

The alternation hierarchy in fixpoint logic with chop is strict too

Martin Lange *

University of Munich, Institut für Informatik, Oettingenstr. 67, D-80538 München, Germany

Received 30 August 2005; revised 12 May 2006

Available online 12 July 2006

Abstract

Fixpoint logic with chop extends the modal μ -calculus with a sequential composition operator which results in an increase in expressive power. We develop a game-theoretic characterisation of its model checking problem and use these games to show that the alternation hierarchy in this logic is strict. The structure of this result follows the lines of Arnold's proof showing that the alternation hierarchy in the modal μ -calculus is strict over the class of binary trees.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Modal logic; Expressive power; Games

1. Introduction

In 1996, Bradfield [3] and Lenzi [16] independently showed that the alternation hierarchy in the modal μ -calculus—multi-modal logic with extremal fixpoint quantifiers—is strict, i.e. there are certain formulas with nested alternating fixpoint quantifiers of depth n that are not equivalent to any formula with less than n alternating nested fixpoint quantifiers. Much earlier, Niwiński [18] already showed that there is a strict hierarchy w.r.t. expressiveness among formulas of the modal μ -calculus that do not contain the intersection operator.

* Fax: +49 89 21809338.

E-mail address: Martin.Lange@ifi.lmu.de.

The importance of these results is motivated by the model checking problem for the modal μ -calculus. The best known algorithms are polynomial in the size of the structure and the size of the formula but still exponential in its alternation depth [7,20,9,19,6]. Furthermore, syntactic alternation makes formulas hard to read. Hence, a collapse of the alternation hierarchy could have led to simpler formulas that are easier to model check.

Equivalence of formulas and, thus, the issue of an expressive hierarchy, is only meaningful on a given class of structures. In the case of the modal μ -calculus, these are primarily transition systems. Note that the existence of such a hierarchy over a certain class of structures implies the existence over any superclass. Hence, one would like to have such a result over the “smallest” possible class of structures. In case of the modal μ -calculus these are binary trees. Note that because of invariance under bisimulation and the finite model property [11] it does not matter whether finite or infinite structures are considered.

Lenzi’s proof originally works on n -ary trees for some fixed n . Since they can be encoded using binary trees, his result implies the existence of the alternation hierarchy over binary trees. Bradfield subsequently extended his proof to binary trees as well [4,5]. Note that the alternation hierarchy collapses over the class of linear structures [22,10,12].

Without a doubt the nicest proof establishing the hierarchy over binary trees was, however, given by Arnold [1]. First of all he shows the existence of a hierarchy w.r.t. expressiveness among parity tree automata, a special case of Rabin tree automata. Second, he uses the equivalence between the model checking problem for the modal μ -calculus and parity games: there are formulas of the modal μ -calculus which describe exactly those games that are won by either of the players. Finally, he uses Banach’s fixpoint theorem on the metric space of binary trees to show that those formulas are hard for each level of the hierarchy, i.e. they are not equivalent to any formula on lower levels.

Those formulas are the so-called Walukiewicz formulas [23] that are simply a generalisation of the Emerson–Jutla [8] formulas and are very similar to the formulas that are shown to be hard in Bradfield’s proof [4].

In 1999, Müller-Olm introduced *Fixpoint Logic with Chop* (FLC), which extends the modal μ -calculus with a sequential composition operator [17]. He showed that the expressive power of FLC reaches far beyond that of the modal μ -calculus. Despite this, its model checking problem remains decidable in deterministic, singly exponential time [15]. Again, the known model checking algorithms are exponential in the syntactic nesting depth of alternating fixpoint quantifiers [15,14]. Thus, it is fair to ask whether the alternation hierarchy within FLC is strict, too.

In the following we will answer this question to the affirmative. In order to do so, we adapt Arnold’s proof for the strictness of the modal μ -calculus hierarchy over binary trees. In Section 2, we recall the syntax and semantics of FLC as well as its fragments of bounded alternation. Since there is no well-known correspondence to an automaton model we show the hierarchy result directly for the logic. This, however, requires a game-based characterisation of the model checking problem for FLC which we introduce and prove correct in Section 3. A preliminary version of these games with a misleading definition of winning condition has been published before [14]. Section 4 starts with another crucial ingredient to the hierarchy theorem: complementation closure. Given that the semantics of an FLC formula is a predicate transformer, it is not obvious that for every formula there is a complement. Yet the games of Section 3 provide a simple explanation that this is indeed the case. The hard part that follows proves the existence of formulas in FLC that describe exactly those FLC games that are won by one of the players. The rest of Section 4 finishes the hierarchy

result by putting everything together just like it is done by Arnold. Finally, Section 5 contains a short discussion of this result.

2. Preliminaries

2.1. Syntax and semantics

Let \mathcal{P} be a countably infinite set of propositions, and \mathcal{V} be a countably infinite set of variable names. Formulas of FLC over \mathcal{P} and \mathcal{V} are given by the following grammar.

$$\varphi ::= q \mid Z \mid \tau \mid \diamond \mid \square \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mu Z. \varphi \mid \nu Z. \varphi \mid \varphi; \varphi$$

where $q \in \mathcal{P}$, and $Z \in \mathcal{V}$. We will write σ for either μ or ν , and $\bar{\sigma}$ for the quantifier that is not σ . To save brackets we introduce the convention that $;$ binds stronger than \wedge which binds stronger than \vee . We also use the abbreviations $\text{tt} := \nu X. X$, $\text{ff} := \mu X. X$ and, if \mathcal{P} is finite, $\bar{q} := \bigvee_{q' \neq q} q'$, and $q \rightarrow \varphi := \bar{q} \vee \varphi$.

The set $\text{Sub}(\varphi)$ of subformulas of φ is defined as usual, with $\text{Sub}(\sigma Z. \psi) = \{\sigma Z. \psi\} \cup \text{Sub}(\psi)$.

Formulas are assumed to be well-named in the sense that no variable in it is bound by a μ or a ν more than once. Our main interest is with formulas that do not have free variables, in which case there is a function $fp_\varphi : \mathcal{V} \cap \text{Sub}(\varphi) \rightarrow \text{Sub}(\varphi)$ that maps each variable X to its unique defining fixpoint formula $\sigma X. \psi$ in φ .

Given two variables $X, Y \in \text{Sub}(\varphi)$ for some φ , we write $X <_\varphi Y$ if Y occurs free in $fp_\varphi(X)$. A variable X is called *outermost* among a set of variables $V \subseteq \mathcal{V} \cap \text{Sub}(\varphi)$ if it is the greatest in V w.r.t. $<_\varphi$.

An infinite binary tree over \mathcal{P} is a function $t : \{0, 1\}^* \rightarrow \mathcal{P}$. The root of a tree is denoted ϵ . Let $\mathcal{T}_\mathcal{P}$ be the set of all infinite binary trees over \mathcal{P} . In the following we will abbreviate the power set $2^{\{0, 1\}^*}$ of the domain of infinite, binary trees simply as \mathbb{D} .

An *environment* $\rho : \mathcal{V} \rightarrow (\mathbb{D} \rightarrow \mathbb{D})$ assigns to each variable a function from sets of positions to sets of positions in a tree; $\rho[Z \mapsto f]$ is the function that maps Z to f and agrees with ρ on all other arguments. The semantics $\llbracket \cdot \rrbracket_\rho^t : \mathbb{D} \rightarrow \mathbb{D}$ of an FLC formula, relative to a tree t is such a function. It is monotone with respect to the inclusion ordering on \mathbb{D} . These functions together with the partial order given by

$$f \sqsubseteq g \quad \text{iff} \quad \forall T \in \mathbb{D} : f(T) \subseteq g(T)$$

form a complete lattice with joins \sqcup and meets \sqcap – defined as the pointwise intersection, respectively, union. By the Knaster-Tarski Theorem [21] the least and greatest fixpoints of functionals $F : (\mathbb{D} \rightarrow \mathbb{D}) \rightarrow (\mathbb{D} \rightarrow \mathbb{D})$ exist. They are used to interpret fixpoint formulas of FLC. The semantics is given in Fig. 1.

For any FLC formula φ and any environment ρ let $\|\varphi\|_\rho^t = \llbracket \varphi \rrbracket_\rho(\{0, 1\}^*)$. We call this the set of positions in t defined by φ and ρ . We also write $t \models_\rho \varphi$ if $\epsilon \in \|\varphi\|_\rho^t$. If φ is closed we may omit ρ in both kinds of notation.

$$\begin{aligned}
\llbracket q \rrbracket_\rho^t &= \lambda T. \{w \in \{0, 1\}^* \mid t(w) = q\} \\
\llbracket Z \rrbracket_\rho^t &= \rho(Z) \\
\llbracket \tau \rrbracket_\rho^t &= \lambda T. T \\
\llbracket \varphi \vee \psi \rrbracket_\rho^t &= \llbracket \varphi \rrbracket_\rho^t \sqcup \llbracket \psi \rrbracket_\rho^t \\
\llbracket \varphi \wedge \psi \rrbracket_\rho^t &= \llbracket \varphi \rrbracket_\rho^t \sqcap \llbracket \psi \rrbracket_\rho^t \\
\llbracket \Diamond \rrbracket_\rho^t &= \lambda T. \{w \in \{0, 1\}^* \mid wi \in T \text{ for some } i \in \{0, 1\}\} \\
\llbracket \Box \rrbracket_\rho^t &= \lambda T. \{w \in \{0, 1\}^* \mid wi \in T \text{ for both } i \in \{0, 1\}\} \\
\llbracket \mu Z. \varphi \rrbracket_\rho^t &= \sqcup \{f : \mathbb{D} \rightarrow \mathbb{D} \mid f \text{ monotone}, \llbracket \varphi \rrbracket_{\rho[Z \mapsto f]}^t \sqsubseteq f\} \\
\llbracket \nu Z. \varphi \rrbracket_\rho^t &= \sqcup \{f : \mathbb{D} \rightarrow \mathbb{D} \mid f \text{ monotone}, f \sqsubseteq \llbracket \varphi \rrbracket_{\rho[Z \mapsto f]}^t\} \\
\llbracket \varphi; \psi \rrbracket_\rho^t &= \llbracket \varphi \rrbracket_\rho^t \circ \llbracket \psi \rrbracket_\rho^t
\end{aligned}$$

Fig. 1. The semantics of FLC formulas.

Two formulas φ and ψ are *equivalent*, written $\varphi \equiv \psi$, iff their semantics are the same, i.e. for every environment ρ and every $t \in \mathcal{T}_P$: $\llbracket \varphi \rrbracket_\rho^t = \llbracket \psi \rrbracket_\rho^t$. Two formulas φ and ψ are *weakly equivalent*, written $\varphi \approx \psi$, iff they define the same set of positions in a tree, i.e. for every ρ and every $t \in \mathcal{T}_P$: $\llbracket \varphi \rrbracket_\rho^t = \llbracket \psi \rrbracket_\rho^t$. The following is immediate.

Lemma 1. *For all $\varphi, \psi \in \text{FLC}$: if $\varphi \equiv \psi$ then $\varphi \approx \psi$.*

Let φ_0 be a closed FLC formula and X_1, \dots, X_n all μ -variables occurring in φ_0 s.t. for all $i, j \in \{1, \dots, n\}$: $X_i <_{\varphi_0} X_j$ implies $j < i$. Note that it is always possible to order variables in such a way since two variables cannot both occur unquantified in each other's fixpoint definition—provided that φ_0 is well-named.

A μ -signature (for φ_0) is a tuple $\eta = (\alpha_1, \dots, \alpha_i)$ of ordinal numbers such that $0 \leq i \leq n$. If $k = 0$ then the signature is the empty tuple. Such μ -signatures, regardless of their actual length, are ordered lexicographically. It is well-known that this ordering is well-founded since the standard ordering on the ordinal numbers is well-founded.

Let $\mu X_i. \psi(X_1, \dots, X_i) \in \text{Sub}(\varphi_0)$. Note that ψ may contain free ν -variables other than X_1, \dots, X_i . We do not mention them explicitly here in order to avoid notational overkill. Approximants of such a formula are defined for every μ -signature of length i as follows. Let ρ be the environment that maps every ν -variable to the constant function $\lambda T. \{0, 1\}^*$, and $t \in \mathcal{T}_P$.

$$\begin{aligned}
X_i^{(\alpha_1, \dots, \alpha_{i-1}, 0)} &:= \lambda T. \emptyset \\
X_i^{(\alpha_1, \dots, \alpha_{i-1}, \alpha+1)} &:= \llbracket \psi \rrbracket_{\rho[X_i \mapsto X_i^{(\alpha_1, \dots, \alpha_{i-1}, \alpha)}, X_j \mapsto X_j^{(\alpha_1, \dots, \alpha_j)} \text{ for all } j=1, \dots, i-1]}^t
\end{aligned}$$

$$X_i^{(\alpha_1, \dots, \alpha_{i-1}, \kappa)} := \bigsqcup_{\alpha < \kappa} X_i^{(\alpha_1, \dots, \alpha_{i-1}, \alpha)}$$

where κ is a limit ordinal.

A ν -signature is defined in just the same way. It interprets possible free μ -variables by the constant function $\lambda T. \emptyset$. We will not distinguish μ - and ν -signatures syntactically.

The next result is a standard result about approximants that follows from the Knaster-Tarski Theorem [21]. Since we interpret free ν -variables inside an approximant for a μ -variable as the maximal element in the function lattice $\mathbb{D} \rightarrow \mathbb{D}$ and also vice-versa, its proof also needs monotonicity.

Lemma 2. *For all trees $t \in \mathcal{T}_{\mathcal{P}}$, all $T \subseteq \{0, 1\}^*$, and all $\varphi(X) \in \text{FLC}$ with at most one free variable X we have:*

- (1) *if $w \in \llbracket \mu X. \varphi \rrbracket^t(T)$ then there is a successor ordinal α s.t. $w \in X^{(\alpha)}(T)$.*
- (2) *if $w \notin \llbracket \nu X. \varphi \rrbracket^t(T)$ then there is a successor ordinal α s.t. $w \notin X^{(\alpha)}(T)$.*

We note that this easily carries over to formulas with more than one free variable $\varphi(X_1, \dots, X_n)$ and n -ary μ -, respectively, ν -signatures. If $w \in \llbracket \mu X. \varphi \rrbracket_\rho^t(T)$ s.t. ρ interprets X_1, \dots, X_{n-1} by $X_1^{\eta_1}, \dots, X_{n-1}^{\eta_{n-1}}$ then there is an η s.t. $w \in X^\eta(T)$. The μ -signature η is obtained by extending the pointwise infimum of the η_i for $i = 1, \dots, n-1$ with an α whose existence is guaranteed by Lemma 2.

2.2. Syntactic alternation

The Niwiński hierarchy categorises formulas of a fixpoint calculus according to the nesting structure of their fixpoint subformulas. Let $\Sigma_0^{\text{syn}} = \Pi_0^{\text{syn}}$ consist of all fixpoint-quantifier-free FLC formulas. Higher levels are built inductively in the following way. $\Sigma_{n+1}^{\text{syn}}$ is the least set of FLC formulas that contains $\Sigma_n^{\text{syn}} \cup \Pi_n^{\text{syn}}$ and satisfies the following constraints.

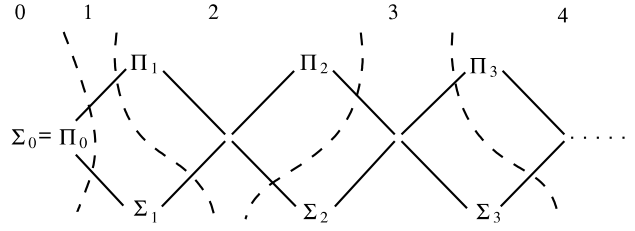
- (1) If $\varphi \in \Sigma_n^{\text{syn}}$ then $\mu X. \varphi \in \Sigma_n^{\text{syn}}$ for any variable X .
- (2) If $\varphi(X_1, \dots, X_m) \in \Sigma_n^{\text{syn}}$ and $\psi_1, \dots, \psi_m \in \Sigma_n^{\text{syn}}$, then $\varphi(\psi_1, \dots, \psi_m) \in \Sigma_n^{\text{syn}}$ for any $m \in \mathbb{N}$, provided that no free variable of any ψ_i gets bound by a quantifier in φ .

Π_{n+1}^{syn} is built in the same way with ν instead of μ .

From this hierarchy we derive two semantical alternation hierarchies, reflecting the weak and strong equivalence in FLC. For all $n \in \mathbb{N}$ we define

$$\begin{aligned} \Sigma_n^w &:= \{ \varphi \in \text{FLC} \mid \exists \psi \in \Sigma_n^{\text{syn}} \text{ s.t. } \varphi \approx \psi \} \\ \Pi_n^w &:= \{ \varphi \in \text{FLC} \mid \exists \psi \in \Pi_n^{\text{syn}} \text{ s.t. } \varphi \approx \psi \} \\ \Sigma_n^s &:= \{ \varphi \in \text{FLC} \mid \exists \psi \in \Sigma_n^{\text{syn}} \text{ s.t. } \varphi \equiv \psi \} \\ \Pi_n^s &:= \{ \varphi \in \text{FLC} \mid \exists \psi \in \Pi_n^{\text{syn}} \text{ s.t. } \varphi \equiv \psi \} \end{aligned}$$

It is not hard to see that the syntactic hierarchy is the finest, the weak semantical hierarchy is the coarsest, and the strong equivalence hierarchy lies in between.

Fig. 2. Slicing the hierarchy succinctly into levels for an odd n .

Lemma 3. For all $n \in \mathbb{N}$ we have $\Sigma_n^{\text{syn}} \subseteq \Sigma_n^s \subseteq \Sigma_n^w$.

Later we will have to build a fixed formula Φ_n that reflects the alternation structure of any $\varphi \in \Sigma_n^{\text{syn}}$. Note that a formula of Σ_n^{syn} can contain fixpoint subformulas of $2n - 1$ different sets Σ_i^{syn} and Π_i^{syn} . The straightforward trick of mapping each subformula in Σ_i^{syn} to the odd $2i - 1$ and every subformula in Π_i^{syn} to the even $2i$ is not applicable here for the resulting formula would be in $\Sigma_{2n-1}^{\text{syn}}$. Hence, we assign levels in a more succinct way, depicted in Fig. 2 for some even n . Formally, let $\varphi_0 \in \Sigma_n^{\text{syn}}$ for some $n \in \mathbb{N}$. For all $Y \in \text{Sub}(\varphi_0) \cap \mathcal{V}$ let

$$\text{lvl}_{\varphi_0}(Y) := \min \{ k \mid \begin{array}{l} \text{fp}_{\varphi_0}(Y) \in \Sigma_k^{\text{syn}} \setminus \Pi_{k-1}^{\text{syn}} \text{ and } k \equiv n \pmod{2}, \text{ or} \\ \text{fp}_{\varphi_0}(Y) \in \Pi_k^{\text{syn}} \setminus \Sigma_{k-1}^{\text{syn}} \text{ and } k \not\equiv n \pmod{2} \end{array} \}$$

3. Model checking games

3.1. The rules and winning conditions

The model checking game $\mathcal{G}(t, \varphi)$ is played between players \exists and \forall on a $t \in \mathcal{T}_{\mathcal{P}}$ and an FLC formula φ . Configurations of the game are of the form $w, \delta \vdash \psi$ where $w \in \{0, 1\}^*$, $\psi \in \text{Sub}(\varphi)$ and $\delta \in \text{Sub}(\varphi)^*$. The latter is interpreted as a stack with the top on the left and the sequential composition operator as a separator. For two stacks δ and δ' we write $\delta \preceq \delta'$ if there is a (possibly empty) stack $\gamma \in \text{Sub}(\varphi)^*$ s.t. $\delta' = \gamma; \delta$, i.e. δ' is an extension of δ .

A play of $\mathcal{G}(t, \varphi)$ starts in the configuration $C_0 = \epsilon, \text{tt} \vdash \varphi$, and proceeds according to the rules presented in Fig. 3. The premisses of a rule are written below the hypothesis. The annotation to the right determines the player whose turn it is to choose one of the premisses.

Let C_0, C_1, \dots be an infinite play of the game $\mathcal{G}(t, \varphi)$ s.t. for all $i \in \mathbb{N}$: $C_i = w_i, \delta_i \vdash \psi_i$ for some w_i, δ_i and ψ_i . We call a variable X *stack-increasing* if there are infinitely many $i_0, i_1, \dots \in \mathbb{N}$, s.t. for all $j \in \mathbb{N}$:

- $C_{i_j} = w_j, \delta_j \vdash X$ for some w_j and some $\delta_j \in \text{Sub}(\varphi)^*$,
- for all $k > i_j$: $\delta_{i_j} \preceq \delta_k$.

In other words, a stack-increasing variable defines an infinite set of configurations s.t. the stack contents of each of these configurations do not get popped.

$$\begin{array}{ll}
(\wedge) \frac{w, \delta \vdash \psi_0 \wedge \psi_1}{w, \delta \vdash \psi_0 \quad w, \delta \vdash \psi_1} \forall & (\vee) \frac{w, \delta \vdash \psi_0 \vee \psi_1}{w, \delta \vdash \psi_0 \quad w, \delta \vdash \psi_1} \exists \\
(\text{FP}) \frac{w, \delta \vdash \sigma Z. \psi}{w, \delta \vdash Z} & (\text{V}) \frac{w, \delta \vdash Z}{w, \delta \vdash \psi} \text{ if } fp_{\varphi_0}(Z) = \sigma Z. \psi \\
(;) \frac{w, \delta \vdash \psi_0; \psi_1}{w, \psi_1; \delta \vdash \psi_0} & (\tau) \frac{w, \psi; \delta \vdash \tau}{w, \delta \vdash \psi} \\
(\diamond) \frac{w, \psi; \delta \vdash \diamond}{w_0, \delta \vdash \psi \quad w_1, \delta \vdash \psi} \exists & (\square) \frac{w, \psi; \delta \vdash \square}{w_0, \delta \vdash \psi \quad w_1, \delta \vdash \psi} \forall
\end{array}$$

Fig. 3. The model checking game rules for FLC.

Player \exists wins the play C_0, C_1, \dots of $\mathcal{G}(t, \varphi)$ if

- (1) there is an $n \in \mathbb{N}$ s.t. $C_n = w, \delta \vdash p$ for some w and δ , s.t. $t(w) = p$, or
- (2) it is infinite and its outermost stack-increasing variable is of type ν .

Player \forall wins the play if

- (3) there is an $n \in \mathbb{N}$ s.t. $C_n = w, \delta \vdash p$ for some w and δ , s.t. $t(w) \neq p$, or
- (4) it is infinite and its outermost stack-increasing variable is of type μ .

3.2. Correctness

Take any $\varphi_0 \in \text{FLC}$. An *unfolding tree* of φ_0 is a ranked tree T with nodes labeled by subformulas of φ_0 which satisfies the following.

- (1) The root of T is labeled with φ_0 .
- (2) If a node is labeled $\psi_1 \vee \psi_2$ or $\psi_1 \wedge \psi_2$ then it has one successor labeled ψ_1 or ψ_2 .
- (3) If a node is labeled $\mu X. \varphi$ or $\nu X. \varphi$ then it has one successor labeled X .
- (4) If a node is labeled X and $fp_{\varphi_0}(X) = \sigma X. \psi$ then it has one successor labeled ψ .
- (5) If a node is labeled $\psi_1; \psi_2$ then it has two successors: the left one is labeled ψ_1 and the right one is labeled ψ_2 .

A *partial* unfolding tree allows the right son of a node labeled $\psi_1; \psi_2$ to be a leaf provided that the same holds for all such nodes above this one. A (partial) unfolding tree is called *tagged* if some of its nodes are tagged with a natural number. In the following we will simply speak of an unfolding tree instead of a tagged partial unfolding tree.

Every play $\pi = C_0, C_1, \dots$ in a game $\mathcal{G}(t, \varphi_0)$ defines an unfolding tree T_π : it is constructed in a left-depth-first manner using a control stack of nodes in T_π . Starting with the actual configuration C_0 , the root of T_π as the actual node, and the empty control stack, proceed as follows.

Let $C_i = w_i, \delta_i \vdash \psi_i$ be the actual configuration. Label the actual node n of T_π with ψ_i and give it the tag i . For as long as there is a successor configuration C_{i+1} continue with it and

- the son of n if the rule that applies in C_i is (\vee) , (\wedge) , (FP) , or (V) .
- the left son of n if the rule that applies in C_i is $(;)$; push the right son onto the stack.
- the node popped from the top of the stack in any other case.

The following facts about an unfolding tree T_π are easy to see.

Fact 4.

- (a) No two nodes in T_π have the same tag.
- (b) T_π has at most one tagged infinite branch.
- (c) A node with tag m is below or right of a node with tag n iff $n \leq m$.

Because of (a) we write $T_\pi(i)$ for any $i \in \mathbb{N}$ to denote the unique node with the tag i . The parts (b) and (c) are due to the fact that T_π is obtained in a left-depth-first manner.

Example 5. Take the infinite binary tree t whose root is labeled b and all of whose other nodes are labeled a . Take the Σ_2^{syn} formula $\varphi = \mu Y.(a \wedge \diamond) \vee \square; (\nu X.Y; X; Y)$ and consider the game $\mathcal{G}(t, \varphi)$. In order to avoid defeat by reaching a , player \exists first chooses the right disjunct. Note that player \forall 's choices with rule (\square) are irrelevant since no node in t has two different subtrees. Also, he immediately loses when he chooses the conjunct a anywhere other than at the root of t . An infinite play of $\mathcal{G}(t, \varphi)$ is sketched in Fig. 4. The left column contains a symbolic name and a label for the respective configuration. The need for the latter will be explained in Section 3.3.

$C_0, e_1 :$	$\epsilon, \text{tt} \vdash \mu Y.(a \wedge \diamond) \vee \square; (\nu X.Y; (X; Y))$
$C_1, e_2 :$	$\epsilon, \text{tt} \vdash Y$
$C_2, e_1 :$	$\epsilon, \text{tt} \vdash (a \wedge \diamond) \vee \square; (\nu X.Y; (X; Y))$
$C_3, d :$	$\epsilon, \text{tt} \vdash \square; (\nu X.Y; (X; Y))$
$C_4, a :$	$\epsilon, (\nu X.Y; (X; Y)); \text{tt} \vdash \square$
$C_5, e_1 :$	$0, \text{tt} \vdash \nu X.Y; (X; Y)$
$C_6, e_1 :$	$0, \text{tt} \vdash X$
$C_7, d :$	$0, \text{tt} \vdash Y; (X; Y)$
$C_8, e_2 :$	$0, (X; Y); \text{tt} \vdash Y$
$C_9, e_1 :$	$0, (X; Y); \text{tt} \vdash (a \wedge \diamond) \vee \square; (\nu X.Y; (X; Y))$
$C_{10}, a_1 :$	$0, (X; Y); \text{tt} \vdash a \wedge \diamond$
$C_{11}, e :$	$0, (X; Y); \text{tt} \vdash \diamond$
$C_{12}, d :$	$00, \text{tt} \vdash X; Y$
$C_{13}, e_1 :$	$00, Y; \text{tt} \vdash X$
$C_{14}, d :$	$00, Y; \text{tt} \vdash Y; (X; Y)$
$C_{15}, e_2 :$	$00, (X; Y); Y; \text{tt} \vdash Y$
$C_{16}, e_1 :$	$00, (X; Y); Y; \text{tt} \vdash (a \wedge \diamond) \vee \square; (\nu X.Y; (X; Y))$
$C_{17}, a_1 :$	$00, (X; Y); Y; \text{tt} \vdash a \wedge \diamond$
$C_{18}, e :$	$00, (X; Y); Y; \text{tt} \vdash \diamond$
$C_{19}, d :$	$000, Y; \text{tt} \vdash X; Y$
$C_{20}, e_1 :$	$000, Y; Y; \text{tt} \vdash X$
	\vdots

Fig. 4. An infinite play of Example 5.

In this play, both X and Y occur infinitely often in the principal position. Note that neither $fp_\varphi(X)$ nor $fp_\varphi(Y)$ occur infinitely often. Furthermore, we have $X <_\varphi Y$. Thus, the outermost variable occurring infinitely often in this play is Y which is of type μ . But the outermost stack-increasing variable is X which is of type ν . Therefore, player \exists wins this play. In fact, $t \models \varphi$ and the strategy described above is a winning strategy for player \exists in the game $\mathcal{G}(t, \varphi)$.

The fact that here it is X rather than Y that determines the winner—unlike in the case of the model checking games for the modal μ -calculus—can be explained as follows. The semantics of both Y and X are functions of type $\mathbb{D} \rightarrow \mathbb{D}$. Each unfolding in the play creates an approximant to this function which, in turn, is also such a function. Consider the second occurrence of Y in principal position. This can be seen as a query asking whether 0 is included in the value of Y at the argument $\llbracket X; Y; \text{tt} \rrbracket_\rho$ for some appropriate ρ . Although Y is defined recursively, the next occurrence of it in principal position simply asks for the value of the same approximant but at another argument.

X , however, is stack-increasing. Consider also the difference to Y in the unfolding tree in Fig. 5: there is no branch on which Y occurs infinitely often. X , however, does occur infinitely often on the branch that is abbreviated at the bottom. It shows that every corresponding occurrence of X in principal position represents another approximation to the value of X relative to the same value for Y .

Lemma 6. *Let $\pi = C_0, C_1, \dots$ be a play of $\mathcal{G}(t, \varphi)$ with $C_i = w_i, \delta_i \vdash \psi_i$ for all $i \in \mathbb{N}$. For all $n, m \in \mathbb{N}$ with $n \leq m$ we have: $\delta_n \preceq \delta_k$ for all k with $n < k \leq m$ iff $T_\pi(m)$ is a successor of $T_\pi(n)$.*

Proof. (\Rightarrow) by contradiction. Suppose $T_\pi(m)$ is not a successor of $T_\pi(n)$. Since $m \geq n$, and in fact $m > n$ in this case, $T_\pi(m)$ must be right of $T_\pi(n)$ in T_π . Remember that T_π is built in a left-depth-first fashion. But then there must be a predecessor of $T_\pi(n)$ with a right son that is a predecessor of $T_\pi(m)$. Moreover, this node must have a tag k' . Now, since $T_\pi(k')$ is right of $T_\pi(n)$ but above $T_\pi(m)$ we have $n \leq k' \leq m$.

Finally, since $T_\pi(k')$ is right of $T_\pi(n)$ it was pushed onto the control stack before $T_\pi(n)$. Hence, it could only be tagged after the top element of the control stack present at the moment that $T_\pi(n)$ got tagged was popped. Now note that a node with tag i gets pushed onto, respectively, popped from the control stack iff the label of this node gets pushed onto, respectively, popped from the

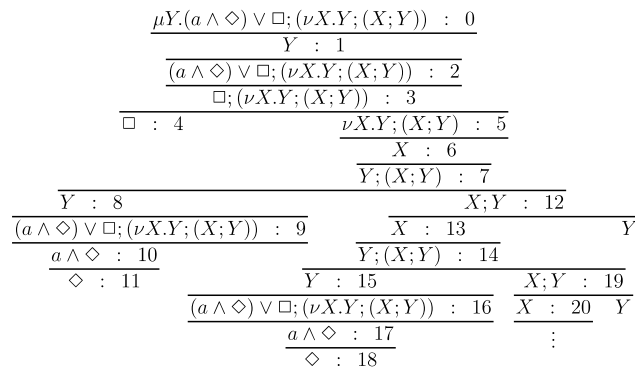


Fig. 5. The unfolding tree for the play in Fig. 4.

game stack δ_i in configuration C_i . Hence, the construction of T_π has reached a node with tag k , s.t. $n < k \leq k'$ and $\delta_k < \delta_n$, i.e. $\delta_n \not\leq \delta_k$.

Thus, if $T_\pi(m)$ is not a successor of $T_\pi(n)$ then there must be a k s.t. $n < k \leq m$ and $\delta_n \not\leq \delta_k$, which proves half of the claim.

(\Leftarrow) Now suppose that $T_\pi(m)$ is below $T_\pi(n)$ but that there is a k' with $n < k' \leq m$ s.t. $\delta_n \not\leq \delta_{k'}$. As in the first part, there must be a k s.t. $n < k \leq k'$ and $\delta_k < \delta_n$. But then $T_\pi(k)$ is right of $T_\pi(n)$ and, since $k \leq m$, $T_\pi(m)$ is right or below of $T_\pi(k)$. This contradicts the assumption that $T_\pi(m)$ is below $T_\pi(n)$. \square

A simple consequence of this lemma is the following.

Corollary 7. *X is stack-increasing in the play π iff X occurs infinitely often on a tagged branch in T_π .*

Lemma 8. *Every play has a unique winner.*

Proof. A play π can either be finite or infinite. It is only finite if no further rule applies to a configuration, but then either winning condition (1) or (3) applies. Note that they are mutually exclusive.

Now let π be of infinite length. For each $i \in \mathbb{N}$ there is a node with tag i in its corresponding unfolding tree T_π . This tree is finitely branching and, by König's Lemma, has an infinite branch. This branch must have infinitely many labels that are variables for otherwise the labels would eventually shrink in size and become atomic propositions. However, those cannot occur on an infinite branch. Since the underlying formula φ_0 only contains finitely many variables, there must be at least one variable that occurs infinitely often on this branch.

Now note that if two variables X and Y occur on one branch then we have $X <_{\varphi_0} Y$ or $Y <_{\varphi_0} X$ for some input formula φ_0 . Hence, there is an outermost variable X that occurs infinitely often on this branch. According to Corollary 7, X is stack-increasing in π . Fact 4 says that there is at most one tagged infinite branch in T_π . Hence, any other stack-increasing variable in π is smaller than X w.r.t. $<_{\varphi_0}$. But then the fixpoint type of the unique outermost stack-increasing variable uniquely determines the winner of π . \square

Theorem 9. *For all $t \in \mathcal{T}_P$ and all closed $\varphi \in \text{FLC}$: player \exists wins the game $\mathcal{G}(t, \varphi)$ iff $t \models \varphi$.*

Proof. (\Leftarrow) Suppose $t \models \varphi$. We need to describe a winning strategy for player \exists in the game $\mathcal{G}(t, \varphi)$. Intuitively, she preserves truth along each play. Since configurations in this game can contain free variables in principal position or on the stack, we need to define a notion of truth that interprets free variables. We let player \exists use μ -signatures for their interpretation and, at the same time, to guide her choices in a truth-preserving way.

Note that each play starts with the configuration $C = \epsilon, \text{tt} \vdash \varphi$ and, by assumption, we have $t \models \varphi$ which entails $t \models \varphi; \text{tt}$. In general, let $C = w, \delta \vdash \psi$ be a configuration, in which each free occurrence of a μ -variable X in δ or ψ is annotated with a μ -signature η . We will call C *true*, if $w \in \|\psi; \delta\|^t$ where each such X is interpreted by the approximant X^η . We will also use ρ to denote such an interpretation, and write $w \in \|\psi; \delta\|_\rho^t$ for example. Note that such a ρ interprets occurrences of variables. Hence, the term $\rho(X)$ can be ambiguous since the variable X can occur several times in C . For example, it could be put onto the stack and into principal position. The former remains whereas the latter gets unfolded. Then these two occurrences represent different approximants. Nev-

ertheless we will use this notation as well as the update $\rho[X \mapsto X^\eta]$ since the relevant occurrence of variable X will easily be derivable from the context.

First we observe that player \exists can indeed preserve truth in each play, and that player \forall must preserve truth with his choices.

- The starting configuration $\epsilon, \text{tt} \vdash \varphi$ contains no free variables and is true under any ρ by assumption.
- If the actual configuration is $w, \delta \vdash \psi_1 \vee \psi_2$ and $w \in \|(\psi_1 \vee \psi_2); \delta\|_\rho^t$ for some ρ then there is an $i \in \{1, 2\}$ s.t. the successor configuration is $w, \delta \vdash \psi_i$ with $w \in \|\psi_i; \delta\|_\rho^t$ for the same ρ .
- If the actual configuration is $w, \delta \vdash \psi_1 \wedge \psi_2$ and $w \in \|(\psi_1 \wedge \psi_2); \delta\|_\rho^t$ for some ρ then for both possible successor configurations $w, \delta \vdash \psi_i, i \in \{1, 2\}$ we have $w \in \|\psi_i; \delta\|_\rho^t$ for the same ρ .
- If the actual configuration is $w, \psi; \delta \vdash \Diamond$ with $w \in \|\Diamond; \psi; \delta\|_\rho^t$ for some ρ then there is an $i \in \{0, 1\}$ s.t. $wi \in \|\psi; \delta\|_\rho^t$, i.e. the corresponding successor configuration $wi, \delta \vdash \psi$ is true under the same ρ .
- If the actual configuration is $w, \psi; \delta \vdash \Box$ and $w \in \|\Box; \psi; \delta\|_\rho^t$ then $wi \in \|\psi; \delta\|_\rho^t$ for both $i \in \{0, 1\}$, i.e. both successor configurations are true under the same ρ .
- The rules (τ) , and $(;)$ trivially preserve truth under the same environment ρ .
- Consider rule (FP) which is to be applied to a configuration $w, \delta \vdash \nu X. \psi$ s.t. $w \in \|(\nu X. \psi); \delta\|_\rho^t$. Hence, $w \in \|X; \delta\|_{\rho'}^t$ where ρ' updates ρ for the new occurrence of a free variable X : $\rho[X \mapsto \|\nu X. \psi\|_\rho^t]$. Note that $\|\nu X. \psi\|_\rho^t \subseteq \lambda T. \{0, 1\}^*$, and, by monotonicity, the successor configuration $w, \delta \vdash X$ is true under $\rho[X \mapsto \lambda T. \{0, 1\}^*]$.
- Consider rule (FP) in a configuration $w, \delta \vdash \mu X. \psi$ which is true under the environment ρ , i.e. $w \in \|(\mu X. \psi); \delta\|_\rho^t$. The following configuration is $C' = w, \delta \vdash X$ which contains a new occurrence of a free variable. According to Lemma 2 and the remark following it, there is a μ -signature η s.t. C' is true under the interpretation $\rho[X \mapsto X^\eta]$. Note that this new interpretation leaves the other occurrences of the variable X in δ untouched. Furthermore, μ -signatures are well-founded and by monotonicity we can assume η to be the lexicographically smallest that makes C' true.
- Finally, consider rule (\vee) which is applied to a configuration $w, \delta \vdash X$, s.t. $w \in \|X; \delta\|_\rho^t$. Let $fp_\varphi(X) = \sigma X. \psi$ for some ψ . If $\sigma = \nu$ then $\rho(X) = \lambda T. \{0, 1\}^*$ and, by monotonicity, the following configuration $C' = w, \delta \vdash \psi$ is true under ρ as well. Assume therefore $\sigma = \mu$, and let $\rho(X) = X^{(\alpha_1, \dots, \alpha_n)}$ for this occurrence of X . By the definition of approximants, α_n cannot be a limit ordinal. Hence, it must be a successor ordinal $\alpha_n = \beta + 1$. Let $\rho' := \rho[X \mapsto X^{(\alpha_1, \dots, \beta)}]$. But then the following configuration C' is true under ρ' .

This defines a simple strategy for player \exists : whenever a subformula of the form $\mu X. \psi$ occurs in principal position, she takes the least μ -signature η according to Lemma 2 and annotates this occurrence of X with η . If a μ -variable is in principal position then it must have an annotation η . When it is unfolded she decreases the last component of η . If this creates a limit ordinal she can replace it by a smaller successor ordinal. Whenever she has to perform a choice with rule (\vee) or (\Diamond) she simply preserves truth under the interpretation given by her annotations.

It remains to be shown that this is indeed a winning strategy. For the sake of contradiction suppose that player \forall plays with his best strategy against this strategy. The result is a single play

π . Remember that every configuration in π is true under the interpretation that is given by the annotations at each occurrence of a variable.

But then player \forall cannot win π with his winning condition (3) because this would require him to reach a configuration that is blatantly false under any interpretation of the free variables. Suppose therefore that his winning condition (4) applies, i.e. the outermost stack-increasing variable X in π is of type μ . Take the last configuration C_{i_1} after which no outer variable than X occurs in a stack-increasing fashion. Remember that it has been annotated with some η . Since X is stack-increasing there are further C_{i_2}, C_{i_3}, \dots with X in their principal position. According to Corollary 7 they induce an infinite branch in the unfolding tree T_π . If there was another stack-increasing variable Y then it would induce the same branch. Since by assumption there is no such Y , this branch in the unfolding tree witnesses an infinite unfolding of a least fixpoint variable. Hence, by the well-foundedness of the ordinals, there would be a configuration $C_{i_k} = w_{i_k}, \delta_{i_k} \vdash X$ for some k that is true under the interpretation that maps X to some $(\alpha_1, \dots, 0)$ which is impossible.

Lemma 8 says that the play is won by player \exists if it is not won by player \forall . We conclude that player \exists 's strategy must in fact be a winning strategy.

(\Rightarrow) This can be proved analogously to the completeness part above. Suppose $t \not\models \varphi$. Now player \forall has a strategy which consists of preserving falsity as well as annotating the occurrences of v -variables with v -signatures.

As above, player \exists cannot win against this strategy for it would contradict the preservation of falsity or Lemma 2. But by the definition of winning strategy, player \exists cannot have one for the game $\mathcal{G}(t, \varphi)$. \square

3.3. Games as binary trees

For any $n \in \mathbb{N}$ let $\mathcal{P}_n := \{e, a, a_1, t, f, d\} \cup \{e_i \mid i \in \{1, \dots, n\}\}$. Next we will modify the rules and winning conditions of the model checking games s.t. the representation of a game $\mathcal{G}(t, \varphi)$ on any $t \in \mathcal{T}_{\mathcal{P}}$ and any $\varphi \in \Sigma_n^{\text{syn}}$ is again an infinite binary tree in $\mathcal{T}_{\mathcal{P}_n}$.

Note that $\mathcal{G}(t, \varphi)$ is a tree, namely the tree of its configurations with the starting configuration as the root and successor configurations as sons in the tree. In order to make it an infinite binary tree we replace the rules for fixpoint quantifiers, variables, sequential compositions and the atomic formula τ by the following ones. They simply duplicate premisses and make player \exists choose one of the identical copies.

$$\frac{w, \delta \vdash \sigma Z. \psi}{w, \delta \vdash Z \quad w, \delta \vdash Z} \quad \exists \quad \frac{w, \delta \vdash Z}{w, \delta \vdash \psi \quad w, \delta \vdash \psi} \quad \exists, \text{ if } fp_\varphi(Z) = \sigma Z. \psi$$

$$\frac{w, \delta \vdash \psi_0; \psi_1}{w, \psi_1; \delta \vdash \psi_0 \quad w, \psi_1; \delta \vdash \psi_0} \quad \exists \quad \frac{w, \psi; \delta \vdash \tau}{w, \delta \vdash \psi \quad w, \delta \vdash \psi} \quad \exists$$

Furthermore, the following rule is added to the game. It ensures that $\mathcal{G}(t, \varphi)$ becomes an infinite tree.

$$\frac{w, \delta \vdash q}{w, \delta \vdash q \quad w, \delta \vdash q} \quad \exists$$

It should be clear that Theorem 9 is still valid under the amended interpretation of $\mathcal{G}(t, \varphi)$. Finally, we define the labeling of the trees' nodes as follows. Let $t' = \mathcal{G}(t'', \varphi)$ for some t'' and φ . Assume n still to be fixed through the choice of \mathcal{P}_n .

$$\begin{aligned}
t'(w, \delta \vdash q) &:= \begin{cases} t & \text{if } t''(\epsilon) = q \\ f & \text{o.w.} \end{cases} \\
t'(w, \delta \vdash X) &:= e_i \quad \text{where } i = \text{lvl}_\varphi(X) \\
t'(w, \delta \vdash \diamond) &:= e \\
t'(w, \delta \vdash \square) &:= a \\
t'(w, \delta \vdash \tau) &:= e \\
t'(w, \delta \vdash \psi_1 \vee \psi_2) &:= e_1 \\
t'(w, \delta \vdash \psi_1 \wedge \psi_2) &:= a_1 \\
t'(w, \delta \vdash \sigma X. \psi) &:= e_1 \\
t'(w, \delta \vdash \psi_1; \psi_2) &:= d
\end{aligned}$$

Hence, e and a mark those configurations in which player \exists , respectively, \forall makes a choice and a formula is popped; t and f mark configurations that are true, respectively, false; e_i and a_1 mark choices by either of the players that continue without popping a formula from the stack; and d marks configurations in which a formula gets pushed onto the stack.

Fact 10. For any $t \in \mathcal{T}_{\mathcal{P}}$ and any $\varphi \in \Sigma_n^{\text{syn}}$ we have $\mathcal{G}(t, \varphi) \in \mathcal{T}_{\mathcal{P}_n}$.

With games being trees themselves, we can interpret FLC formulas over them, i.e. play the model checking game on such trees. This bears some notational difficulties. In order to simplify this we introduce the following convention. Let $t' := \mathcal{G}(t, \varphi)$ be the game tree for some other tree t and some $\varphi \in \text{FLC}$. Let φ' be another FLC formula. Configurations of the game $\mathcal{G}(t', \varphi')$ are, of course, of the form $w, \delta \vdash \psi$. Note that w determines a configuration C of the game $\mathcal{G}(t, \varphi)$, namely the unique configuration at position w in t' . Since C will often be more meaningful than w in such a context, we will allow ourselves to write $C, \delta \vdash \psi$ for a configuration in $\mathcal{G}(\mathcal{G}(t, \varphi), \varphi')$ instead.

4. The alternation hierarchy

4.1. Game formulas

The proof of the hierarchy theorem relies on the fact that FLC is closed under complementation. This is not true w.r.t. the strong equivalence relation. Clearly there is no formula φ whose semantics is the complement of the identity function $\llbracket \tau \rrbracket^t$ because it is not monotone.

However, FLC is closed under complementation w.r.t. weak equivalence. It is possible to find, for any φ , a dual counterpart that defines exactly the complement of the tree models of φ .

Lemma 11. For every $n \in \mathbb{N}$ and every $\varphi \in \Pi_n^{\text{syn}}$ there is a $\bar{\varphi} \in \Sigma_n^{\text{syn}}$, s.t. for all $t \in \mathcal{T}_{\mathcal{P}} : t \models \varphi$ iff $t \not\models \bar{\varphi}$.

Proof. First we define for each $\varphi \in \text{FLC}$ an auxiliary formula φ' as

$$\begin{array}{ll} q' &:= \bigvee_{p \neq q} p \\ \tau' &:= \tau \\ X' &:= X \\ \diamond' &:= \square \\ \square' &:= \diamond \end{array} \quad \begin{array}{ll} (\psi_0 \vee \psi_1)' &:= \psi'_0 \wedge \psi'_1 \\ (\psi_0 \wedge \psi_1)' &:= \psi'_0 \vee \psi'_1 \\ (\psi_0; \psi_1)' &:= \psi'_0; \psi'_1 \\ (\mu X. \psi)' &:= \nu X. \psi' \\ (\nu X. \psi)' &:= \mu X. \psi' \end{array}$$

For every $T \in \mathbb{D}$ we use \bar{T} to denote $\{0, 1\}^* \setminus T$. A straight-forward induction on the formula structure then shows that we have $\llbracket \varphi \rrbracket_{\rho}^t(T) = \llbracket \varphi' \rrbracket_{\rho}^t(\bar{T})$ for all $t \in \mathcal{T}_{\mathcal{P}}$, all $T \in \mathbb{D}$, all $\varphi \in \text{FLC}$ and all environments ρ .

Hence, the complement of φ w.r.t. weak equivalence can be defined as $\bar{\varphi} := \varphi'; \text{ff}$. Then we have $t \models \varphi$ iff $t \not\models \bar{\varphi}$ for all $t \in \mathcal{T}_{\mathcal{P}}$. Note that $\varphi \approx \varphi; \text{tt}$. \square

We note that it is also possible to prove Lemma 11 using Theorem 9. The game $\mathcal{G}(t, \bar{\varphi})$ is the dual game of $\mathcal{G}(t, \varphi)$ in which the players' choices and winning conditions are swapped. Hence, player \exists wins the game $\mathcal{G}(t, \varphi)$ iff player \forall wins the game $\mathcal{G}(t, \bar{\varphi})$ from which the claim follows, too.

Next we will introduce formulas that later we will show to be complete for each Σ_n^{w} -level of the alternation hierarchy. These describe exactly those game trees that are winning for player \exists on appropriate formulas.

Definition 12. For any $n \in \mathbb{N}$ let

$$\begin{aligned} \Phi_n &:= \mu X_n. \nu X_{n-1}. \mu X_{n-2}. \dots. \sigma X_1. t \vee \left(\bar{f} \wedge (e \rightarrow \diamond) \wedge (a \rightarrow \square) \wedge (d \rightarrow (\diamond; X_1); X_1) \right. \\ &\quad \left. \wedge (a_1 \rightarrow \square; X_1) \wedge \bigwedge_{i=0}^n (e_i \rightarrow \diamond; X_i) \right) \end{aligned}$$

Fact 13. For all $n \in \mathbb{N}$ we have $\Phi_n \in \Sigma_n^{\text{syn}}$.

Consider the model checking game played on any tree t' and Φ_n . Note that because of Φ_n 's structure, any play of $\mathcal{G}(t', \Phi_n)$ proceeds as follows.

- (1) Depending on which variable occurred last, the play enters the fixpoint prefix somewhere and proceeds through all the fixpoint quantifiers further inside.
- (2) Player \exists can choose the proposition t or the big conjunction. In the former case the play finishes immediately.
- (3) Player \forall chooses one of the conjuncts. The play is finished only if it is the first.
- (4) Player \exists chooses one of the disjuncts presented as implications. Again, the play is either finished or continues in the same manner. The latter might involve a choice through a modal formula, and a pushing onto or a popping from the stack.

Our goal is to show that Φ_n describes exactly those trees that are winning games for player \exists . In order to lead up to this we recall Example 5.

$C_0, \text{tt} \vdash \mu X_2. \nu X_1. \dots$	$C_{11}, X_1; \text{tt} \vdash X_1$
\vdots	\vdots
$C_0, \text{tt} \vdash \overline{e}_1 \vee \diamond; X_1$	$C_{11}, X_1; \text{tt} \vdash \diamond$
\vdots	$C_{12}, \text{tt} \vdash X_1$
$C_1, \text{tt} \vdash X_1$	\vdots
\vdots	$C_{13}, X_1; \text{tt} \vdash X_1$
$C_1, \text{tt} \vdash \overline{e}_2 \vee \diamond; X_2$	\vdots
\vdots	$C_{14}, X_1; \text{tt} \vdash X_1$
$C_2, \text{tt} \vdash X_2$	\vdots
\vdots	$C_{15}, X_1; \text{tt} \vdash X_1$
$C_7, \text{tt} \vdash X_1$	\vdots
\vdots	$C_{16}, X_1; X_1; \text{tt} \vdash X_2$
$C_7, \text{tt} \vdash (\diamond; X_1); X_1$	\vdots
$C_7, X_1; \text{tt} \vdash \diamond; X_1$	$C_{17}, X_1; X_1; \text{tt} \vdash X_1$
\vdots	\vdots
$C_7, X_1; X_1; \text{tt} \vdash \diamond$	$C_{18}, X_1; X_1; \text{tt} \vdash \diamond$
$C_8, X_1; \text{tt} \vdash X_1$	$C_{19}, X_1; \text{tt} \vdash X_1$
\vdots	\vdots
$C_8, X_1; \text{tt} \vdash \diamond; X_2$	$C_{19}, X_1; \text{tt} \vdash (\diamond; X_1); X_1$
$C_8, X_2; X_1; \text{tt} \vdash \diamond$	\vdots
$C_9, X_1; \text{tt} \vdash X_2$	$C_{20}, X_1; X_1; \text{tt} \vdash X_1$
\vdots	\vdots

Fig. 6. Excerpts of a play involving the game formula Φ_2 .

Example 14. Let t and φ be as defined in Example 5. In particular, $\varphi = \mu Y. (a \wedge \diamond) \vee \square; (\nu X. Y; X; Y)$. Let $\pi' = C_0, C_1, \dots$ be the play of $\mathcal{G}(t, \varphi)$ that is presented in Fig. 4. Note that $\varphi \in \Sigma_2^{\text{syn}}$. Fig. 6 sketches a play π of $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_2)$ that corresponds to π' in the sense that the projection of π onto its first component yields π' . In π , both players avoid sudden defeat by never choosing an atomic proposition.

Recall that the outermost stack-increasing variable in π is X , not Y . Furthermore, $\text{lvl}_\varphi(X) = 1$ and $\text{lvl}_\varphi(Y) = 2$. The excerpts of π presented in Fig. 6 show that, consequently, X_1 is stack-increasing whereas X_2 is not.

Suppose that $\varphi \in \Sigma_n^{\text{syn}}$ and that player \exists has a winning strategy ζ' for the game $\mathcal{G}(t, \varphi)$ on some tree t . Now consider the game $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$. We describe a strategy ζ for player \exists in this game. It simply makes player \exists choose, respectively, avoid those disjuncts that are obviously true, respectively, false, and appeals to ζ' in case of \diamond -formulas.

Let ζ' be a strategy for player \exists in the game $t' := \mathcal{G}(t, \varphi)$ with $\varphi \in \Sigma_n^{\text{syn}}$. Note that ζ' maps positions w in t' , s.t. player \exists has to perform a choice in the corresponding configuration C , to successor positions wi , $i \in \{0, 1\}$. A position in t' uniquely defines a configuration. Hence, we simply regard the strategy ζ' as a function that maps configurations to configurations.

Definition 15. Let ζ' be a strategy for player \exists in the game $t' := \mathcal{G}(t, \varphi)$ with $\varphi \in \Sigma_n^{\text{syn}}$. Define a strategy ζ for player \exists in the game $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ as follows.

$$\begin{aligned}
\zeta(w, \delta \vdash q \vee \psi) &:= \begin{cases} w, \delta \vdash q, & \text{if } t'(w) = q \\ w, \delta \vdash \psi, & \text{o.w.} \end{cases} \\
\zeta(w, \delta \vdash \overline{q} \vee \psi) &:= \begin{cases} w, \delta \vdash \overline{q}, & \text{if } t'(w) \neq q \\ w, \delta \vdash \psi, & \text{o.w.} \end{cases} \\
\zeta(w, \psi; \delta \vdash \diamond) &:= \zeta'(w), \delta \vdash \psi
\end{aligned}$$

For the last case note that a configuration of the form $w, \psi; \delta \vdash \Diamond$ with configuration $C = w', \delta' \vdash \psi'$ at position w is only reachable in $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ if ψ' requires player \exists to make a choice in C in $\mathcal{G}(t, \varphi)$. Otherwise, $\zeta'(w)$ would be undefined.

Lemma 16. *Let $\varphi \in \Sigma_n^{\text{syn}}$ and $t \in \mathcal{T}_P$. For every play π of $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ there is a unique play π' of $\mathcal{G}(t, \varphi)$, s.t. π conforms to ζ iff π' conforms to ζ' . Moreover, the mapping \cdot' is injective.*

Proof. Let π be a play of $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$. The projection onto its first component yields a sequence of configurations of $\mathcal{G}(t, \varphi)$. Collapsing adjacent and equal configurations in this sequence yields a play π' of $\mathcal{G}(t, \varphi)$. The play π conforms to ζ iff the play π' conforms to ζ' simply by the definition of ζ .

What remains to be shown is that π can be reconstructed from π' . Remember the description of how a play in $\mathcal{G}(t', \Phi_n)$ proceeds for any t' . Note that there is only one possibility to obtain an infinite play and this is done if both players avoid defeat by never choosing an atomic proposition that is currently false. Clearly, the projection onto the first component and collapsing yields the play π' again. Hence, every π' also uniquely determines a play π of $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$. \square

In the following, we will assume π' and π both to be infinite plays that correspond to each other in the described way.

Next we consider a binary relation R between configurations in π' and configurations in π defined as

$$R(C', (C, \delta \vdash \psi)) \quad \text{iff} \quad C = C'$$

For every configuration C' let $\text{lst}(C')$ denote the last configuration C (in the natural occurrence order) in π s.t. $R(C', C)$. Note that there need not necessarily be a last one because of the added game rules that simply replicate configurations. This, however, is only possible if eventually all configurations are the same. In this case let $\text{lst}(C')$ be this unique configuration. Similarly, let $\text{fst}(C')$ be the first configuration C s.t. $R(C', C)$. Note that $\text{fst}(C')$ and $\text{lst}(C')$ always exist if π' is an infinite play.

Lemma 17. *For any C' , let $\delta_1, \dots, \delta_m$ be all the stacks occurring in configurations between and including $\text{fst}(C')$ and $\text{lst}(C')$. Then for all $i = 1, \dots, m$: $\delta_1 \preceq \delta_i$.*

Proof. Remember the above description of how a play in $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ proceeds. The configuration $\text{fst}(C')$ has in its principal position either Φ_n itself or a variable X_k . From then on, rules (FP) and (V) are played possibly several times, then (\vee) , then (\wedge) , (\vee) again, and finally $(;)$ once, twice or not at all. Then, $\text{lst}(C')$ is reached. Note that the stack remains unchanged until, in the end, something possibly gets pushed onto it with rule $(;)$. Popping an element from the stack is only possible in the transition from a $\text{lst}(C')$ to a $\text{fst}(C'')$ where C'' is the successor configuration of C' . Hence, between $\text{fst}(C')$ and $\text{lst}(C')$ the stack always is an extension of the stack in $\text{fst}(C')$. \square

Let $C = w, \delta \vdash \psi$ be a configuration in a play π of some game $\mathcal{G}(t, \varphi)$. We write $\|C\|$ for the size of its stack, i.e. $\|C\| = 1 + n - m$ where n is the number of rules that push something and m is the number of rules that pop something played in π upto C .

Lemma 18. *Let $\pi' = C_0, C_1, \dots$. For all $i \in \mathbb{N}$ we have: $\|C_{i+1}\| - \|C_i\| = \|fst(C_{i+1})\| - \|fst(C_i)\|$.*

Proof. By case distinction on the rule that transforms C_i into C_{i+1} . Suppose it is rule (\vee) , (\wedge) , (FP) or (V) . Then C_i is labeled with some e_j or a_1 , and $\|C_{i+1}\| - \|C_i\| = 0$. Remember that both players are assumed to avoid defeat in π that contains $fst(C_i)$ and $fst(C_{i+1})$. Let $fst(C_i) = C_i, \delta \vdash \psi$ for some δ and ψ . If $i = 0$ then $\psi = \Phi_n$, otherwise $\psi = X_k$ for some k . But then the play proceeds with the application of rules (V) or (FP) possibly repeatedly, followed by rules (\vee) , (\wedge) , and (\vee) again to a configuration $C_i, \delta \vdash \diamond; X_j$ or $C_i, \delta \vdash \square; X_j$. Note that none of these rules changes the stack. Subsequently, after applying rule $(;)$ and then (\diamond) or (\square) it reaches $C_{i+1}, \delta \vdash X_j = fst(C_{i+1})$. Hence, $\|fst(C_{i+1})\| - \|fst(C_i)\| = 0$.

Similarly, if the rule that applies in C_i is (\diamond) , (\square) , or (τ) , then $\|C_{i+1}\| - \|C_i\| = -1$. Moreover, let $fst(C_i) = C_i, \psi'; \delta \vdash \psi$. Because the label of C_i is a or e we have $fst(C_{i+1}) = C_{i+1}, \delta \vdash \psi'$ and, thus, $\|fst(C_{i+1})\| - \|fst(C_i)\| = -1$.

Finally, the last remaining case is that of rule $(;)$ which yields $\|C_{i+1}\| - \|C_i\| = 1$. Here, the label of C_i is d , and if $fst(C_i) = C_i, \delta \vdash \psi$ then π contains the fragment

$$\begin{array}{ll}
 C_i, \delta \vdash \psi & = \quad fst(C_i) \\
 \vdots & \\
 C_i, \delta \vdash d \rightarrow (\diamond; X_1); X_1 & \\
 C_i, \delta \vdash (\diamond; X_1); X_1 & \\
 C_i, X_1; \delta \vdash \diamond; X_1 & \\
 C_i, X_1; X_1; \delta \vdash \diamond & = \quad lst(C_i) \\
 C_{i+1}, X_1; \delta \vdash X_1 & = \quad fst(C_{i+1})
 \end{array}$$

Hence, $\|fst(C_{i+1})\| - \|fst(C_i)\| = 1$. \square

Theorem 19. *For every $n \in \mathbb{N}$, every $\varphi \in \Sigma_n^{\text{syn}}$ and every $t \in \mathcal{T}_{\mathcal{P}}$ for some \mathcal{P} we have: if $t \models \varphi$ then $\mathcal{G}(t, \varphi) \models \Phi_n$.*

Proof. Suppose $\varphi \in \Sigma_n^{\text{syn}}$ and $t \models \varphi$. According to Theorem 9, player \exists has a winning strategy ζ' for the game $\mathcal{G}(t, \varphi)$. By Fact 10, this forms a binary tree in $\mathcal{T}_{\mathcal{P}_n}$ itself, and we abbreviate $t' := \mathcal{G}(t, \varphi)$. Now consider the strategy ζ for player \exists in the game $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ as constructed in Definition 15. It remains to be seen that this strategy is winning.

Assume therefore that player \forall plays against ζ with his best strategy. As said above, the result is a single play π in $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ which defines—by a projection—the single play π' in $\mathcal{G}(t, \varphi)$. According to Fact 16, π' conforms to ζ' and is thus winning for player \exists . There are two possibilities depending on the winning condition that applies.

If she wins π' with her winning condition (1), i.e. by reaching an atomic propositions which is true then she does so in π , namely the proposition t . Suppose therefore that she wins with her winning condition (2), i.e. the outermost stack-increasing variable X is of type v . Then $lvl_{\varphi}(X) = k$ for some $k \not\equiv n \pmod{2}$.

Now let $C'_{i_0}, C'_{i_1}, \dots$ be all the positions in π' witnessing that X is stack-increasing, i.e. X is the principal formula in C'_{i_j} for all $j \in \mathbb{N}$. Then $t'(C'_{i_j}) = e_k$ and, hence, X_k is the principal formula in the successor of $lst(C'_{i_j})$. However, the successor of $lst(C'_{i_j})$ is $fst(C'_{i_{j+1}})$. Since the rule that is

played between each C'_{ij} and $C'_{i_{j+1}}$ is (\vee) for every $j \in \mathbb{N}$ we have $\|C'_{ij}\| = \|C'_{i_{j+1}}\|$. Applying Lemma 18 yields $\|fst(C'_{ij})\| = \|fst(C'_{i_{j+1}})\|$ for every $j \in \mathbb{N}$. Since $\|C'_{ij}\| \leq \|C'_{i_{j+1}}\|$ for every $j \in \mathbb{N}$ we also get $\|fst(C'_{i_{j+1}})\| \leq \|fst(C'_{i_{(j+1)+1}})\|$. These, however, are configurations with X_k in principal position. Finally, Fact 17 shows that between these configurations, the stack content of each $fst(C'_{i_{j+1}})$ never gets popped. Hence, X_k is stack-increasing, too.

What remains to be shown is that X_k is outermost among the stack-increasing variables in π . Suppose it is not. Then there must be another $X_{k'}$ that is outermost and stack-increasing in π s.t. $k' > k$. Since $X_{k'}$ is outermost, there are only finitely many configurations with $X_{k'}$ in principal position that result from an application of rule (FP). But there are infinitely many positions with $X_{k'}$ in principal position. Hence, there are infinitely many C_{i_0}, C_{i_1}, \dots s.t. for all $j \in \mathbb{N}$: $C_{ij} = C'_{ij}, \delta_{ij} \vdash X_{k'}$ for some C'_{ij} and δ_{ij} . But then the rule that put $X_{k'}$ into principal position must have been (\diamond) or (\square) infinitely many times. Hence, there are infinitely many configurations $C'_{i_0}, C'_{i_1}, \dots$ in π' that are labeled $e_{k'}$. Note that $k' > 1$ since $k \geq 1$, hence, $a_{k'}$ is impossible. But then the principal formula in each C'_{ij} must be a variable Y_{ij} with $lvl_\varphi(Y_{ij}) = k'$. Since there are only finitely many variables in $Sub(\varphi)$, there must be a Y with $lvl_\varphi(Y) = k'$. According to Lemma 18, Y must be stack-increasing in π' which contradicts the assumption that X is the outermost stack-increasing variable in π' .

We conclude that ζ is indeed a winning strategy for player \exists in the game $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$, and according to Theorem 9 we have $\mathcal{G}(t, \varphi) \models \Phi_n$. \square

Theorem 20. *For every $n \in \mathbb{N}$, every $\varphi \in \Sigma_n^{sym}$ and every $t \in \mathcal{T}_{\mathcal{P}}$ for some \mathcal{P} we have: if $\mathcal{G}(t, \varphi) \models \Phi_n$ then $t \models \varphi$.*

Proof. Suppose $\mathcal{G}(t, \varphi) \models \Phi_n$, i.e. player \exists has a winning strategy ζ for the game $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$. As above, we will define a strategy ζ' for her in the game $\mathcal{G}(t, \varphi)$.

Suppose a $\psi \in Sub(\varphi)$ requires player \exists to make a genuine choice, i.e. ψ is a disjunction or a \diamond . Note that for every configuration $C' = w', \delta' \vdash \psi$ that is reachable in $\mathcal{G}(t, \varphi)$ there is a unique configuration $lst(C') = C', \delta \vdash \diamond$ in the corresponding play for some δ . By assumption, strategy ζ tells player \exists how to choose in $lst(C')$. This is used to define a corresponding choice in C' :

$$\zeta'(C') := C'' \quad \text{iff} \quad \zeta(lst(C')) = C'', \delta \vdash \psi \quad \text{for some } \delta, \psi$$

Now suppose for the sake of contradiction that player \forall plays with his best strategy against ζ' and the resulting play π' is winning for player \forall . According to Lemma 16, the corresponding play π of $\mathcal{G}(\mathcal{G}(t, \varphi), \Phi_n)$ conforms to ζ . Thus, π is winning for player \exists .

Suppose player \forall wins π' with his winning condition (3), i.e. by reaching a false atomic proposition. Then the path π' in $\mathcal{G}(t, \varphi)$ is eventually labeled with f only, and player \exists cannot possibly win π because f does not occur positively in Φ_n .

Suppose therefore that player \forall wins π' with his winning condition (4), i.e. the outermost stack-increasing variable is of type μ . Analogously to the proof of Theorem 19, the outermost stack-increasing variable in π would have to be of type μ , too. But this contradicts the assumption that player \exists is the winner of π .

We conclude that ζ' must be a winning strategy for player \exists in the game $\mathcal{G}(t, \varphi)$ and, thus, $t \models \varphi$ because of Theorem 9. \square

Merging Theorems 19 and 20 yields that Φ_n describes exactly those game trees that are winning for player \exists .

Corollary 21. *For every $n \in \mathbb{N}$, every $\varphi \in \Sigma_n^{\text{syn}}$ and every $t \in \mathcal{T}_{\mathcal{P}}$ for some \mathcal{P} we have: $t \models \varphi$ iff $\mathcal{G}(t, \varphi) \models \Phi_n$.*

4.2. Strict formulas

Lemma 22. *For every $n \in \mathbb{N}$ and every closed $\varphi \in \Sigma_n^{\text{syn}}$ there is a unique $t^* \in \mathcal{T}_{\mathcal{P}_n}$ s.t. $t^* = \mathcal{G}(t^*, \varphi)$.*

Proof. It is known [2] that $\mathcal{T}_{\mathcal{P}_n}$ forms a metric space with the metric $\delta(t_1, t_2) := \inf \{ 2^{-k} \mid \text{for all } m \leq k \text{ and all } w \in \{0, 1\}^m: t_1(w) = t_2(w) \}$. Furthermore, this metric space is complete since every sequence of trees t_0, t_1, \dots s.t. t_i and t_{i+1} have a bigger common tree prefix than t_{i-1} and t_i for all $i \geq 1$ has a limit.

A closed $\varphi \in \Sigma_n^{\text{syn}}$ induces a mapping $\lambda t. \mathcal{G}(t, \varphi)$ of type $\mathcal{T}_{\mathcal{P}_n} \rightarrow \mathcal{T}_{\mathcal{P}_n}$. This mapping is contracting, i.e. there is a $c \in \mathbb{R}$, s.t. $0 \leq c < 1$ and for all $t_1, t_2 \in \mathcal{T}_{\mathcal{P}_n}$: $\delta(\mathcal{G}(t_1, \varphi), \mathcal{G}(t_2, \varphi)) \leq c \cdot \delta(t_1, t_2)$. Suppose $\delta(t_1, t_2) = 2^{-k}$, i.e. t_1 and t_2 do not differ on the first k levels. Note that if φ is atomic then $\delta(\mathcal{G}(t_1, \varphi), \mathcal{G}(t_2, \varphi)) = 0$. If it is not atomic then the rules ensure that the same is done on the formula part in both $\mathcal{G}(t_1, \varphi)$ and $\mathcal{G}(t_2, \varphi)$ before the next level of t_1 , respectively, t_2 is seen. Hence, $c = 0.5$ makes the contraction inequality true.

Finally, according to Banach's Fixpoint Theorem, every contracting mapping on a complete metric space has a unique fixpoint which proves the claim. \square

Theorem 23. *For every $n \geq 1$ we have $\Phi_n \in \Sigma_n^w \setminus \Pi_n^w$.*

Proof. By Fact 13 we have $\Phi_n \in \Sigma_n^{\text{syn}}$. Lemma 3 immediately gives us $\Phi_n \in \Sigma_n^w$. Now suppose that also $\Phi_n \in \Pi_n^w$. By definition, there is a $\varphi \in \Pi_n^{\text{syn}}$ s.t. $\Phi_n \approx \varphi$. By Lemma 11 there is a $\bar{\varphi} \in \Sigma_n^{\text{syn}}$ s.t. for any \mathcal{P} and all $t \in \mathcal{T}_{\mathcal{P}}$ we have $t \models \varphi$ iff $t \not\models \bar{\varphi}$. According to Lemma 22, there is a $t^* \in \mathcal{T}_{\mathcal{P}_n}$ s.t. $t^* = \mathcal{G}(t^*, \bar{\varphi})$. Now,

$$t^* \models \Phi_n \quad \text{iff} \quad t^* \not\models \bar{\varphi} \quad \text{iff} \quad \mathcal{G}(t^*, \bar{\varphi}) \not\models \Phi_n \quad \text{iff} \quad t^* \not\models \Phi_n$$

by Theorems 19 and 20. Hence, the assumption $\Phi_n \in \Pi_n^w$ cannot be valid. \square

Corollary 24. *For all $n \in \mathbb{N}$: $\Sigma_n^w \subsetneq \Sigma_{n+1}^w$.*

Corollary 25. *For all $n \in \mathbb{N}$: $\Sigma_n^s \subsetneq \Sigma_{n+1}^s$.*

Proof. For all $n \geq 1$ we have $\Phi_n \in \Sigma_n^{\text{syn}}$ by Fact 13. From Lemma 3 then follows $\Phi_n \in \Sigma_n^s$. The goal is to show that $\Phi_n \notin \Pi_n^s$. Suppose $\Phi_n \in \Pi_n^s$, i.e. there is a $\varphi \in \Pi_n^{\text{syn}}$ s.t. $\Phi_n \equiv \varphi$. By Lemma 1 we have $\Phi_n \approx \varphi$, hence $\Phi_n \in \Pi_n^w$ which contradicts Theorem 23. \square

5. Conclusion and open questions

Clearly, the strictness of the alternation hierarchy over a class of structures \mathfrak{K} implies its strictness over any superclass $\mathfrak{K}' \supseteq \mathfrak{K}$. Suppose there is a $\varphi \in \Sigma_n^{\text{syn}}$ for some n s.t. there is no $\psi \in \Pi_n^{\text{syn}}$ with $K \models \varphi$ iff $K \models \psi$ for all $K \in \mathfrak{K}$. Then this is certainly still the case for all $K \in \mathfrak{K}'$.

Thus, the alternation hierarchy in FLC is strict over the class of all node-labeled Kripke structures. For the class of trees with arbitrary but finite and fixed degree this also follows from the fact that they can be encoded using binary trees. It then follows for graphs of such out-degree because FLC formulas cannot distinguish bisimilar structures [17,15].

Finally, strictness of the alternation hierarchy holds for arbitrary node- and edge-labeled structures K because they can be encoded by node-labeled structures K' only. K' is obtained from K by replacing every transition $s \xrightarrow{a} t$ with a pair of transitions $s \rightarrow (s, a, t)$ and $(s, a, t) \rightarrow t$ where (s, a, t) is a new state in which the proposition a is true. Now take any formula φ of multi-modal FLC – i.e. with modalities of the form $\langle a \rangle$ and $[a]$ for some a . Let φ' result from φ by replacing every $\langle a \rangle$ with $\diamond; (a \wedge \diamond)$ and every $[a]$ with $\square; (\bar{a} \vee \square)$. Then we have $K \models \varphi$ iff $K' \models \varphi'$. Furthermore, for all $n \in \mathbb{N}$: $\varphi \in \Sigma_n^{\text{syn}}$ iff $\varphi' \in \Sigma_n^{\text{syn}}$. This transfers the alternation hierarchy to node- and edge-labeled structures.

Three natural questions, however, arise. The first one is: is the hierarchy strict over the class of finite models? For the modal μ -calculus, this is a consequence of the finite model property. If the hierarchy was strict over arbitrary structures but collapsed over finite structures then there would be a formula that is satisfiable but has no finite model. Clearly, this is excluded by the finite model property. FLC however does not have the finite model property. Hence, such a scenario could exist.

A second approach to show the strictness over finite models is to restrict the class of models even further. Note that Theorem 23 shows that the hierarchy is already strict over the class $\{t^* \in \mathcal{T}_{\mathcal{P}_n} \mid n \in \mathbb{N}, t^* = \mathcal{G}(t^*, \varphi) \text{ for some } \varphi \in \Sigma_n^{\text{syn}}\}$ of trees that are fixpoints according to Lemma 22. Unfortunately, these trees do not necessarily have finite representations because each such t^* basically is the infinite game tree on its corresponding φ disregarding the trees' nodes. If there were finite representations of these t^* then the strictness of the hierarchy over finite models would follow from bisimulation invariance.

The second question concerns a class of structures that is important in computer science but is not a superclass of the class of infinite, binary trees: infinite words. Does the alternation hierarchy in FLC interpreted over infinite words only collapse? This question has been answered for the modal μ -calculus to the affirmative [22,10,12]. It is known that FLC formulas on ω -words are equi-expressive to alternating context-free ω -grammars with a parity condition [13]. A possible approach to show the collapse of the alternation hierarchy is to translate these grammars into ones with a weak parity acceptance condition. However, Kupferman and Vardi's technique used for finite automata does not seem amendable because the “run” of an FLC formula on an ω -word can be a DAG of unbounded width. Hence, it is not possible to assign finite ranks to its nodes anymore.

The third question is: does the hierarchy of the modal μ -calculus (\mathcal{L}_μ) collapse inside FLC? Note that \mathcal{L}_μ is a fragment of FLC: a modal μ -calculus subformula $\diamond\varphi$ is read in FLC as $\diamond; \varphi$. It is thinkable that any $\varphi \in \Sigma_n^{\text{syn}} \cap \mathcal{L}_\mu$ for any n is equivalent to some ψ in, say, Σ_{17}^{syn} . A possible approach to this question would be to extend Bradfield's technique of expressing the semantics of a μ -calculus formula in μ -arithmetic to the whole of FLC. This could link the three alternation hierarchies in μ -arithmetic, the modal μ -calculus and FLC and show that the strict modal μ -calculus formulas are strict in FLC, too.

Acknowledgments

We thank the two anonymous referees for various suggestions that have led to considerable improvements in this work regarding complementation closure of FLC for example. The most valuable one, however, was the idea to extend Arnold's proof of the strict hierarchy in the modal μ -calculus.

References

- [1] A. Arnold, The modal μ -calculus alternation hierarchy is strict on binary trees, *RAIRO—Theoretical Informatics and Applications* 33 (1999) 329–339.
- [2] A. Arnold, M. Nivat, The metric space of infinite trees. Algebraic and topological properties, *Fundamenta Informaticae* 3 (4) (1980) 181–205.
- [3] J.C. Bradfield, The modal μ -calculus alternation hierarchy is strict, in: U. Montanari, V. Sassone (Eds.), *Proceedings of 7th Conference on Concurrency Theory, CONCUR'96*, Lecture Notes in Computer Science, vol. 1119, Springer, Pisa, Italy, 1996, pp. 233–246.
- [4] J.C. Bradfield, Simplifying the modal μ -calculus alternation hierarchy, in: M. Morvan, C. Meinel, D. Krob (Eds.), *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science, STACS'98*, Lecture Notes in Computer Science, vol. 1373, Springer, Paris, France, 1998, pp. 39–49.
- [5] J.C. Bradfield, Fixpoint alternation: arithmetic, transition systems, and the binary tree, *RAIRO—Theoretical Informatics and Applications* 33 (4/5) (1999) 341–356.
- [6] A. Browne, E.M. Clarke, S. Jha, D.E. Long, W. Marrero, An improved algorithm for the evaluation of fixpoint expressions, *Theoretical Computer Science* 178 (1–2) (1997) 237–255.
- [7] R. Cleaveland, Tableau-based model checking in the propositional μ -calculus, *Acta Informatica* 27 (8) (1990) 725–748.
- [8] E.A. Emerson, C.S. Jutla. Tree automata, μ -calculus and determinacy, in: *Proceedings of the 32nd Symposium on Foundations of Computer Science, IEEE, San Juan, Puerto Rico*, pp. 368–377, 1991..
- [9] M. Jurdziński, Small progress measures for solving parity games, in: H. Reichel, S. Tison (Eds.), *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, STACS'00*, Lecture Notes in Computer Science, vol. 1770, Springer, Berlin, 2000, pp. 290–301.
- [10] R. Kaivola, Using Automata to Characterise Fixed Point Temporal Logics. PhD thesis, LFCS, Division of Informatics, The University of Edinburgh, 1997. Tech. Rep. ECS-LFCS-97-356..
- [11] D. Kozen, Results on the propositional μ -calculus, *Theoretical Computer Science* 27 (1983) 333–354.
- [12] O. Kupferman, M.Y. Vardi, Weak alternating automata and tree automata emptiness, in: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing, STOC'98*, ACM Press, Washington, DC, 1998, pp. 224–233.
- [13] M. Lange, Alternating context-free languages and linear time μ -calculus with sequential composition, in: P. Panangaden, U. Nestmann (Eds.), *Proceedings of the 9th Workshop on Expressiveness in Concurrency, EXPRESS'02*, ENTCS, vol. 68.2, Elsevier, Brno, Czech Republic, 2002, pp. 71–87.
- [14] M. Lange, Local model checking games for fixed point logic with chop, in: L. Brim, P. Jancar, M. Kretínský, A. Kucera (Eds.), *Proceedings of the 13th Conference on Concurrency Theory, CONCUR'02*, Lecture Notes in Computer Science, vol. 2421, Springer, Brno, Czech Republic, 2002, pp. 240–254.
- [15] M. Lange, C. Stirling, Model checking fixed point logic with chop, in: M. Nielsen, U.H. Engberg (Eds.), *Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures, FOSSACS'02*, Lecture Notes in Computer Science, vol. 2303, Springer, Grenoble, France, 2002, pp. 250–263.
- [16] G. Lenzi, A hierarchy theorem for the μ -calculus, in: F. Meyer auf der Heide, B. Monien (Eds.), *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming, ICALP'96*, Lecture Notes in Computer Science, vol. 1099, Springer, Paderborn, Germany, 1996, pp. 87–97.
- [17] M. Müller-Olm, A modal fixpoint logic with chop, in: C. Meinel, S. Tison (Eds.), *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science, STACS'99*, Lecture Notes in Computer Science, vol. 1563, Springer, Trier, Germany, 1999, pp. 510–520.

- [18] D. Niwiński, Fixed point characterization of infinite behavior of finite-state systems, *Theoretical Computer Science* 189 (1–2) (1997) 1–69.
- [19] H. Seidl, Fast and simple nested fixpoints, *Information Processing Letters* 59 (6) (1996) 303–308.
- [20] P. Stevens, C. Stirling, Practical model-checking using games, in: B. Steffen (Ed.), *Proceedings of the 4th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'98*, Lecture Notes in Computer Science, vol. 1384, Springer, Berlin, 1998, pp. 85–101.
- [21] A. Tarski, A lattice-theoretical fixpoint theorem and its application, *Pacific Journal of Mathematics* 5 (1955) 285–309.
- [22] M.Y. Vardi, P. Wolper, Reasoning about infinite computations, *Information and Computation* 115 (1) (1994) 1–37.
- [23] I. Walukiewicz, Monadic second order logic on tree-like structures, in: *Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science, STACS'96*, Lecture Notes in Computer Science, vol. 1046, Springer, Grenoble, France, 1996, pp. 401–413.