and $\mathcal{A} = \{a, b, \ldots\}$ a set of action names. A labeled *transition system* is a graph $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ where $\mathcal{S}$ is a set of states, $\xrightarrow{a}$ for each $a \in \mathcal{A}$ is a binary relation on states and $L : \mathcal{S} \to 2^{\mathcal{P}}$ labels the states s.t. for all q and s: $q \in L(s)$ iff $\overline{q} \notin L(s)$.

Formulas of FLC are given by

$$\varphi \quad ::= \quad q \mid Z \mid \tau \mid \langle a \rangle \mid [a] \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mu Z.\varphi \mid \nu Z.\varphi \mid \varphi ; \varphi$$

where $q \in \mathcal{P}$, $Z \in \mathcal{V}$, and $a \in \mathcal{A}$. We will write σ for μ or ν. Formulas are assumed to be well named in the sense that each binder variable is distinct. Each closed φ induces a function $\mathit{fp}_{\varphi} : \mathcal{V} \to \text{FLC}$ that maps each variable to its defining fixed point formula. The fixpoint *type* of a variable is the type (μ or ν) of its corresponding fixpoint quantifier: X is of type σ if $\mathit{fp}_{\varphi}(X) = \sigma X.\psi$ for some ψ.

We will use the following standard abbreviations: $q \to \varphi := \overline{q} \vee \varphi$, $\mathtt{tt} := q \vee \overline{q}$ and $\mathtt{ff} := q \wedge \overline{q}$ for some $q \in \mathcal{P}$.

The set $\mathit{Sub}(\varphi)$ of subformulas of φ is defined as usual. The size of a formula is the number of its subformulas: $|\varphi| := |\mathit{Sub}(\varphi)|$.

Given a $\varphi \in$ FLC and $X, Y \in \mathit{Sub}(\varphi)$ we write $X \prec_{\varphi} Y$ if Y occurs free in $\mathit{fp}_{\varphi}(X)$. Let $<_{\varphi}$ denote the transitive closure of $\prec_{\varphi}$. The *alternation-depth* of φ, $\mathit{ad}(\varphi)$ is the index n in a maximal chain $X_0 <_{\varphi} X_2 <_{\varphi} \ldots <_{\varphi} X_n$ of variables such that adjacent variables in this chain have different fixpoint types. Let FLC^k := $\{\varphi \in \mathit{FLC} \mid \mathit{ad}(\varphi) \leq k\}$.

The semantics of an FLC formula φ, w.r.t. a transition system $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ and an environment ρ interpreting its free variables is a monotone predicate transformer $[\![\varphi]\!]^{\mathcal{T}}_{\rho} : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$ where $\rho : \mathcal{V} \to 2^{\mathcal{S}} \to 2^{\mathcal{S}}$. It is inductively defined using the fact that the monotone functions of type $2^{\mathcal{S}} \to 2^{\mathcal{S}}$ form a complete lattice together with the partial order $\sqsubseteq$ defined by

$$f \sqsubseteq g \quad \text{iff} \quad \forall T \subseteq \mathcal{S} : f(T) \subseteq g(T) \ .$$

We write $\sqcap$ and $\sqcup$ for the infima and suprema according to this ordering, and $\circ$ for function composition.

$$\begin{aligned} [\![q]\!]^{\mathcal{T}}_{\rho} &:= \lambda T.\{s \mid q \in L(s)\} \\ [\![Z]\!]^{\mathcal{T}}_{\rho} &:= \rho(Z) \end{aligned}$$

$$
\begin{array}{rcl}
[\![\tau]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.T \\
[\![\langle a\rangle]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.\{s \mid \exists t \in T, s \xrightarrow{a} t\} \\
[\![[a]]\!]^{\mathcal{T}}_{\rho} & := & \lambda T.\{s \mid \forall t \in \mathcal{S} : s \xrightarrow{a} t \Rightarrow t \in T\} \\
[\![\varphi \vee \psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \sqcup [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\varphi \wedge \psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \sqcap [\![\psi]\!]^{\mathcal{T}}_{\rho} \\
[\![\mu Z.\varphi]\!]^{\mathcal{T}}_{\rho} & := & \bigsqcap\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone}, [\![\varphi]\!]^{\mathcal{T}}_{\rho[Z\mapsto f]} \sqsubseteq f\} \\
[\![\nu Z.\varphi]\!]^{\mathcal{T}}_{\rho} & := & \bigsqcup\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone}, f \sqsubseteq [\![\varphi]\!]^{\mathcal{T}}_{\rho[Z\mapsto f]}\} \\
[\![\varphi;\psi]\!]^{\mathcal{T}}_{\rho} & := & [\![\varphi]\!]^{\mathcal{T}}_{\rho} \circ [\![\psi]\!]^{\mathcal{T}}_{\rho}
\end{array}
$$

Define the usual satisfaction relation for a closed φ as: $\mathcal{T}, s \models \varphi$ iff $s \in [\![\varphi]\!]^{\mathcal{T}}(\mathcal{S})$.

Proposition 1.1. [16] *The model checking problem for FLC is in EXPTIME.*

In order to ease notation in the rest of this paper, we introduce formulas of simultaneous fixpoint inductions that are admissible w.r.t. succinctness and expressiveness in FLC.

If $X_1, \ldots, X_n$ are variables for some $n \in \mathbb{N}$ and $\varphi_1, \ldots, \varphi_n$ are possibly open FLC formulas, then the following is a simultaneous fixpoint formula of FLC where $\sigma \in \{\mu, \nu\}$.

$$
\sigma \left(\begin{array}{ccc} X_1 & \Leftarrow & \varphi_1 \\ & \vdots & \\ X_n & \Leftarrow & \varphi_n \end{array} \right)
$$

The semantics of such a formula Φ is given by the following translation into plain FLC:

$$
\Phi \;:=\; \sigma X_1.\varphi_1(X_1, \sigma X_2.\varphi_2(X_1, X_2, \sigma X_3. \ldots, \ldots), \ldots)
$$

Note that this is just an incorporation of the Békiȅ Lemma [1] into the syntax of FLC. It obviously does not increases the expressive power. The explicit unfolding into formulas with non-simultaneous fixpoints can be exponentially longer than the formula with simultaneous fixpoints, but the unfolded version has a DAG representation that is only linear in the size of the simultaneous one.

2. Note I: The Expression Complexity of FLC

First we present a game-theoretic characterisation of FLC's model checking problem. This is a slight modification of a global model checking tableau calculus used to show inclusion in EXPTIME of FLC's model checking problem [16].

The *global* model checking game $\mathcal{G}^1_{\mathcal{T}}(s_0, \varphi_0)$ for a transition system $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \Sigma\}, L)$, a starting state s_0 and an FLC formula φ_0 is played between players $\exists$ and $\forall$ on the game board $\mathcal{S} \times 2^{\mathcal{S}} \times \mathit{Sub}(\varphi)$. A configuration is written $s, T \vdash \psi$ and its intended meaning is $s \in [\![\varphi]\!]^{\mathcal{T}}(T)$.

$$\frac{s,S \vdash \psi_1 \vee \psi_2}{s,S \vdash \psi_i}\ \exists : i \in \{0,1\} \qquad\qquad \frac{s,S \vdash \psi_1 \wedge \psi_2}{s,S \vdash \psi_i}\ \forall : i \in \{0,1\}$$

$$\frac{s,S \vdash \sigma X.\psi}{s,S \vdash X} \qquad\qquad \frac{s,S \vdash X}{s,S \vdash \psi}\ \text{if } \mathit{fp}(X) = \sigma X.\psi$$

$$\frac{s,S \vdash \psi_1;\psi_2}{s,T \vdash \psi_1 \quad |_{\forall} \quad t,S \vdash \psi_2}\ \exists : T \subseteq \mathcal{S}, \forall : t \in T$$

FIGURE 1. The rules of the global model checking game.

Every play of the game $\mathcal{G}^1_{\mathcal{T}}(s_0,\varphi_0)$ starts with the configuration $s_0,\mathcal{S} \vdash \varphi_0$ and proceeds according to the rules presented in Fig. 1. The annotation to the right of a rule explains which player chooses what in which configuration. For example, if the current configuration is of the form $s,S \vdash \psi_1;\psi_2$ then player $\exists$ chooses a $T \subseteq \mathcal{S}$ first, then player $\forall$ chooses whether to continue with $s,T \vdash \psi_1$ or to select a $t \in T$ and to continue with $t,S \vdash \psi_2$.

Player $\exists$ wins the play $C_0,\ldots$ iff one of the following holds.

(1) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash \tau$ and $s \in S$.
(2) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash \langle a\rangle$ and there is a $t \in S$ with $s \xrightarrow{a} t$.
(3) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash [a]$ and for all t with $s \xrightarrow{a} t$ we have $t \in S$.
(4) The play is infinite and the outermost variable that occurs infinitely often in it is of type ν.

Player $\forall$ wins the play $C_0,\ldots$ iff one of the following holds.

(5) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash \tau$ and $s \notin S$.
(6) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash \langle a\rangle$ and for all t with $s \xrightarrow{a} t$ we have $t \notin S$.
(7) There is an $n \in \mathbb{N}$, s.t. $C_n = s,S \vdash [a]$ and there is a t with $s \xrightarrow{a} t$ and $t \notin S$.
(8) The play is infinite and the outermost variable that occurs infinitely often in it is of type μ.

Proposition 2.1. [16] *Player $\exists$ wins the global model checking game $\mathcal{G}^1_{\mathcal{T}}(s,\varphi)$ iff $\mathcal{T}, s \models \varphi$.*

It is not hard to see that each $\mathcal{G}^1_{\mathcal{T}}(s,\varphi)$ is in fact a parity game. The nodes of this game are the configurations of $\mathcal{G}^1_{\mathcal{T}}(s,\varphi)$, the edges are given by the game rules. The priorities are assigned in the usual manner: configurations with μ-, resp. ν-variables in their formula component get odd, resp. even priorities such that the dependency order on the variables is reflected in these priorities [8]. Note that every dead-end in $\mathcal{G}^1_{\mathcal{T}}(s,\varphi)$ can be made into a loop on a single node with priority 0, resp. 1, depending on whether the winner of a play ending in this node is $\exists$, resp. $\forall$.

Proposition 2.2. *For every transition system $\mathcal{T}$ with state set $\mathcal{S}$, every $s \in \mathcal{S}$, and every $\varphi \in$ FLC there is a parity game of size at most $|\mathcal{S}| \cdot 2^{|\mathcal{S}|} \cdot |\varphi|$ that is won by player $\exists$ iff $\mathcal{T}, s \models \varphi$.*

Corollary 2.3. *The FLC model checking problem for fixed transition systems is in NP$\cap$co-NP.*

Proof. According to Proposition 2.2, the model checking problem for FLC on a transition system of size n and a formula φ can be reduced to the problem of solving a parity game of size at most $n \cdot 2^n \cdot |\varphi|$. If the transition system is fixed, i.e. n is a constant, then this yields a linear time reduction. Since solving parity games is in NP$\cap$co-NP [7], so is the model checking for FLC on a fixed transition system. □

Proposition 2.2 together with the fact that solving parity games even is in UP$\cap$-co-UP [10] yields the following stronger observation.

Corollary 2.4. *The FLC model checking problem for fixed transition systems is in UP$\cap$co-UP.*

Moreover, the maximal priority assigned in the parity game for an FLC formula φ only depends on the alternation depth of φ. Hence, if that is fixed, so is the maximal priority. Since the best known algorithms for solving parity games are exponential in the number of priorities only [2,3,11,18,19], we get a stronger result for the expression complexity of FLC^k for any $k \in \mathbb{N}$.

Corollary 2.5. *The FLC^k model checking problem for fixed transition systems is in P.*

3. Note II: A Lower Bound on Number of Unfoldings

Approximants to a fixpoint formula $\sigma X.\psi$ are defined as usual: $\mu^0 X.\psi(X) := \mathtt{ff}$, $\mu^{m+1} X.\psi(X) := \psi(\mu^m X.\psi(X))$, etc. Since here we are only concerned with finite models, we do not need to consider transfinite approximants.

We will prove that there are FLC fixpoint formulas that need an exponential number of unfoldings, resp. approximants to determine their truth value.

Let $\mathcal{T}_n$ be the transition system consisting of states $\mathcal{S}_n = \{0_0, \dots, 0_{n-1}, 1_0, \dots, 1_{n-1}\}$ s.t. each 0_i is labeled 0 and each 1_i is labeled 1. The transitions between the states are as follows.

- $0_i \xrightarrow{set} 1_j$ and $1_i \xrightarrow{set} 1_j$ for all $0 \le j < i < n$;
- $0_i \xrightarrow{unset} 0_j$ and $1_i \xrightarrow{unset} 0_j$ for all $0 \le j < i < n$;
- $0_i \xrightarrow{flip} 1_i$ and $1_i \xrightarrow{flip} 0_i$ for all $0 \le i < n$;

Let $\psi_{inc} = ([set] \wedge [\mathit{flip}]) \vee (\tau \wedge \langle unset \rangle)$. Note that ψ_{inc} formalises the step-wise increase of a binary counter modeled by $\mathcal{T}_n$. It can equivalently be rewritten to

$$\psi_{inc} \;\equiv\; (0 \wedge [set] \wedge [\mathit{flip}]) \vee (1 \wedge [set] \wedge [\mathit{flip}]) \vee (\tau \wedge \langle unset \rangle)$$

This says: a bit is set to 0 if itself and all lower bits are 1; it is set to 1 if all lower bits are 1 but itself is 0; and its value is kept if there is a lower bit that is set to 0.

With any natural number k in the range of $\{0, \dots, 2^n - 1\}$ we will associate a set of states $\|k\| \subseteq \mathcal{S}_n$ as follows: $\|k\| := \{x_i \mid \text{the } i\text{-th bit of } k \text{ in binary coding is } x\}$. For example, if $n = 3$ then $\|6\| = \{0_0, 1_1, 1_2\}$.

We say that a closed FLC formula φ *represents* the number k in the range of $\{0, \dots, 2^n - 1\}$ iff $\|k\| = [\![\varphi]\!]^{\mathcal{T}_n}(\mathcal{S}_n)$. For example, the atomic FLC formula 1 represents the number $2^n - 1$.

For any FLC formula φ let φ^k denote the k-fold sequential composition of φ, i.e. $\varphi^0 := \tau$ and $\varphi^{k+1} := \varphi; \varphi^k$.

Lemma 3.1. *For all $n > 0$ and all $k \in \{0, \dots, 2^n - 1\}$: the FLC formula $\psi_{inc}^k; 0$ represents the number k on $\mathcal{T}_n$.*

Proof. Take any $n > 0$. We prove the claim by induction on k. If $k = 0$ then $\psi_{inc}^k; 0 \equiv 0$, and $\|0\| = \{0_0, \dots, 0_{n-1}\} = [\![0]\!]^{\mathcal{T}_n}(\mathcal{S}_n)$.

Now let $k > 0$. By the induction hypothesis we have $[\![\psi_{inc}^{k-1}; 0]\!]^{\mathcal{T}_n} = \|k - 1\|$. In particular, $[\![\psi_{inc}^{k-1}; 0]\!]^{\mathcal{T}_n}$ is a constant predicate transformer, i.e. a predicate, and therefore we have

$$\begin{aligned} [\![\psi_{inc}^k; 0]\!]^{\mathcal{T}_n}(\mathcal{S}_n) &= [\![\psi_{inc}; \psi_{inc}^{k-1}; 0]\!]^{\mathcal{T}_n}(\mathcal{S}_n) = [\![\psi_{inc}]\!]^{\mathcal{T}_n}\Big([\![\psi_{inc}^{k-1}; 0]\!]^{\mathcal{T}_n}(\mathcal{S}_n)\Big) \\ &= [\![\psi_{inc}]\!]^{\mathcal{T}_n}\big(\|k-1\|\big) = \|k\| \,. \end{aligned}$$

The last equality simply holds because ψ_{inc} encodes the increase of a binary counter. □

Theorem 3.2. *There is a family of transition systems $\{\mathcal{T}_n \mid n \geq 1\}$ with a state s and FLC formulas $\mu X.\psi$ and q, s.t. for all n: $\mathcal{T}_n, s \models (\mu X.\psi); q$ but the sequence $\{m_n \mid n \geq 1\}$ of smallest approximation indices s.t. $\mathcal{T}_n, s \models (\mu^{m_n} X.\psi); q$ is $m_n = \Omega(2^n)$.*

Proof. Consider the FLC formula $(\mu Z.\tau \vee Z; \psi_{inc}); 0$. Note that for all $k \in \mathbb{N}$ we have

$$(\mu^k Z.\tau \vee Z; \psi_{inc}); 0 \;\equiv\; \bigvee_{i \leq k} \psi_{inc}^i; 0$$

because of the general equivalences $\mathtt{ff}; \varphi \equiv \mathtt{ff}$, $(\varphi_1 \vee \varphi_2); \psi \equiv \varphi_1; \psi \vee \varphi_2; \psi$, and $\tau; \varphi \equiv \varphi$.

Using Lemma 3.1 we obtain

(1) $1_{n-1} \models (\mu Z.\tau \vee Z; \psi_{inc}); 0$
(2) $1_{n-1} \models (\mu^{2^{n-1}} Z.\tau \vee Z; \psi_{inc}); 0$ but
(3) $1_{n-1} \not\models (\mu^k Z.\tau \vee Z; \psi_{inc}); 0$ for all $k = 0, \dots, 2^{n-1} - 1$.

□

This insinuates that the model checking problem for FLC requires at least deterministic exponential time. The rest of this paper shows that this is indeed the case.

4. Pushdown Games

A *pushdown frame* is a tuple $G = (Q, \Sigma, R, q_0, q_f)$ where Q is a finite set of states, Σ a finite alphabet and $R \subseteq Q \times \Sigma \times (Q \times \Sigma^*)^*$ a finite set of rules. The two designated states $q_0, q_f \in Q$ are the starting and the final state. A rule in R is denoted $(q, a) \vdash (q_1, w_1), \dots, (q_n, w_n)$.

A *pushdown game* is a pair (G, w_0) where G is a pushdown frame over the alphabet Σ and $w_0 \in \Sigma^+$. The game is played between players $\exists$ and $\forall$ starting in the configuration (q_0, w_0). It proceeds as follows.

If the current configuration is (q, aw) then player $\exists$ chooses a rule $(q, a) \vdash (q_1, w_1), \dots, (q_n, w_n)$. Then player $\forall$ chooses an $i \in \{1, \dots, n\}$, and the play continues with the configuration $(q_i, w_i w)$.

Player $\exists$ wins a play if it eventually reaches the configuration (q_f, ϵ), otherwise player $\forall$ wins. We say that she wins the game (G, w) iff she has a winning strategy for the game. The pushdown game problem is: given a pushdown game (G, w), does player $\exists$ win this game?

Proposition 4.1 (Walukiewicz [21])**.** *The pushdown game problem is EXPTIME-complete.*

In order to obtain EXPTIME-hardness of FLC's data complexity we eliminate the input word from pushdown games.

Theorem 4.2. *The pushdown game problem for games of the form (G, a) for some fixed $a \in \Sigma$ is EXPTIME-complete.*

Proof. By a simple reduction from arbitrary pushdown games to the ones in the prescribed form. Let (G, w) be a pushdown game with $G = (Q, \Sigma, R, q_0, q_f)$, and input word $w = a_0 \dots a_n$. Define the pushdown frame $G_{w,a}$ as $(Q', \Sigma, R', q_0', q_f)$ where

- $Q' = Q \cup \{q_0'\}$ for some $q_0' \notin Q$;
- $R' = R \cup \{(q_0', a) \vdash (q_0, w)\}$

Now consider the pushdown game $(G_{w,a}, a)$. Because of the added deterministic rule, each play starts by putting the original input word w onto the stack to reach the configuration (q_0, w). From then on it proceeds like any play in (G, w). Hence, player $\exists$ wins $(G_{w,a}, a)$ iff she wins (G, w). □

$$\frac{t, \delta \vdash \psi_1 \vee \psi_2}{t, \delta \vdash \psi_i} \ \exists : i \in \{0,1\} \qquad \frac{t, \delta \vdash \psi_1 \wedge \psi_2}{t, \delta \vdash \psi_i} \ \forall : i \in \{0,1\}$$

$$\frac{t, \delta \vdash \sigma X.\psi}{t, \delta \vdash X} \qquad \frac{t, \delta \vdash X}{t, \delta \vdash \psi} \ \text{if } \mathit{fp}(X) = \sigma X.\psi$$

$$\frac{t, \delta \vdash \psi_1 ; \psi_2}{t, \psi_2 ; \delta \vdash \psi_1} \qquad \frac{t, \psi ; \delta \vdash \tau}{t, \delta \vdash \psi}$$

$$\frac{t, \psi ; \delta \vdash \langle a \rangle}{t', \delta \vdash \psi} \ \exists : t \xrightarrow{a} t' \qquad \frac{t, \psi ; \delta \vdash [a]}{t', \delta \vdash \psi} \ \forall : t \xrightarrow{a} t'$$

Figure 2. The rules of the local model checking game.

5. Note III: The Data Complexity of FLC

We start by presenting another game-based characterisation of FLC's model checking problem which provides a link to pushdown games. This is a simplification of the so-called local FLC model checking games [14].

Our aim is to establish a lower complexity bound by reduction from the pushdown game problem. Note that a pushdown game is a reachability game: the winner of a play is solely determined by which configurations have been reached. Not surprisingly, alternation-free FLC formulas – even those with μ-quantifiers only – suffice for a reduction of this kind. We will therefore present these model checking games for alternation-free formulas only.

Given a transition system $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$, a starting state $s \in \mathcal{S}$, and a $\varphi \in \mathrm{FLC}^0$, the game $\mathcal{G}^2_{\mathcal{T}}(s, \varphi)$ is played between players $\exists$ and $\forall$ on the game board $\mathcal{S} \times \mathit{Sub}(\varphi)^* \times \mathit{Sub}(\varphi)$. A configuration is written $s, \delta \vdash \psi$ where δ is interpreted as a stack with the top on the left. The stack's elements are separated by the sequential composition operator ";".

Every play of $\mathcal{G}_{\mathcal{T}}(s, \varphi)$ starts with the configuration $C_0 = s, \mathtt{tt} \vdash \varphi$, and proceeds according to the rules presented in Fig. 2.

Player $\exists$ wins the play $C_0, C_1, \ldots$ iff one of the following holds.

(1) There is an $n \in \mathbb{N}$, s.t. $C_n = t, \delta \vdash q$ for some $q \in \mathcal{P}$ and $q \in L(t)$.
(2) There is an $n \in \mathbb{N}$, s.t. $C_n = t, \delta \vdash [a]$ for some $a \in \mathcal{A}$ and there is no t' with $t \xrightarrow{a} t'$.
(3) There are infinitely many i_j with $C_{i_j} = t_{i_j}, \delta_{i_j} \vdash X$ for a ν-variable X.

Player $\forall$ wins the play $C_0, C_1, \ldots$ iff one of the following holds

(4) There is an $n \in \mathbb{N}$, s.t. $C_n = t, \delta \vdash q$ for some $q \in \mathcal{P}$ and $q \notin L(t)$.
(5) There is an $n \in \mathbb{N}$, s.t. $C_n = t, \delta \vdash \langle a \rangle$ for some $a \in \mathcal{A}$ and there is no t' with $t \xrightarrow{a} t'$.
(6) There are infinitely many i_j with $C_{i_j} = t_{i_j}, \delta_{i_j} \vdash X$ for a μ-variable X.

The local game $\mathcal{G}^2_{\mathcal{T}}(s,\varphi)$ can be seen as a left-depth-first search through the global game $\mathcal{G}^1_{\mathcal{T}}(s,\varphi)$ in which the state set component is represented symbolically by an FLC formula, viz the stack. We remark that the winning conditions become more complicated for alternating formulas. It is the outermost variable that occurs infinitely often in the *limit* of the stack that determines the winner. It is therefore not evident that a reduction to parity games exists, as it does for the global game.

It is possible to show that every play has a unique winner, that the games are determined, that they characterise exactly the model checking problem for FLC, and that positional strategies suffice for winning the game.

Proposition 5.1. [14] *Player $\exists$ wins the local model checking game $\mathcal{G}^2_{\mathcal{T}}(s,\varphi)$ iff $\mathcal{T}, s \models \varphi$.*

Theorem 5.2. *The FLC model checking problem for fixed formulas is EXPTIME-hard.*

Proof. By reduction from the pushdown game problem for games of the simple form (G,a) for some fixed $a \in \Sigma$ which is EXPTIME-hard according to Theorem 4.2. We define a polynomial reduction from such pushdown games to transition systems $\mathcal{T}_G$ over the set $\{s,f\}$ of atomic propositions and the set $\Sigma \cup \{\#\}$ of transition labels.

Let (G,a) be a pushdown game with $G = (Q,\Sigma,R,q_0,q_f)$. First of all, for every $q \in Q$ there is a state q in $\mathcal{T}_G$. These are exactly those states that are labeled with s – marking it as a *state*. State q_f is the only state that is also labeled with f – marking it as *final*. Take any $q \in Q$ and any $a \in \Sigma$, and let $(q,a) \vdash \Delta_1, \ldots, (q,a) \vdash \Delta_n$ be all the rules for (q,a) in R. $\mathcal{T}_G$ has a state Δ_i and a transition $q \xrightarrow{a} \Delta_i$ for every $i = 1,\ldots,n$.

Finally, take any i and let $\Delta_i = (q_{i,1}, w_{i,1}), \ldots, (q_{i,m_i}, w_{i,m_i})$. For every $j = 1,\ldots,m_i$ there is a sequence of states and transitions

$$\Delta_i \xrightarrow{\#} \bullet \xrightarrow{b_k} \bullet \xrightarrow{b_{k-1}} \bullet \ldots \bullet \xrightarrow{b_1} q_{i,j}$$

s.t. $w_{i,j} = b_1 \ldots b_k$. In case $w_{i,j} = \epsilon$ there is only a transition $\Delta_i \xrightarrow{\#} q_{i,j}$.

Next we construct a fixed formula Φ_a describing the existence of a winning strategy for player $\exists$ in any pushdown game (G,a). Let $\Sigma = \{a_1,\ldots,a_k\}$.

$$\Phi_a \;:=\; \Big(\mu \begin{pmatrix} S & \Leftarrow & X_a \\ X_{a_1} & \Leftarrow & \Psi_{a_1} \\ & \vdots & \\ X_{a_k} & \Leftarrow & \Psi_{a_k} \end{pmatrix} \Big); f$$

where

$$\Psi_b \;:=\; \langle b\rangle; [\#]; \big(\mu Y.(s \wedge \tau) \vee (\overline{s} \wedge \bigvee_{c\in\Sigma} \langle c\rangle; Y; X_c)\big)$$

It remains to be seen that this reduction is sound and complete, i.e. player $\exists$ wins the pushdown game (G,a) with starting state q_0 iff $\mathcal{T}_G, q_0 \models \Phi_a$.

Suppose that player $\exists$ has a winning strategy ζ for the game (G,a). It can be used to define a winning strategy for her in the game $\mathcal{G}^2_{\mathcal{T}_G}(q_0,\Phi_a)$: in a configuration $q,\delta\vdash\langle b\rangle;\ldots$ with $\delta = X_{b_1};\ldots;X_{b_k};f;\mathtt{tt}$ choose the transition $q\xrightarrow{b}\Delta_i$ s.t. $\zeta(q,bb_1\ldots b_k)=\Delta_i$. In the following, player $\forall$ will implicitly select a (q',v) through his choice with $[\#]$. Let $v=c_1\ldots c_m$. Then simply unfold the fixpoint formula m times and follow the deterministic transitions to the state q'. Because of the reversed order of these transitions, the configuration reached will be

$$q',X_{c_2};\ldots;X_{c_m};X_{b_2};\ldots;X_{b_k};q_f;\mathtt{tt}\vdash X_{c_1}$$

which corresponds to the pushdown game configuration $(q',vb_2\ldots b_k)$ that player $\forall$ led to. It is not hard to see that this strategy is winning because ζ is winning. According to Proposition 5.1, we have $\mathcal{T}_G,q_0\models\Phi_a$.

The converse direction follows from determinacy: if player $\exists$ does not have a winning strategy for the pushdown game (G_w,a) then player $\forall$ has a winning strategy for this game. It can be assumed to be positional. This carries over in the same way to a winning strategy for him in the game $\mathcal{G}^2_{\mathcal{T}_G}(q_0,\Phi_a)$. Note that the conjuncts s and $\overline{s}$ in each Ψ_b enforce player $\exists$ to unfold the least fixpoint formula exactly k times if the chosen rule $(q,b)\vdash(q',v)$ in (G,a) pushes k elements onto the stack, i.e. there are k states to traverse before state q' is reached. According to Proposition 5.1 again, we have $\mathcal{T}_G,q_0\not\models\Phi_a$.

Finally, $|\mathcal{T}_G| = O(|G|)$ and $|\Phi_a| = O(|\Sigma|)$. Hence, for a fixed Σ this is a polynomial time reduction from the pushdown game problem to the set of models of the fixed FLC formula Φ_a. This even holds if the simultaneous fixpoint operator is eliminated from Φ_a at a possible exponential blow-up in length. □

A quick inspection of the structure of Φ_a yields the following stronger result.

Corollary 5.3. *The model checking problem for fixed formulas of one fixpoint type only (and thus FLC^0) is EXPTIME-hard.*

Clearly, the combined complexity of a model checking problem is at least as hard as the expression or data complexity.

Corollary 5.4. *The model checking problem for FLC is EXPTIME-hard.*

6. Conclusion

The following table summarises the statements of Proposition 1.1, Theorem 5.2, and Corollaries 2.4, 2.5, 5.3 and 5.4.

model checking	FLC	FLC^k, $k\in\mathbb{N}$
combined complexity	EXPTIME-complete	EXPTIME-complete
expression complexity	P-hard, $\in$ UP$\cap$co-UP	P-complete
data complexity	EXPTIME-complete	EXPTIME-complete

P-hardness of FLC's expression complexity follows from an unpublished P-hardness-result of the expression complexity for the modal μ-calculus by Dziembowski, Jurdziński and Niwiński [5].

The lower bounds in this paper do not only close the gap for FLC model checking but also provide insight into two further problems.

First, note that because of the predicate transformer, i.e. monotone first-order function semantics, FLC can be regarded as the natural extension of the modal μ-calculus to the next function order. This idea has been followed consequently in the introduction of *Higher Order Fixpoint Logic* (HFL) by Viswanathan and Viswanathan [20]. It combines the modal μ-calculus and a typed λ-calculus which achieves even higher expressive power than that of FLC. HFL contains a natural hierarchy of syntactic fragments s.t. level k, denoted HFL^k, contains formulas whose subformulas can all be typed using function types of order at most k. It is easy to see that the modal μ-calculus is HFL^0 and that FLC is a fragment of HFL^1.

As with FLC, HFL's model checking problem is trivially decidable as noted by Viswanathan and Viswanathan [20]. Recently, it has been shown [15] that the general problem is non-elementary, and that the combined complexity of model checking HFL^k is at least hard for deterministic $(k-3)$-fold exponential space when $k \geq 3$. Theorem 5.4 improves on this showing that model checking HFL^1, HFL^2 and HFL^3 is at least EXPTIME-hard.

Second, Corollary 5.2 separates FLC from yet another logic that is a proper extension of the modal μ-calculus and, hence, capable of defining certain non-regular properties: Dawar, Grädel and Kreutzer's *Modal Iteration Calculus* (MIC) [4]. It uses inflationary and deflationary rather than least and greatest fixpoints. Its model checking problem is PSPACE-complete [4] and in P for fixed formulas.

Corollary 6.1. *FLC $\not\leq$ MIC.*

Proof. Suppose FLC $\leq$ MIC, i.e. for every $\varphi \in$ FLC there is a $\varphi' \in$ MIC s.t. $\varphi \equiv \varphi'$. Let φ_0 be the FLC formula constructed in the proof of Theorem 5.4. Then every language in EXPTIME could be polynomially reduced to the MIC model checking problem on the fixed formula φ_0'. This, however, would imply P = EXPTIME which is impossible. □

Acknowledgments. We would like to thank Damian Niwiński for questioning a linear upper bound on necessary unfoldings of fixpoint formulas in FLC and various suggestions that have led to considerable improvements in this paper. We are also very grateful to Marcin Jurdziński for discussing FLC's model checking problem — in particular asking for the data and expression complexity. It was his observation that led to Corollaries 2.3–2.5. Finally, we would also like to thank an anonymous referee for very useful comments – in particular the observation that led to Corollary 5.2 in its simplified present form.

References

[1] H. Békič. *Programming Languages and Their Definition, Selected Papers*, volume 177 of *LNCS*. Springer, 1984.

[2] A. Browne, E. M. Clarke, S. Jha, D. E. Long, and W. Marrero. An improved algorithm for the evaluation of fixpoint expressions. *TCS*, 178(1–2):237–255, 1997.

[3] R. Cleaveland. Tableau-based model checking in the propositional μ-calculus. *Acta Informatica*, 27(8):725–748, 1990.

[4] A. Dawar, E. Grädel, and S. Kreutzer. Inflationary fixed points in modal logic. In L. Fribourg, editor, *Proc. 15th Workshop on Computer Science Logic, CSL'01*, LNCS, pages 277–291, Paris, France, 2001. Springer.

[5] S. Dziembowski, M. Jurdziński, and D. Niwiński. On the expression complexity of the modal μ-calculus model checking. Unpublished manuscript, 1996.

[6] E. A. Emerson and C. S. Jutla. Tree automata, μ-calculus and determinacy. In *Proc. 32nd Symp. on Foundations of Computer Science*, pages 368–377, San Juan, Puerto Rico, 1991. IEEE.

[7] E. A. Emerson, C. S. Jutla, and A. P. Sistla. On model-checking for fragments of μ-calculus. In *Proc. 5th Conf. on Computer Aided Verification, CAV'93*, volume 697 of *LNCS*, pages 385–396. Springer, 1993.

[8] E. Grädel, W. Thomas, and Th. Wilke, editors. *Automata, Logics, and Infinite Games*, LNCS. Springer, 2002.

[9] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus with respect to monadic second order logic. In U. Montanari and V. Sassone, editors, *Proc. 7th Conf. on Concurrency Theory, CONCUR'96*, volume 1119 of *LNCS*, pages 263–277, Pisa, Italy, 1996. Springer.

[10] M. Jurdziński. Deciding the winner in parity games is in $UP \cap$co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.

[11] M. Jurdziński. Small progress measures for solving parity games. In H. Reichel and S. Tison, editors, *Proc. 17th Ann. Symp. on Theoretical Aspects of Computer Science, STACS'00*, volume 1770 of *LNCS*, pages 290–301. Springer, 2000.

[12] D. Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, December 1983.

[13] M. Lange. Alternating context-free languages and linear time μ-calculus with sequential composition. In P. Panangaden and U. Nestmann, editors, *Proc. 9th Workshop on Expressiveness in Concurrency, EXPRESS'02*, volume 68.2 of *ENTCS*, pages 71–87, Brno, Czech Republic, 2002. Elsevier.

[14] M. Lange. Local model checking games for fixed point logic with chop. In L. Brim, P. Jančar, M. Křetínský, and A. Kučera, editors, *Proc. 13th Conf. on Concurrency Theory, CONCUR'02*, volume 2421 of *LNCS*, pages 240–254, Brno, Czech Republic, 2002. Springer.

[15] M. Lange and R. Somla. The complexity of model checking higher order fixpoint logic. In J. Jedrzejowicz and A. Szepietowski, editors, *Proc. 30th Int. Symp. on Math. Foundations of Computer Science, MFCS'05*, volume 3618 of *LNCS*, pages 640–651, Gdansk, Poland, 2005. Springer.

[16] M. Lange and C. Stirling. Model checking fixed point logic with chop. In M. Nielsen and U. H. Engberg, editors, *Proc. 5th Conf. on Foundations of Software Science and Computation Structures, FOSSACS'02*, volume 2303 of *LNCS*, pages 250–263, Grenoble, France, 2002. Springer.

[17] M. Müller-Olm. A modal fixpoint logic with chop. In C. Meinel and S. Tison, editors, *Proc. 16th Symp. on Theoretical Aspects of Computer Science, STACS'99*, volume 1563 of *LNCS*, pages 510–520, Trier, Germany, 1999. Springer.

[18] H. Seidl. Fast and simple nested fixpoint. *Information Processing Letters*, 59(6):303–308, 1996.

[19] C. Stirling. Local model checking games. In I. Lee and S. A. Smolka, editors, *Proc. 6th Conf. on Concurrency Theory, CONCUR'95*, volume 962 of *LNCS*, pages 1–11, Berlin, Germany, 1995. Springer.

[20] M. Viswanathan and R. Viswanathan. A higher order modal fixed point logic. In Ph. Gardner and N. Yoshida, editors, *Proc. 15th Int. Conf. on Concurrency Theory, CONCUR'04*, volume 3170 of *LNCS*, pages 512–528, London, UK, 2004. Springer.

[21] I. Walukiewicz. Pushdown processes: Games and model-checking. *Information and Computation*, 164(2):234–263, 2001.

Communicated by (The editor will be set by the publisher).
September 2005.