Model Checking Fixed Point Logic with Chop

Martin Lange and Colin Stirling

Laboratory for Foundations of Computer Science
Division of Informatics
University of Edinburgh
{martin,cps}@dcs.ed.ac.uk

Abstract. This paper examines FLC, which is the modal μ-calculus enriched with a sequential composition operator. Bisimulation invariance and the tree model property are proved. Its succinctness is compared to the modal μ-calculus. The main focus lies on FLC's model checking problem over finite transition systems. It is proved to be PSPACE-hard. A tableau model checker is given and an upper EXPTIME bound is derived from it. For a fixed alternation depth FLC's model checking problem turns out to be PSPACE-complete.

1 Introduction

Modal and temporal logics are well established research areas in computer science, artificial intelligence, philosophy, etc. [2, 4, 10]. An important temporal logic is Kozen's modal μ-calculus $\mathcal{L}_\mu$ [7] because it contains almost all other propositional temporal logics. In fact, it is equi-expressive to the bisimulation invariant fragment of monadic second-order logic over transition graphs [6]. Therefore, properties expressed by $\mathcal{L}_\mu$ formulas are essentially "regular".

In [9], Müller-Olm introduced FLC, fixed point logic with chop, that extends $\mathcal{L}_\mu$ with sequential composition. He showed that the expressive power of FLC is strictly greater than $\mathcal{L}_\mu$ because FLC can define non-regular properties. Whereas the semantics of a modal μ-calculus formula is a subset of states of a transition system, the semantics of an FLC formula is a predicate transformer, a function from sets of states to sets of states. This makes it easy to introduce a composition operator in the logic.

Müller-Olm proved that the satisfiability problem for FLC is undecidable because of its rich expressiveness. However, he notes that model checking finite transition systems is decidable. There are only finitely many (monotonic) functions from subsets to subsets of a finite set. Using the Tarski-Knaster Theorem [11], he shows that model checking can be done in the function lattice using fixed point approximants.

In this paper we examine FLC in more detail. We show that FLC retains some features of $\mathcal{L}_\mu$ such as the tree model property. However, most of the paper is devoted to FLC model checking over finite transition systems. We provide a tableau based model checker that avoids explicit calculation of functions and approximants. We also give lower, PSPACE, and upper, EXPTIME, complexity bounds on model checking. The upper bound is derived directly from the tableau model checker. An interesting open question is whether there is a suitable notion of automaton that captures FLC model checking.

In Section 2 we recall the syntax and semantics of FLC, example formulas that express non-regular properties and a proof of the tree model property. The tableau based model checker is defined and shown to be sound and complete in Section 3. In Section 4 we examine the complexity of model checking FLC in the general case and for fixed alternation depth, and we give upper and lower bounds. The paper concludes with some remarks on possible further research.

2 Preliminaries

Let $\mathcal{P} = \{\mathtt{tt}, \mathtt{ff}, q, \overline{q}, \ldots\}$ be a set of propositional constants that is closed under complementation, $\mathcal{V} = \{Z, Y, \ldots\}$ a set of propositional variables, and $\mathcal{A} = \{a, b, \ldots\}$ a set of action names. A labelled *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 such that, for all $s \in \mathcal{S} : q \in L(s)$ iff $\overline{q} \notin L(s)$, $\mathtt{tt} \in L(s)$, and $\mathtt{ff} \notin L(s)$. We will use infix notation $s \xrightarrow{a} t$ for transition relations.

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 $a \in \mathcal{A}$.[1] We will write σ for μ or ν. To save brackets we introduce the convention that ; binds stronger than $\wedge$ which binds stronger than $\vee$. Formulas are assumed to be well named in the sense that each binder variable is distinct. Our main interest is with closed formulas, that do not have free variables. In which case there is a function $\mathit{fp} : \mathcal{V} \to \mathrm{FLC}$ that maps each variable to its defining fixed point formula (that may contain free variables).

The set $\mathit{Sub}(\varphi)$ of subformulas of φ is defined as usual, with $\mathit{Sub}(\sigma Z.\psi) = \{\sigma Z.\psi\} \cup \mathit{Sub}(\psi)$. We say that Y depends on Z in φ, written $Z \prec_\varphi Y$, if Y occurs free in $\mathit{fp}(Z)$. We write $Z <_\varphi Y$ iff (Z, Y) is in the transitive closure of $\prec_\varphi$. The *alternation depth* of φ, $\mathit{ad}(\varphi)$, is the maximum number of variables of φ in a chain $Z_0 <_\varphi Z_1 <_\varphi \ldots <_\varphi Z_k$ where Z_{i-1} and Z_i are of different fixed point types for $0 < i \leq k$. $\mathrm{FLC}^k = \{\varphi \in \mathrm{FLC} \mid \mathit{ad}(\varphi) \leq k\}$.

An *environment* $\rho : \mathcal{V} \to (2^{\mathcal{S}} \to 2^{\mathcal{S}})$ maps variables to monotone functions of sets to sets. $\rho[Z \mapsto f]$ is the function that maps Z to f and agrees with ρ on all other arguments. The semantics $[\![\cdot]\!]^{\mathcal{T}}_\rho : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$ of an FLC formula, relative to $\mathcal{T}$ and ρ, is a monotone function on subsets of states with respect to the inclusion ordering on $2^{\mathcal{S}}$. These functions together with the partial order given by

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

form a complete lattice with joins $\sqcup$ and meets $\sqcap$. By the Tarski-Knaster Theorem [11] the least and greatest fixed points of functionals $F : (2^{\mathcal{S}} \to 2^{\mathcal{S}}) \to (2^{\mathcal{S}} \to 2^{\mathcal{S}})$ exist. They are used to interpret fixed point formulas of FLC.

[1] In [9], τ is called `term`.

To simplify the notation we assume a transition system $\mathcal{T}$ to be fixed for the remainder of the paper, and drop it from the semantic brackets.

$$\begin{aligned}
[\![q]\!]_\rho &= \lambda X.\{s \in \mathcal{S} \mid q \in L(s)\} \\
[\![Z]\!]_\rho &= \rho(Z) \\
[\![\tau]\!]_\rho &= \lambda X.X \\
[\![\varphi \vee \psi]\!]_\rho &= \lambda X.[\![\varphi]\!]_\rho(X) \cup [\![\psi]\!]_\rho(X) \\
[\![\varphi \wedge \psi]\!]_\rho &= \lambda X.[\![\varphi]\!]_\rho(X) \cap [\![\psi]\!]_\rho(X) \\
[\![\langle a\rangle]\!]_\rho &= \lambda X.\{s \in \mathcal{S} \mid \exists t \in X,\ \text{s.t. } s \xrightarrow{a} t\} \\
[\![[a]]\!]_\rho &= \lambda X.\{s \in \mathcal{S} \mid \forall t \in \mathcal{S}, s \xrightarrow{a} t \Rightarrow t \in X\} \\
[\![\mu Z.\varphi]\!]_\rho &= \bigsqcap\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone, } [\![\varphi]\!]_{\rho[Z\mapsto f]} \sqsubseteq f\} \\
[\![\nu Z.\varphi]\!]_\rho &= \bigsqcup\{f : 2^{\mathcal{S}} \to 2^{\mathcal{S}} \mid f \text{ monotone, } f \sqsubseteq [\![\varphi]\!]_{\rho[Z\mapsto f]}\} \\
[\![\varphi;\psi]\!]_\rho &= [\![\varphi]\!]_\rho \circ [\![\psi]\!]_\rho
\end{aligned}$$

A state s satisfies a formula φ, written $s \models_\rho \varphi$, iff $s \in [\![\varphi]\!]_\rho(\mathcal{S})$ for some ρ. If φ is a closed formula then ρ can be omitted and we write $s \models \varphi$.

Two formulas φ and ψ are *equivalent*, written $\varphi \equiv \psi$, iff their semantics are the same, i.e. for every $\mathcal{T}$ and every ρ: $[\![\varphi]\!]^{\mathcal{T}}_\rho = [\![\psi]\!]^{\mathcal{T}}_\rho$. This equivalence is a congruence and thus allows substituitivity. It is easy to see that there is no FLC formula φ that does not contain τ as a subformula, s.t. $\varphi \equiv \tau$.

For model checking purposes it is useful to consider a weaker equivalence. φ and ψ are called *weakly equivalent*, written $\varphi \approx \psi$, iff they are satisfied by the same states, i.e. $s \models_\rho \varphi$ iff $s \models_\rho \psi$ for any state s of any transition system $\mathcal{T}$ and every ρ. Note that weak equivalence is not a congruence.

Lemma 1. ***(Equivalences)***
a) If $\varphi \equiv \psi$ then $\varphi \approx \psi$.
b) If $\varphi \approx \psi$ then $\varphi;\mathtt{tt} \equiv \psi;\mathtt{tt}$.
c) $\varphi \approx \varphi;\mathtt{tt}$.
d) $(\varphi \vee \psi);\chi \equiv \varphi;\chi \vee \psi;\chi$ and $(\varphi \wedge \psi);\chi \equiv \varphi;\chi \wedge \psi;\chi$
e) $\tau;\varphi \equiv \varphi \equiv \varphi;\tau$.
f) $q;\varphi \equiv q$ for $q \in \mathcal{P}$.

Proof. a) If $\varphi \equiv \psi$ then $[\![\varphi]\!]_\rho(\mathcal{S}) = [\![\psi]\!]_\rho(\mathcal{S})$ for every set of states $\mathcal{S}$ and every ρ, and therefore $\varphi \approx \psi$. b) $[\![\varphi;\mathtt{tt}]\!]_\rho = [\![\varphi]\!]_\rho \circ [\![\mathtt{tt}]\!]_\rho = \lambda X.[\![\varphi]\!]_\rho(\mathcal{S})$ for any transition system with state set $\mathcal{S}$ and any ρ. But $\varphi \approx \psi$ implies $[\![\varphi]\!]_\rho(\mathcal{S}) = [\![\psi]\!]_\rho(\mathcal{S})$ and therefore $\varphi;\mathtt{tt} \equiv \psi;\mathtt{tt}$. c) Trivial. d) $[\![(\varphi \vee \psi);\chi]\!]_\rho = (\lambda X.[\![\varphi]\!]_\rho(X) \cup [\![\psi]\!]_\rho(X)) \circ [\![\chi]\!]_\rho = \lambda X.[\![\varphi]\!]_\rho([\![\chi]\!]_\rho(X)) \cup [\![\psi]\!]_\rho([\![\chi]\!]_\rho(X)) = [\![\varphi;\chi \vee \psi;\chi]\!]_\rho$ and similar for $\wedge$. e)–f) Trivial. □

In [9] it is shown how to embed $\mathcal{L}_\mu$ into FLC by using sequential composition: for instance, $\langle a\rangle\varphi$ becomes $\langle a\rangle;\varphi$. Therefore, we will sometimes omit the semicolon to maintain a strong resemblance to the syntax of $\mathcal{L}_\mu$. For example, $\langle a\rangle Z\langle a\rangle$ abbreviates $\langle a\rangle;Z;\langle a\rangle$.

In order to prove correctness of the tableau construction in Section 3 we introduce *approximants* of fixed point formulas. Let $fp(Z) = \mu Z.\varphi$ for some φ

and let $\alpha, \lambda \in \mathbb{O}\mathrm{rd}$, the ordinals, where λ is a limit ordinal. Then $Z^0 := \mathtt{ff}$, $Z^{\alpha+1} = \varphi[Z^\alpha/Z]$, $Z^\lambda = \bigvee_{\alpha<\lambda} Z^\alpha$. If $fp(Z) = \nu Z.\varphi$ then $Z^0 := \mathtt{tt}$, $Z^{\alpha+1} = \varphi[Z^\alpha/Z]$, $Z^\lambda = \bigwedge_{\alpha<\lambda} Z^\alpha$. Note that $\mu Z.\varphi \equiv \bigvee_{\alpha\in\mathbb{O}\mathrm{rd}} Z^\alpha$ and $\nu Z.\varphi \equiv \bigwedge_{\alpha\in\mathbb{O}\mathrm{rd}} Z^\alpha$. If only finite transition systems are considered $\mathbb{O}\mathrm{rd}$ can be replaced by $\mathbb{N}$.

First we recall some properties of FLC shown by Müller-Olm.

Theorem 1. [9]
a) Satisfiability for FLC is undecidable.
b) FLC does not have the finite model property.

The proof of part a) uses a reduction from the simulation equivalence problem for Basic Process Algebra which is undecidable [5]. For every BPA process P one can construct two characteristic FLC formulas ϕ_{P-} that is satisfied by all processes that simulate P, and ϕ_{P+} that is satisfied by all processes that are simulated by P. Hence the formula $\phi_{P-} \wedge \phi_{P+} \wedge \phi_{Q-} \wedge \phi_{Q+}$ is satisfiable iff P is simulation equivalent to Q. Part b) also follows from the existence of such characteristic formulas: however, also see example 2 later.

Example 1. Let $\mathcal{A} = \{a, b\}$ and $\varphi = \nu Y.[b]\mathtt{ff} \wedge [a](\nu Z.[b] \wedge [a](Z;Z));(([a]\mathtt{ff} \wedge [b]\mathtt{ff}) \vee Y)$. Formula φ expresses "the number of bs never exceeds the number of as" which is non-regular and, therefore, is not expressible in $\mathcal{L}_\mu$. This is an interesting property of protocols when a and b are the actions *send* and *receive.*

The subformula $\psi = \nu Z.[b] \wedge [a](Z;Z)$ expresses "there can be at most one b more than there are as". This can be understood best by unfolding the fixed point formula and thus obtaining sequences of modalities and variables. It is easy to see that replacing a Z with a $[b]$ reduces the number of Zs whereas replacing it with the other conjunct adds a new Z to the sequence.

Then, $[b]\mathtt{ff} \wedge [a]\psi$ postulates that at the beginning no b is possible and for every n as there can be at most n bs. Finally, the Y in φ allows such sequences to be composed or finished in a deadlock state.

We now establish that FLC has the tree model property, by showing that each closed formula defines a bisimulation invariant property[2].

Theorem 2. ***(Bisimulation invariance)*** *Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ and $s, t \in \mathcal{S}$. If s and t are bisimilar, $s \sim t$, then for all closed $\varphi \in$ FLC: $s \models \varphi$ iff $t \models \varphi$.*

Proof. Let $\varphi \in$ FLC be closed. φ is equivalent to an infinitary formula of FLC without fixed point operators and variables, using $\mu Z.\varphi \equiv \bigvee_{i\in\mathbb{N}} Z^i$ and $\nu Z.\varphi \equiv \bigwedge_{i\in\mathbb{N}} Z^i$ since $\mathcal{T}$ is assumed to be finite. Lemma 1 c) says that the resulting φ' is weakly equivalent to $\varphi';\mathtt{tt}$. Using parts d)–f) of Lemma 1 one can transform $\varphi';\mathtt{tt}$ into a formula that does not contain τ, and which is a (possibly infinitary) boolean combination of sequences of the form q or $\langle a\rangle;\psi$ or $[a];\psi$ where ψ again is of the described form. Every α, obtained in such a way, is

[2] The definition of bisimulation here includes the condition that states also preserve atomic properties in $\mathcal{P}$.

equivalent to an infinitary modal formula q or $\langle a\rangle\psi$ or $[a]\psi$, where equivalence means being satisfied by the same states. But a formula of infinitary modal logic cannot distinguish between bisimilar states and weak equivalence preserves this property. □

An immediate corollary of Theorem 2 is the tree model property.

Corollary 1. *FLC has the tree model property.*

Example 2. Let $\mathcal{A} = \{a, b\}$. An FLC formula that is satisfiable but does not have any finite model is $\varphi = (\nu Z.\langle a\rangle(Z \wedge \tau); ([b] \wedge \langle b\rangle)); ([a]\mathtt{ff} \wedge [b]\mathtt{ff})$. The formula postulates the existence of an infinite a-path, s.t. after every prefix of n as exactly n bs are possible. The body of the fixed point formula can be rewritten as $\langle a\rangle(([b] \wedge \langle b\rangle) \wedge Z; ([b] \wedge \langle b\rangle))$. This expresses that there must be a path labelled ab and all such bs lead to states that have similar properties. Moreover, after the a there is another path of the same style with one more b at its end.

Clearly, φ has an infinite model. Suppose φ has a finite model, too. This could be regarded as a finite automaton $\mathfrak{A}$ with final states being the deadlock states. But $\mathfrak{A}$ would accept the context-free and non-regular language $L = \{a^n b^n \mid n \in \mathbb{N}\}$.

The proof of Theorem 2 is similar to showing bisimulation invariance of $\mathcal{L}_\mu$ formulas. If the transition system is image-finite, i.e. $|\{t \in \mathcal{S} \mid s \xrightarrow{a} t\}| < \infty$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$, the converse implication in Theorem 2 holds, too. It is well-known that, if the transition system is finite and fixed, $\mathcal{L}_\mu$ formulas become equivalent to formulas of finitary modal logic. In particular, $\mu Z.\varphi \equiv \bigvee_{i \leq n} Z^i$ and $\nu Z.\varphi \equiv \bigwedge_{i \leq n} Z^i$ where $n = |\mathcal{S}|$. In the case of an FLC formula, $|\mathcal{S}| \cdot 2^{|\mathcal{S}|}$ is an upper bound for n according to Tarski-Knaster since this is the maximal length of a chain $f_0 \sqsubseteq f_1 \sqsubseteq \ldots \sqsubseteq f_n$. From Theorem 2 follows that, in fact, there is a linear upper bound for the number of approximants needed to eliminate fixed points in FLC.

Theorem 3. ***(Approximants)*** *Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ be finite with $s \in \mathcal{S}, S \subseteq \mathcal{S}$.*
a) $s \in [\![\mu Z.\varphi]\!]^{\mathcal{T}}_\rho(S)$ iff $\exists k \leq |\mathcal{S}|$, s.t. $s \in [\![Z^k]\!]^{\mathcal{T}}_\rho(S)$.
b) $s \in [\![\nu Z.\varphi]\!]^{\mathcal{T}}_\rho(S)$ iff $\forall k \leq |\mathcal{S}|$: $s \in [\![Z^k]\!]^{\mathcal{T}}_\rho(S)$.

Proof. a) The “if” part is trivial. For the “only if” part consider the general approximant characterisation of fixed point formulas. It implies the existence of a $k \in \mathbb{N}$ that makes $s \in [\![Z^k]\!]^{\mathcal{T}}_\rho(S)$ true. To show that it is bounded we introduce a new proposition q_S s.t. $[\![q_S]\!]^{\mathcal{T}} = \lambda X.S$. Then $s \in [\![\mu Z.\varphi]\!]^{\mathcal{T}}_\rho(S)$ iff $s \models (\mu Z.\varphi); q_S$. According to Theorem 2 $(\mu Z.\varphi); q_S$ can be translated into a sequence $\{\alpha_i \mid i \in \mathbb{N}\}$ of formulas of infinitary modal logic. We show by induction on the fixed point depth of φ that finitary modal logic suffices.

Suppose φ does not contain any $\sigma Y.\psi$. Clearly, in this case every α_i is a formula of finitary modal logic. Consider now the function $f : \alpha_i \mapsto \alpha_{i+1}$ for every $i \in \mathbb{N}$. f is monotone since α_{i+1} arises from α_i by variable substitution and transformations that preserve equivalence. This means that $s \models (\mu Z.\varphi); q_S$ implies the existence of a $k \leq |\mathcal{S}|$ s.t. $s \models \alpha_k$. But then $s \in [\![Z^k]\!]^{\mathcal{T}}_{\rho}(S)$.

Suppose now that φ has fixed point depth $n+1$ and every $\sigma Y.\psi \in Sub(\varphi)$ has fixed point depth at most n and can therefore be translated into a formula of finitary modal logic. Replacing every such $\mu Y.\psi$ in φ by $\bigvee_{k=0}^{|\mathcal{S}|} Z^k$, and every $\nu Y.\psi$ with $\bigwedge_{k=0}^{|\mathcal{S}|} Z^k$, yields a formula φ' of fixed point depth 0 that is equivalent to φ. The latter substitution uses part b) of the Lemma on a smaller formula. The same argument as above holds now for translating $\mu Z.\varphi'$ into a sequence $\{\alpha_i \mid i \leq |\mathcal{S}|\}$.

b) Here, the "only if" part is trivial. The "if" part is dual to the "only if" part of a). □

Theorem 4. *FLC is exponentially more succinct than* $\mathcal{L}_\mu$.

Proof. Let $\mathcal{A} = \{a_0, b_0, \dots, a_n, b_n\}$. Consider the binary, finite tree of depth $n+1$ whose nodes at level i have two transitions labelled a_i and b_i. It is easy to see that the minimal characteristic $\mathcal{L}_\mu$ formula χ_n for this tree is exponential in n. Consider now the infinite tree that arises from the finite one by pasting itself iteratively to its leaves. Again, every $\mathcal{L}_\mu$ formula describing this tree must be exponentially long in n. However, $\nu Z.(\langle a_0\rangle \wedge \langle b_0\rangle); \dots; (\langle a_n\rangle \wedge \langle b_n\rangle); Z$ describes this tree too and has linear length in n. □

For each $a \in \mathcal{A}$ one can regard *converse modalities* $\langle a^- \rangle, [a^-]$. Their semantics is

$$[\![\langle a^-\rangle]\!] = \lambda X.\{s \in \mathcal{S} \mid \exists t \in X, \text{ s.t. } t \xrightarrow{a} s\}$$
$$[\![[a^-]]\!] = \lambda X.\{s \in \mathcal{S} \mid \forall t \in \mathcal{S}, t \xrightarrow{a} s \Rightarrow t \in X\}$$

The tableaux of section 3 can easily be extended to handle these formulas as well. Indeed, all the complexity results of section 4 also hold for the extended logic.

Example 3. This extension of FLC is capable of defining *uniform inevitability*, ψ holds in all paths of a transition system at the same moment. Let $\mathcal{A} = \{a\}$ and $\varphi = \mu Y.\langle a\rangle Y \vee (\psi \wedge (\nu Z.[a^-]; (Z \wedge \tau); [a]); \psi)$. φ is an instance of an *eventually* formula of $\mathcal{L}_\mu$, i.e. $\mu Y.\langle a\rangle Y \vee \psi'$ says that there is a path on which ψ' eventually holds. $(\nu Z.[a^-]; (Z \wedge \tau); [a]); \psi$ says that at every state that can be reached by a sequence of n as backwards and then n as forwards ψ holds. Composing these two formulas achieves uniform inevitability. In [3] it is shown that φ has no equivalent in $\mathcal{L}_\mu$.

3 A tableau based model checker for FLC

For the remainder of the paper we restrict ourselves to finite transition systems only. In this section we present a tableau based model checker for FLC that is

$$(\vee)\ \frac{(T,S) \vdash \varphi_0 \vee \varphi_1}{(T_0,S) \vdash \varphi_0 \qquad (T_1,S) \vdash \varphi_1} \quad T = T_0 \cup T_1$$

$$(\wedge)\ \frac{(T,S) \vdash \varphi_0 \wedge \varphi_1}{(T_0,S) \vdash \varphi_0 \qquad (T_1,S) \vdash \varphi_1} \quad T = T_0 \cap T_1$$

$$(;)\ \frac{(T,S) \vdash \varphi_0 ; \varphi_1}{(T,T') \vdash \varphi_0 \qquad (T',S) \vdash \varphi_1}$$

$$\mathtt{FP}\ \frac{(T,S) \vdash \sigma Z.\varphi}{(T,S) \vdash Z} \qquad\qquad \mathtt{VAR}\ \frac{(T,S) \vdash Z}{(T,S) \vdash \varphi} \quad \text{if } \mathit{fp}(Z) = \sigma Z.\varphi$$

Fig. 1. The tableau rules for FLC.

sound and complete and that avoids explicit calculation of functions and approximants. The extra expressiveness of FLC and its succinctness suggest that the complexity of model checking FLC is higher than $\mathcal{L}_\mu$: exact bounds are presented in the next section.

Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ and assume that $T, S \subseteq \mathcal{S}$. A *tableau* for T, S and $\varphi \in$ FLC is a finite tree whose nodes are labelled $(T', S') \vdash \psi$, where $T', S' \subseteq \mathcal{S}$, and $\psi \in \mathit{Sub}(\varphi)$. The intended meaning of such a configuration is $T' \subseteq [\![\psi]\!](S')$, i.e. all the states in T' satisfy ψ relative to S'.

The tableau rules for the boolean connectives and the sequential composition operator are justified by the semantics of FLC. Fixed point formulas are replaced by their corresponding variables. A variable itself is replaced by the body of its fixed point definition. The rules are shown in figure 1.

Let $C_0 = (T_0, S_0) \vdash \varphi$. A branch $C_0, C_1, \ldots, C_n$ of a tableau for T_0, S_0 and φ is *successful* iff

- $C_n = (\emptyset, S) \vdash \psi$ for some S and ψ, or
- $C_n = (T, S) \vdash \psi$, $\psi \in \{\tau, q, \langle a \rangle, [a]\}$, and $T \subseteq [\![\psi]\!](S)$, or
- $C_n = (T_n, S_n) \vdash Z$ with $\mathit{fp}(Z) = \nu Z.\varphi$ for some φ, and
  - $\exists i < n$, s.t. $C_i = (T_i, S_i) \vdash Z$, and
  - $T_n \subseteq T_i$ and $S_n \supseteq S_i$, and
  - $\nexists j$, s.t. $i < j < n$ and $C_j = (T, S) \vdash Y$ and $Z <_\varphi Y$.

It is *unsuccessful* iff

- $C_n = (T, S) \vdash \psi$, $\psi \in \{\tau, q, \langle a \rangle, [a]\}$, and $T \not\subseteq [\![\psi]\!](S)$, or
- $C_n = (T_n, S_n) \vdash Z$ with $\mathit{fp}(Z) = \mu Z.\varphi$ for some φ, and
  - $\exists i < n$, s.t. $C_i = (T_i, S_i) \vdash Z$, and
  - $T_n \supseteq T_i$ and $S_n \subseteq S_i$, and
  - $\nexists j$, s.t. $i < j < n$ and $C_j = (T, S) \vdash Y$ and $Z <_\varphi Y$.

In all cases, C_n is called a *leaf*. A tableau is successful if all its branches are successful.

$$\dfrac{(\{s\},\mathcal{S}) \vdash \nu Z.\mu Y.\langle a\rangle Z \wedge ([b];(Y\vee\tau);\langle b\rangle)}{\dfrac{(\{s\},\mathcal{S})\vdash Z}{\dfrac{(\{s\},\mathcal{S})\vdash \mu Y.\langle a\rangle Z \wedge ([b];(Y\vee\tau);\langle b\rangle)}{\dfrac{(\{s\},\mathcal{S})\vdash Y}{\dfrac{(\{s\},\mathcal{S})\vdash \langle a\rangle Z\wedge([b];(Y\vee\tau);\langle b\rangle)}{\dfrac{(\{s\},\mathcal{S})\vdash\langle a\rangle Z}{(\{s\},\{s\})\vdash\langle a\rangle \quad (\{s\},\mathcal{S})\vdash Z} \qquad \dfrac{(\mathcal{S},\mathcal{S})\vdash[b];(Y\vee\tau);\langle b\rangle}{(\mathcal{S},\mathcal{S})\vdash[b] \quad \dfrac{(\mathcal{S},\mathcal{S})\vdash Y\vee\tau}{(\emptyset,\mathcal{S})\vdash Y \quad (\mathcal{S},\mathcal{S})\vdash\tau} \quad (\mathcal{S},\mathcal{S})\vdash\langle b\rangle}}}}}}$$

Fig. 2. The tableau for example 4.

Example 4. Let $\varphi = \nu Z.\mu Y.\langle a\rangle Z \wedge ([b];(Y \vee \tau);\langle b\rangle)$ and $\mathcal{T}$ be the transition system consisting of states $\mathcal{S} = \{s,t\}$ and transitions $s \xrightarrow{a} s$, $s \xrightarrow{b} t$, and $t \xrightarrow{b} s$. Φ says "there exists an infinite a-path from which every sequence of n b-actions can be repeated by another n b-actions.". The tableau of figure 2 shows that state s satisfies φ. To save space, rule $(;)$ has been extended to

$$\dfrac{(T,S) \vdash \varphi_0;\ldots;\varphi_k}{(T,T_0)\vdash\varphi_0 \quad (T_0,T_1)\vdash\varphi_1 \quad \ldots \quad (T_{k-1},S)\vdash\varphi_k}$$

3.1 Correctness

Theorem 5. ***(Soundness)*** *Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ be a finite transition system, $T_0, S_0 \subseteq \mathcal{S}$, and $\varphi \in FLC$. If there is a successful tableau with root $(T_0,S_0) \vdash \varphi$, then $T_0 \subseteq [\![\varphi]\!](S_0)$.*

Proof. Let $C = (T,S) \vdash \psi$ be a configuration with a $t \in T$ s.t. $t \notin [\![\psi]\!](S)$. C will be called *false* in this case. The tableau rules are backwards sound, i.e. if all the successors of a configuration C are not false then C is not false. This holds for rule `VAR` because a fixed point is equivalent to its unfolding, and for rule `FP` when variables are interpreted as approximants. We show that rule $(\vee)$ is backwards sound. Suppose there is a $t \in T$, s.t. $t \notin [\![\varphi_0 \vee \varphi_1]\!](S)$. Then $t \in T_i$ for some $i \in \{0,1\}$ because $T = T_0 \cup T_1$. But $[\![\varphi_0 \vee \varphi_1]\!](S) = [\![\varphi_0]\!](S) \cup [\![\varphi_1]\!](S)$ and therefore $t \notin [\![\varphi_i]\!](S)$ which means that $(T_i,S) \vdash \varphi_i$ is false. Backwards soundness of rules $(\wedge)$ and $(;)$ is established similarly.

Suppose now that the tableau for $C_0 = (T_0,S_0) \vdash \varphi$ is successful but $T_0 \not\subseteq [\![\varphi]\!](S_0)$, i.e. C_0 is false. From backwards soundness follows that at least one leaf $(T,S) \vdash \psi$ of the tableau must be false. $\psi \in \{\tau, q, \langle a\rangle, [a]\}$ is impossible because the branch to this leaf would be unsuccessful and, hence, the tableau itself cannot be successful.

Suppose therefore it is a configuration $C = (T, S) \vdash Z_0$ with Z_0 denoting a greatest fixed point. Then Z_0 is interpreted as the least approximant $Z_0^{k_0}$, s.t. $T \not\subseteq [\![Z_0^{k_0}]\!](S)$ but $T \subseteq [\![Z_0^{k_0-1}]\!](S)$. Note that $k_0 > 0$ because it is impossible to have a false configuration $(T, S) \vdash Z$ where Z is interpreted as Z^0. Starting from C one can continue to build a tableau and using backwards soundness again, falsity of a configuration can be pushed through this tableau towards a leaf $C' = (T', S') \vdash Z_1$. Note that the false successor of a configuration may depend on the index of an approximant and therefore Z_1 need not equal Z_0. However, Z_1 is interpreted as the least approximant $Z_1^{k_1}$ in the same way as above and the argument can be iterated. Since φ contains only a finite number of variables and the transition system at hand is finite too, the tableau must contain a branch $C_0, \ldots, C_j, \ldots, C'_j$ s.t. $C_j = (T, S) \vdash Z_j$, $C'_j = (T', S') \vdash Z_j$, $T' \subseteq T$, $S' \supseteq S$ and C'_j is false. But if there is a t s.t. $t \notin T$ then $t \notin T'$, and $[\![Z_j]\!](S) \subseteq [\![Z_j]\!](S')$. By monotonicity C_j must be false too. Note that in C_j Z_j was interpreted as the least approximant $Z_j^{k_j}$ in the described sense. Since between C_j and C'_j rule `VAR` must have been applied at least once and no greater variable occurs, C'_j contradicts the assumption that k_j was the least approximant index for which C_j is false. □

Theorem 6. ***(Completeness)*** *Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ be a finite transition system, $T_0, S_0 \subseteq \mathcal{S}$, and $\varphi \in FLC$. If $T_0 \subseteq [\![\varphi]\!](S_0)$ then there is a successful tableau rooted $(T_0, S_0) \vdash \varphi$.*

Proof. Let $C = (T, S) \vdash \psi$ be a configuration s.t. $T \subseteq [\![\psi]\!](S)$. In this case C will be called *true*. Note that the tableau rules can always be applied so that they preserve truth: if the antecedent of a rule is true then so are the consequents. This remains true when variables are interpreted by their approximants. The proof proceeds by constructing a tableau such that each node is true, and then stopping a branch whenever there is a leaf. However, if the application of the rule is `FP` and $fp(Z) = \mu Z.\psi$ then the formula in the consequent is interpreted as the least approximant Z^k s.t. $T \subseteq [\![Z^k]\!](S)$ but $T \not\subseteq [\![Z^{k-1}]\!](S)$. Note that $k > 0$ since $T \subseteq [\![Z^0]\!](S)$ only if $T = \emptyset$ and so a leaf is already reached.

Continuing from this configuration $(T, S) \vdash Z$ a tableau is built preserving truth. Suppose this tableau is unsuccessful, i.e. it has an unsuccessful branch. This branch cannot end on a configuration $(T', S') \vdash \psi$ where ψ is atomic because this configuration would be false. As in the proof of Theorem 5 a configuration $(T', S') \vdash Z$ with $T' \subseteq T$ and $S' \supseteq S$ must eventually be reached. But this contradicts the assumption that Z^k is the least approximant. We conclude that there is no least approximant and therefore that $T \not\subseteq [\![\mu Z.\psi]\!](S)$ which means the configuration $(T, S) \vdash \mu Z.\psi$ could not have been true. □

Corollary 2. *If there is a successful tableau for $(T, S) \vdash \varphi$ then there are successful tableaux $(T', S') \vdash \varphi$ for every $T' \subseteq T$ and every $S' \supseteq S$.*

4 Complexity of model checking

In this section we provide upper and lower bounds on the complexity of model checking FLC over finite transition systems.

Theorem 7. ***(General upper bound)*** *FLC model checking is in EXPTIME.*

Proof. We describe an alternating algorithm that, given a finite transition system $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$, a set $S_0 \subseteq \mathcal{S}$, and an FLC formula φ, finds a successful tableau for $(S_0, \mathcal{S}) \vdash \varphi$ if one exists. Alternating algorithms allow both nondeterministic and co-nondeterministic choices. They are taken by players $\exists$ and $\forall$. $\exists$ wants to show that a successful tableau exists. $\forall$ wants to show the opposite. Therefore he will choose which branch of the tableau is inspected, whereas $\exists$ is in charge of choosing the correct elements of the next configuration on this branch. She wins the play if the branch is successful. $\forall$ wins if they exhibit an unsuccessful branch of the tableau. It is easy to see that $\exists$ has a winning strategy iff there is a successful tableau for $(S_0, \mathcal{S}) \vdash \varphi$.

During the play each player is allowed to store one configuration $C_i = (T, S) \vdash Z$. If the play visits another configuration $C' = (T', S') \vdash Y$ with $Z <_\varphi Y$ then C_i will be deleted or overwritten by C'. Note that only an actual configuration can be stored. If the actual configuration is $(T, S) \vdash \psi$ and $\psi = \psi_0 \vee \psi_1$ or $\psi = \psi_0 \wedge \psi_1$ player $\exists$ chooses two sets T_0 and T_1 s.t. $T = T_0 \cup T_1$, resp. $T = T_0 \cap T_1$, and player $\forall$ chooses an $i \in \{0, 1\}$. The next configuration will be $(T_i, S) \vdash \psi_i$. If $\psi = \psi_0; \psi_1$ player $\exists$ chooses a $T' \subseteq \mathcal{S}$. Again, $\forall$ determines which branch of rule $(;)$ to follow. The rules for fixed points and variables are deterministic. The stored configurations are used to determine whether a branch is successful or unsuccessful.

A play can be implemented using polynomial space since three configurations only need to be kept in memory, namely the actual one and a C_i for each player in the sense of the condition for success and unsuccess. Their sizes are polynomial in both the size of the transition system and the size of the formula. Therefore, FLC model checking can be done in alternating PSPACE. This is the same as EXPTIME [1]. □

Theorem 8. ***(Upper bounds)*** *FLC^k model checking is in PSPACE for every $k \in \mathbb{N}$.*

Proof. Let $\mathcal{T} = (\mathcal{S}, \{\xrightarrow{a} \mid a \in \mathcal{A}\}, L)$ be finite and $\varphi \in$ FLC. Using Theorems 2 and 3 one can translate a fixed point formula $\sigma Z.\varphi$ into a series $\{\alpha_i \mid 0 \leq i \leq |\mathcal{S}|\}$ of modal formulas, where α_i is allowed to contain the subformula Z^{i-1}. Every such α_i can be stored as a directed acyclic graph with atomic formulas as nodes and the connectives $\vee$, $\wedge$ and ; as labelled edges. In fact, the entire sequence can be represented as one directed acyclic graph with a counter for the approximant. Evaluating a formula in a state corresponds to tracing the paths of this graph. This avoids a possible exponential blow-up in the size of α_i which could occur if the technique described in the proofs of Theorems 2 and 3 was explicitly used.

To establish whether for some $s \in \mathcal{S}$, $s \models \sigma Z.\varphi$ holds, it is enough to check $s \models \alpha_i$ for $0 \leq i \leq |\mathcal{S}|$. This might involve checking whether $t \models Z^{i-1}$ for some $t \in \mathcal{S}$. It is possible to store this information in a table of size $|\mathcal{S}| \cdot O(\varphi)^{ad(\varphi)}$, which is polynomial in the size of the input if the alternation depth of φ is fixed. □

Theorem 9. ***(Lower bound)*** *FLC model checking is PSPACE-hard.*

Proof. It is known that QBF (quantified boolean logic) is PSPACE-hard [8]. We show a reduction from QBF to the model checking problem for FLC. Let $\Phi = Q_1 x_1 \ldots Q_k x_k (C_1 \wedge \ldots \wedge C_n)$ with each $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ and each $l_{i,j} \in \{x_h, \overline{x}_h \mid 1 \leq h \leq k\}$. We construct a finite transition system $\mathcal{T}$ and a formula φ, s.t. $\mathcal{T} \models \varphi$ iff Φ is valid. The actions of $\mathcal{T}$ will be $\mathcal{A} = \{c, 0, 1, x_1, \ldots, x_k, \overline{x}_1, \ldots, \overline{x}_k\}$.

For each clause C_i construct a tree of depth $k+1$ in the following way. Beginning with $j = k$ introduce a node that has two transitions labelled x_j and $\overline{x_j}$ to two different subtrees if x_j appears in clause C_i, and to the same subtree if it does not. Continue with $j-1$ at the successor(s) until $j = 1$. Every path π in this tree induces a valuation function $\eta_\pi : \{x_1, \ldots, x_k\} \to \{0, 1\}$ by $\eta_\pi(x_i) = 1$ if x_i occurs on π. The leaves at the end of each path π are now extended with one further transition which is labelled 1, resp. 0, if η_π makes the clause evaluate to true, resp. false.

The second part of the transition system consists of the states $\{0, 1, \overline{1}, \ldots, k, \overline{k}\}$. The transitions are $0 \xrightarrow{x_1} 1, 0 \xrightarrow{\overline{x_1}} \overline{1}$, $z \xrightarrow{x_i} i, z \xrightarrow{\overline{x_i}} \overline{i}$ for $z \in \{i-1, \overline{i-1}\}$ and $2 \leq i \leq k$.

Finally, there are transitions labelled c from nodes $k, \overline{k}$ to every root of the trees representing the clauses. As an example the corresponding transition system for $\Phi = \exists x_1 \forall x_2 \forall x_3 \exists x_4 (x_1 \vee \overline{x}_2 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3 \vee x_4)$ is given in figure 3.

The formula φ is constructed in the following way.

$$\psi_i = \begin{cases} \langle x_i \rangle Z \langle x_i \rangle \vee \langle \overline{x_i} \rangle Z \langle \overline{x_i} \rangle & \text{if } Q_i = \exists \\ \langle x_i \rangle Z \langle x_i \rangle \wedge \langle \overline{x_i} \rangle Z \langle \overline{x_i} \rangle & \text{if } Q_i = \forall \end{cases} \quad \text{for } 1 \leq i < k$$

$$\psi_k = \begin{cases} \langle x_k \rangle [c] \langle x_k \rangle \vee \langle \overline{x_k} \rangle [c] \langle \overline{x_k} \rangle & \text{if } Q_i = \exists \\ \langle x_k \rangle [c] \langle x_k \rangle \wedge \langle \overline{x_k} \rangle [c] \langle \overline{x_k} \rangle & \text{if } Q_i = \forall \end{cases}$$

$$\varphi = (\mu Z. \bigvee_{i=1}^{k} \psi_i); \langle 1 \rangle$$

Intuitively for a Φ that has existential quantification only φ says: There exists a path labelled with a $w \in \mathcal{A}^*$ s.t. after every c action there is a path labelled with $\overleftarrow{w}$ and after that a 1 action is possible. If Φ contains universal quantification the path becomes a tree.

The resulting transition system has $O(|\Phi|^2)$ transitions, and $|\varphi| = O(|\Phi|)$. Therefore the reduction can be computed in polynomial time. Since only counters for the clauses and variables are needed it can even be computed in logarithmic space. It remains to show that the reduction is correct.

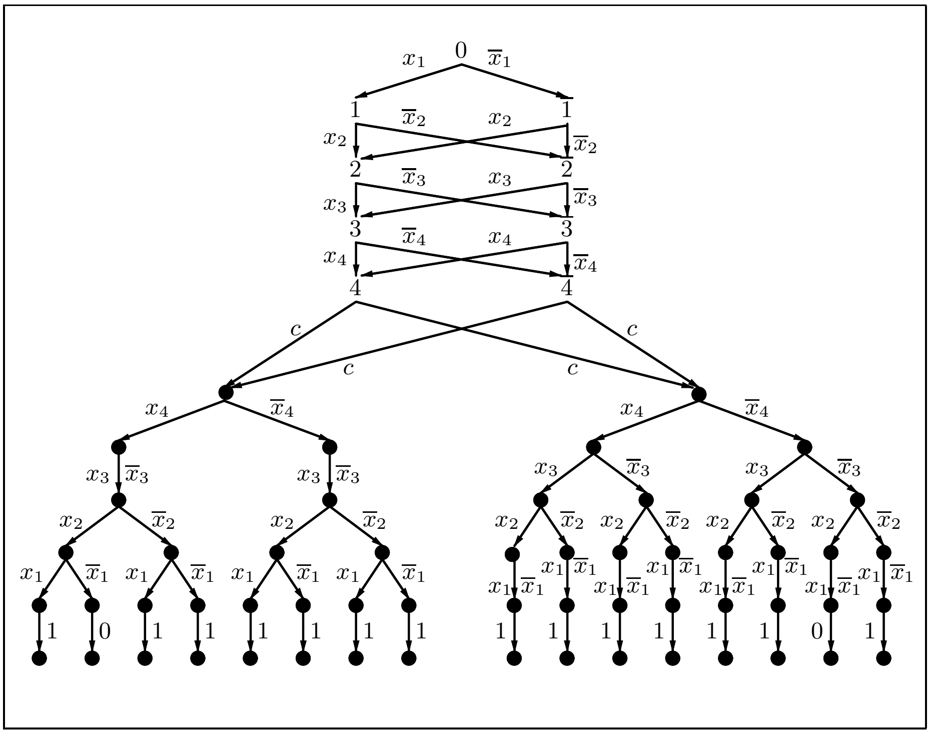

Fig. 3. The transition system for Φ.

Suppose $\mathcal{T}, 0 \models \varphi$. Since $\mathcal{T}$ is acyclic and an action 1 only occurs at the end of a path through $\mathcal{T}$ this is only possible if $0 \models Z^k; \langle 1 \rangle$. Then $Z^k; \langle 1 \rangle$ describes a tree through $\mathcal{T}$ starting with node 0 s.t. every path of the tree ends on an action 1. Furthermore, every universally quantified variable x_i corresponds to a genuine branching $\bullet \xrightarrow{x_i} \bullet$ and $\bullet \xrightarrow{\overline{x_i}} \bullet$ whereas every existentially quantified variable corresponds to either of these transitions. It is now easy to see that this tree is a witness for the validity of Φ.

Suppose now that Φ is valid. Let $\Phi = Q_1 x_1 \Phi'$. Case $Q_1 = \exists$. Φ is valid if there exists $v \in \{0, 1\}$ s.t. $\Phi'[v/x_1]$ is valid. By hypothesis there exists a tree through $\mathcal{T}$, starting with node 1 if $v = 1$ or with node $\overline{1}$ if $v = 0$, that witnesses the validity of $\Phi'[v/x_1]$. Extend this tree at its root with a transition $\bullet \xrightarrow{x_1} \bullet$ if $v = 1$, and $\bullet \xrightarrow{\overline{x_1}} \bullet$ if $v = 0$. In the case of $Q_1 = \forall$ there are two trees through $\mathcal{T}$ that witness the validity of $\Phi'[0/x_1]$ and $\Phi'[1/x_1]$. It remains to show that this tree witnesses that $\mathcal{T}, 0 \models \varphi$.

Again, let $\Phi = \exists x_1 \Phi'$ and $\Phi'[v/x_1]$ be valid, and assume w.l.o.g. that $v = 1$. Then $0 \models \varphi$ iff $1 \models (\mu Z. \bigvee\limits_{i=2}^{k} \psi_i); \langle x_1 \rangle; \langle 1 \rangle$. Suppose $\Phi = \forall x_1 \Phi'$. Then $0 \models \varphi$ iff $1 \models (\mu Z. \bigvee\limits_{i=2}^{k} \psi_i); \langle x_1 \rangle; \langle 1 \rangle$ and $\overline{1} \models (\mu Z. \bigvee\limits_{i=2}^{k} \psi_i); \langle \overline{x_1} \rangle; \langle 1 \rangle$. The fixed point formula can be unfolded further, ruling out those disjuncts that can obviously not be satisfied by the current state. Finally, after $(k-1)$ unfoldings one obtains a formula that implies $Z^k; \langle 1 \rangle$ by propositional reasoning already. $\square$

Müller-Olm has found a simpler proof of PSPACE-hardness but not published it. He uses a reduction from the universal acceptance problem for nondeterministic finite automata. Given an NFA $\mathfrak{A}$ over the alphabet Σ, does $\mathfrak{A}$ accept Σ^*? For the reduction, $\mathfrak{A}$ is regarded as a transition system that satisfies $(\nu Z.\tau \wedge \bigwedge_{a \in \Sigma} Z; \langle a \rangle); q_{fin}$ iff $\mathfrak{A}$ accepts Σ^* where q_{fin} is true in all final states of $\mathfrak{A}$.

This proves the even stronger result that model checking FLC is PSPACE-hard for fixed formulas already. The reduction does not work for $\mathcal{L}_\mu$ since fixed point formulas in $\mathcal{L}_\mu$ are right-linear. But the automaton at hand is nondeterministic and a left-linear formula is needed to allow prefixes of a word w to be accepted along paths that are not prefixes of the one accepting w.

Corollary 3. *FLC^k model checking is PSPACE-complete for every $k \geq 0$.*

5 Conclusion

FLC is a very interesting general temporal logic. Its expressive power goes beyond regular properties: indeed it can express both context-free and context-sensitive features (such as, "every path has the label sequence $a^n b^n c^n$", see [9]). Although satisfiability is undecidable, model checking finite transition systems is decidable. In the paper we have provided a reasonably simple tableau based model checker that does not explicitly calculate functions and approximants. This model checker yields an EXPTIME complexity upper bound. We also showed a PSPACE lower bound and PSPACE-completeness when alternation depth is fixed. There is a similarity with the model checking $\mathcal{L}_\mu$ problem. It is P-complete for fixed alternation depth and is in NP∩co-NP for the general case.

An interesting open question is whether there is a suitable notion of (alternating) automaton or graph game that is equivalent to model checking FLC.

Acknowledgments We would like to thank Markus Müller-Olm and the people of the Concurrency Workshop at BRICS for helpful comments on this topic.

References

[1] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, January 1981.

[2] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 14, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, New York, N.Y., 1990.

[3] E. Allen Emerson. Uniform inevitability is tree automaton ineffable. *Information Processing Letters*, 24(2):77–79, January 1987.

[4] R. Goré. Tableau methods for modal and temporal logics. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*. Kluwer, Dordrecht, 1999.

[5] J. F. Groote and H. Hüttel. Undecidable equivalences for basic process algebra. *Information and Computation*, 115(2):354–371, December 1994.

[6] 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, *CONCUR '96: Concurrency Theory, 7th Int. Conf.*, volume 1119 of *LNCS*, pages 263–277, Pisa, Italy, 26–29 August 1996. Springer.

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

[8] A. R. Meyer and L. J. Stockmeyer. Word problems requiring exponential time. In *ACM Symp. on Theory of Computing (STOC '73)*, pages 1–9, New York, April 1973. ACM Press.

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

[10] C. Stirling. Modal and temporal logics. In *Handbook of Logic in Computer Science*, volume 2 (Background: Computational Structures), pages 477–563. Clarendon Press, Oxford, 1992.

[11] A. Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific J.Math.*, 5:285–309, 1955.