On the variable hierarchy of the modal μ-calculus

Dietmar Berwanger[1], Erich Grädel[1], and Giacomo Lenzi[2]

[1] Mathematische Grundlagen der Informatik, RWTH Aachen, D-52056 Aachen,
{berwanger, graedel}@informatik.rwth-aachen.de
[2] Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, I-56127 Pisa,
lenzi@mail.dm.unipi.it

Abstract. We investigate the structure of the modal μ-calculus L_μ with respect to the question of how many different fixed point variables are necessary to define a given property. Most of the logics commonly used in verification, such as CTL, LTL, CTL*, PDL, etc. can in fact be embedded into the two-variable fragment of the μ-calculus. It is also known that the two-variable fragment can express properties that occur at arbitrarily high levels of the alternation hierarchy. However, it is an open problem whether the variable hierarchy is strict.

Here we study this problem with a game-based approach and establish the strictness of the hierarchy for the case of existential (i.e., $\Box$-free) formulae. It is known that these characterize precisely the L_μ-definable properties that are closed under extensions. We also relate the strictness of the variable hierarchy to the question whether the finite variable fragments satisfy the existential preservation theorem.

Keywords: modal μ-calculus, games, descriptive complexity

1 Introduction

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

One of the most important open problems concerning the μ-calculus is the complexity of the model checking problem: Is there a polynomial-time algorithm that, given a formula $\psi \in L_\mu$ and a finite Kripke structure $\mathcal{K}$, computes the set of nodes v such that $\mathcal{K}, v \models \psi$. Like most evaluation problems for logical systems, the model checking problem for L_μ can be reformulated as the strategy problem for appropriate evaluation games. The games associated with fixed point logics are *parity games*, which are infinite games where each position is assigned a natural number, called its *priority*, and the winner of an infinite play is determined according to whether the least priority seen infinitely often during the play is even or odd. It is open whether winning sets and winning strategies for parity games can be computed in polynomial time. The best algorithms known today are polynomial in the size of the game, but exponential with respect to the number of priorities. Competitive model checking algorithms for the modal μ-calculus work by solving the strategy problem for the associated parity game (see, e.g., [11]).

The number of priorities, the main source of difficulty for solving parity games, is tightly related to the *alternation depth* of L_μ-formulae, i.e., the number of (genuine) alternations between least and greatest fixed points. The correspondence goes both ways: The model checking problem for a formula ψ with alternation depth d on a finite transition system $\mathcal{K}$ translates to the strategy problem of a parity game of size $O(|\psi| \cdot \|\mathcal{K}\|)$ and with at most $d+1$ priorities. Conversely, for any $d \in \mathbb{N}$, there exists an L_μ-formula W^d of alternation depth d that defines the winning positions of Player 0 in any parity game with d priorities.

It has been shown by Bradfield [6] that the alternation hierarchy of the μ-calculus is strict. Variants of this result have also been proven by Lenzi [13] and Arnold [2]. In fact the parity game formulae W^d witness the strictness of the alternation hierarchy.

Theorem 1 (Bradfield). *For any number $d > 0$, the formula W^d is not equivalent to any L_μ-formulae of alternation depth $d-1$.*

Fortunately, most specification properties used in practical applications require very small alternation depths. Indeed, many popular sublogics of L_μ, such as LTL, CTL, PDL, and CTL* can be embedded into the first or second alternation level of L_μ. However, an interesting counterexample to this pattern is Parikh's game logic GL [15]. To exploit its full reasoning power, GL should be interpreted over neighbourhood structures, which are more general than Kripke structures, with accessibility relations between *sets* of states. But GL is also a very interesting logic on Kripke structures. It looks very similar to PDL, but is used to reason about games rather than programs. For instance, a GL-formula of form $\langle g \rangle \psi$ expresses that Player 0 has a strategy in the game g to achieve an outcome where ψ holds. Games are constructed from atomic games by similar operations as the program operations of PDL, namely union, composition, and iteration, and in addition a dualization operation, mapping g to the dual game g^d, in which the roles of the two players are switched. The intertwining of role switches and iteration leads to unexpected expressive power, high complexity, and unbounded alternation levels. Indeed it has been shown by Berwanger [4] that GL (on Kripke structures) intersects non-trivially with all levels of the alternation hierarchy. In fact GL is strong enough to reason about parity games, in the sense that GL contains, for every d, a formula that is equivalent to W^d. It is currently still open whether GL (on Kripke structures) has the same expressive power as the μ-calculus.

While the strictness of the alternation hierarchy does not help us to separate GL and L_μ, recent investigations have brought to light another interesting hierarchy inside the μ-calculus, namely the *variable hierarchy.* By re-using fixed point variables several times it is possible to write many L_μ-formulae, even with very high nesting and alternation depth, using only very few variables. For any k, we denote by $L_\mu[k]$ the fragment of L_μ consisting of those formulae that make use of at most k distinct fixed-point variables. It turns out that most of the common sublogics of L_μ that are used in verification can be embedded into $L_\mu[2]$, the two-variable fragment of the μ-calculus. In particular this is the case for GL (on Kripke structures) and hence for all logics subsumed by GL, including CTL, LTL, CTL*, PDL, and Δ-PDL (see [16]).

The problem we are investigating in this paper is whether the variable hierarchy is strict. We conjecture that for every $k \geq 1$ there exist formulae in $L_\mu[k]$ that are not equivalent to any formula in $L_\mu[k-1]$. This can easily be shown for $k = 1$ and $k = 2$ (see Section 2) but it is currently unknown whether $L_\mu[2] = L_\mu$. Clearly, separating $L_\mu[2]$ from L_μ would also separate GL from L_μ,

The question whether the variable hierarchy is strict is meaningful and interesting not only for the μ-calculus itself, but also for relevant fragments. More precisely, given any set of formulae $L \subseteq L_\mu$ the variable hierarchy problem for L is the question whether there exist for every $k \geq 1$ formulae in $L[k] := L \cap L_\mu[k]$ that are not equivalent to any formula in $L[k-1]$.

In this paper we answer the question affirmatively for the $\Box$*-free fragment* of the μ-calculus. A formula $\psi \in L_\mu$ is called $\Box$-free or existential if it can be built from atoms and negated atoms by means of $\wedge, \vee$, existential modalities $\langle a \rangle$ and least and greatest fixed point. It is not difficult to see that every $\Box$-free formula is closed under extensions. This means that whenever $\mathcal{K}, v \models \psi$ and $\mathcal{K} \subseteq \mathcal{K}'$, then also $\mathcal{K}', v \models \psi$. The $\Box$-free fragment is important because of the preservation theorem that has recently been established by D'Agostino and Hollenberg [7] which shows that also the converse holds.

Theorem 2. *A formula $\psi \in L_\mu$ is closed under extensions if and only if it is equivalent to a $\Box$-free formula in L_μ.*

We describe here a sequence of rather simple $\Box$-free formulae $\psi^{(k)} \in L_\mu[k]$ and prove that, for every k, $\psi^{(k)}$ is not equivalent to any $\Box$-free formula with less than k variables. Intuitively, the formulae $\psi^{(k)}$ express that the models contain a substructure that is bisimilar to a kind of k-clique with labeled edges (see Section 4 for precise definitions). These formulae use only conjunctions, existential modalities, and greatest fixed points. In particular they are alternation free.

We conjecture that the formulae $\psi^{(k)}$ witness in fact the strictness of the variable hierarchy for the full μ-calculus, not just for its existential fragment. This conjecture is related to the question whether the existential preservation theorem holds for the bounded variable fragments $L_\mu[k]$ of the μ-calculus.

Our analysis suggests that the variable hierarchy is 'orthogonal' to the alternation hierarchy. We have proved [4] that already in $L_\mu[2]$ one can define properties on arbitrary levels of the alternation hierarchy. Further, while the alternation hierarchy of L_μ is of importance for the complexity, at least for the currently known model checking algorithms (which are exponential in the number of alternations), this does not seem to be the case for the variable hierarchy.

Theorem 3 (Berwanger). *The model checking problem for $L_\mu[2]$ can be solved in polynomial time iff this is the case for the full μ-calculus.*

Finally, the clique formulae by which we proved the strictness of the hierarchy for existential formulae are pure ν-formulae, and thus on the lowest level of the alternation hierarchy.

Here is an overview over the contents of this paper. In Section 2 we define the μ-calculus, explain parity games and introduce the variable hierarchy. In Section 3 we discuss the $\Box$-free fragment of L_μ and prove a technical lemma on strategy trees for this fragment. In Section 4 we define the formulae that will witness the strictness of the variable hierarchy for the $\Box$-free fragment of L_μ. Finally, in Section 5 we prove our hierarchy theorem.

2 The μ-calculus

Fix a set ACT of actions and a set PROP of atomic propositions. A transition system or Kripke structure for ACT and PROP is a structure $\mathcal{K}$ with universe V (whose elements are called states), binary relations $E_a \subseteq V \times V$ for each $a \in$ ACT, and monadic relations $p \subseteq V$ for each atomic proposition $p \in$ PROP (we do not distinguish notationally between atomic propositions and their interpretations).

Syntax of L_μ. For a set ACT of actions, a set PROP of atomic propositions, and a set VAR of variables, the formulae of L_μ are defined by the grammar

$$\varphi ::= \mathit{false} \mid \mathit{true} \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $p \in$ PROP, $a \in$ ACT, and $X \in$ VAR.

Semantics of L_μ. Formulae of L_μ are evaluated on transition systems at a particular state. Given a sentence ψ and a transition system $\mathcal{K}$ with state v, we write $\mathcal{K}, v \models \psi$ to denote that ψ holds in $\mathcal{K}$ at state v. The set of states $v \in V$ such that $\mathcal{K}, v \models \psi$ is denoted by $[\![\psi]\!]^{\mathcal{K}}$. We omit the definition of $[\![\psi]\!]^{\mathcal{K}}$ for the obvious cases. For the modal operators,

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

To understand the semantics of fixed point formulae, note that a formula $\psi(X)$ with a propositional variable X defines on every transition system $\mathcal{K}$ (with state set V, and with interpretations for free variables other than X occurring in ψ) an operator $\psi^{\mathcal{K}} : \mathcal{P}(V) \to \mathcal{P}(V)$ assigning to every set $X \subseteq V$ the set $\psi^{\mathcal{K}}(X) := [\![\psi]\!]^{\mathcal{K},X} = \{v \in V : (\mathcal{K}, X), v \models \psi\}$. As X occurs only positively in ψ, the operator $\psi^{\mathcal{K}}$ is *monotone* for every $\mathcal{K}$, and therefore, by a well-known theorem due to Knaster and Tarski, has a least fixed point $\mathrm{lfp}(\psi^{\mathcal{K}})$ and a greatest fixed point $\mathrm{gfp}(\psi^{\mathcal{K}})$. Now we put

$$[\![\mu X.\psi]\!]^{\mathcal{K}} := \mathrm{lfp}(\psi^{\mathcal{K}}) \text{ and } [\![\nu X.\psi]\!]^{\mathcal{K}} := \mathrm{gfp}(\psi^{\mathcal{K}}).$$

Model checking games. The semantics of L_μ can also be described in terms of *parity games.* Such a game is given by a transition system $\mathcal{G} = (V, V_0, E, \Omega)$, where V is a set of positions with a designated subset V_0, $E \subseteq V \times V$ is a transition relation, and $\Omega : V \to \mathbb{N}$ assigns to every position a *priority.* A *play* of $\mathcal{G}$ is a path $v_0, v_1, \ldots$ formed by the two players starting from a given position v_0. If the current position v belongs to V_0, Player 0 chooses a move $(v, w) \in E$ and the play proceeds from w. Otherwise, her opponent, Player 1, chooses the move. When no moves are available at the current position, the player who has to choose loses. In case this never occurs the play goes on infinitely and the winner is established by looking at the sequence $\Omega(v_0), \Omega(v_1), \ldots$ If the least priority appearing infinitely often in this sequence is even, Player 0 wins the play, otherwise Player 1 wins.

Let $V_1 := V \setminus V_0$ be the set of positions where Player 1 moves. A *strategy* for Player i in $\mathcal{G}$ is a partial function $f : V^*V_i \to V$ which indicates for an initial play $v_0, v_1, \ldots, v_r$ up to some position $v_r \in V_i$ a possible prolongation w, so that $(v_r, w) \in E$. If Player i wins every play where he moves according to f, we say that f is a *winning* strategy. A strategy that does not depend on the history of the play but only on the current position is called a *positional* strategy. The Forgetful Determinacy Theorem for parity games [8] states that these games are always determined (i.e., from each position one of the players has a winning strategy) and, in fact, positional strategies always suffice.

Theorem 4 (Forgetful Determinacy). *In any parity game the set of positions can be partitioned into two sets W_0 and W_1 such that Player 0 has a positional winning strategy on W_0 and Player 1 has a positional winning strategy on W_1.*

Given a transition system $\mathcal{K}, v_0$ and a L_μ-sentence ψ, the model checking game $\mathcal{G}(\mathcal{K}, \psi)$ is a parity game associated with the problem whether $\mathcal{K}, v_0 \models \psi$. There are several, essentially equivalent, ways to define this game. In the more transparent one, positions are pairs (v, φ) where φ is any (not necessarily closed) subformula of ψ, and it is assumed that every variable is bound at most once by a fixed-point definition (see, e.g., [5, 17]). For certain technical reasons, and since we want to re-use variables several times, we use here the slightly less intuitive variant (more familiar from automata theory [8, 12]), which, instead of subformulae, uses their closure, that is, the sentence obtained by replacing, recursively, every free occurence of a variable by its binding definition.

Definition 5. *The* closure $\mathrm{cl}(\psi)$ *of a sentence* $\psi \in L_\mu$ *is the smallest set of sentences so that* $\psi \in \mathrm{cl}(\psi)$ *and*

(1) if $\varphi_1 \vee \varphi_2 \in \mathrm{cl}(\psi)$ *or* $\varphi_1 \wedge \varphi_2 \in \mathrm{cl}(\psi)$ *then* $\{\varphi_1, \varphi_2\} \subseteq \mathrm{cl}(\psi)$*;*
(2) if $\langle a \rangle \varphi \in \mathrm{cl}(\psi)$ *or* $[a]\varphi \in \mathrm{cl}(\psi)$ *then* $\varphi \in \mathrm{cl}(\psi)$*;*
(3) for λ *denoting either* μ *or* ν*, if* $\lambda X.\varphi(X) \in \mathrm{cl}(\psi)$ *then* $\varphi(\lambda X.\varphi(X)) \in \mathrm{cl}(\psi)$.

Then the positions in the game $\mathcal{G}(\mathcal{K}, \psi)$ are pairs (v, φ) of states $v \in V$ and sentences $\varphi \in \mathrm{cl}(\psi)$. Player 0 moves from the positions $(v, \varphi_1 \vee \varphi_2)$, $(v, \langle a \rangle \varphi)$,

(v,p) with $v \notin p$, and $(v, \neg p)$ with $v \in p$. All plays start at (v_0, ψ) and the transitions in E are such that

- no moves are possible from (v, α) where α is atomic or negated atomic;
- from $(v, \varphi_1 \vee \varphi_2)$ or $(v, \varphi_1 \wedge \varphi_2)$ transitions lead to (v, φ_1) and (v, φ_2);
- from $(v, \langle a \rangle \varphi)$ or $(v, [a]\varphi)$ there are transitions to all positions (w, φ) where w is an a-successor of v.
- from $(v, \lambda X.\varphi(X))$ there is a transition to $(v, \varphi(\lambda X.\varphi(X))$;

Thus, a play proceeds along the paths in $\mathcal{K}$ and in the syntax tree of ψ, until it hits a fixed point variable (which is a leaf in the syntax tree). There the play resumes with the binding definition of the variable. We call this event the *regeneration* of a variable.

By repeatedly regenerating variables, it may happen that neither Player 0 nor Player 1, here called Verifier and Falsifier, ever gets stuck. To decide the winner of such plays, priorities have to be defined appropriately. The intuition is that, to establish the truth of a μ-formula, Verifier should regenerate it only finitely often whereas ν-formulae can be regenerated infinitely often. Of course the difficulty may be that μ- and ν-formulae are deeply nested and there are several fixed-point formulae that are regenerated infinitely often during a play. But it can be shown that among these, there is always an outermost one, which determines the winner: if it is a ν-formula Verifier wins, if it is a μ-formula, Falsifier wins. Hence, the priority labelling assigns even priorities to positions $(v, \nu X.\varphi)$ and odd priorities to positions $(v, \mu X.\varphi)$. Further, priorities respect dependencies. If $\nu Y.\varphi$ depends on $\mu X.\eta$ then priorities of positions $(v, \nu Y.\varphi)$ are higher than those of positions $(w, \mu X.\eta)$. The remaining positions receive priorities that are higher than those associated with fixed-point formulae. For details (which are not needed in this paper) see, e.g., [5, 17].

Theorem 6. *Verifier has a winning strategy in the model checking game $\mathcal{G}(\mathcal{K}, \psi)$ from position (u, ψ) iff $\mathcal{K}, u \models \psi$.*

As a modal logic, the μ-calculus distinguishes between transitions systems only up to behavioral equivalence, captured by the notion of bisimulation.

Definition 7. *A* bisimulation *between two transition systems $\mathcal{K}$ and $\mathcal{K}'$ is a relation $Z \subseteq V \times V'$ between the domains of $\mathcal{K}$ and $\mathcal{K}'$, respecting the atomic propositions $p \in$* PROP *in the sense that $\mathcal{K}, v \models p$ iff $\mathcal{K}', v' \models p$, for $(v, v') \in Z$, and satisfying the following back and forth conditions.*

Forth: *for all $(v, v') \in Z$, $a \in$* ACT *and every w such that $(v, w) \in E_a$, there exists a w' such that $(v', w') \in E'_a$ and $(w, w') \in Z$.*

Back: *for all $(v, v') \in Z$, $a \in$* ACT *and every w' such that $(v', w') \in E'_a$, there exists a w such that $(v, w) \in E_a$ and $(w, w') \in Z$.*

Two transition systems $\mathcal{K}, u$ and $\mathcal{K}', u'$ are *bisimilar*, if there is a bisimulation Z between them with $(u, u') \in Z$.

An important model theoretic feature of modal logics is the *tree model property*, the fact that every satisfiable formula is satisfiable in a tree. This is a straightforward consequence of bisimulation invariance, since $\mathcal{K}, u$ is bisimilar to its tree unravelling.

Definition 8. *The* unravelling $\mathcal{T}(\mathcal{K}, u)$ *of a Kripke structure* $\mathcal{K}$ *from node* u *is the tree of all paths through* $\mathcal{K}$ *that start at* u*. More formally,*

- *the domain of* $\mathcal{T}(\mathcal{K}, u)$ *is the set* $V^{\mathcal{T}}$ *of all sequences* $\overline{v} = v_0 a_1 v_1 a_2 \cdots v_{r-1} a_r v_r$ *where* $v_i \in V$, $a_i \in \text{ACT}$, *such that* $v_0 = u$ *and* $(v_{i-1}, v_i) \in E_{a_i}$*;*
- *an atomic proposition* p *is true at* $v_0 a_1 v_1 a_2 \ldots v_{r-1} a_r v_r$ *in* $\mathcal{T}(\mathcal{K}, u)$ *iff it is true at* v_r *in* $\mathcal{K}$*;*
- *for all actions* a, $E_a^{\mathcal{T}}$ *contains the pairs* $(\overline{v}, \overline{v}av)$ *in* $V^{\mathcal{T}} \times V^{\mathcal{T}}$*.*

Obviously, the natural projection $\pi \colon \mathcal{T}(\mathcal{K}, u) \to \mathcal{K}, u$ which sends every sequence $\overline{v} = v_0 a_1 v_1 a_2 \ldots v_{r-1} a_r v_r \in V^{\mathcal{T}}$ to its last node v_r defines a bisimulation between $\mathcal{T}(\mathcal{K}, u)$ and $\mathcal{K}, u$.

The variable hierarchy.

Definition 9. *For any* $k \in \mathbb{N}$*, the* k-variable fragment $L_\mu[k]$ *of the* μ*-calculus is the set of formulae* $\psi \in L_\mu$ *that contain at most* k *distinct variables.*

The first three levels of this hierarchy are very easy to separate.

Proposition 10. $L_\mu[0] \subsetneq L_\mu[1] \subsetneq L_\mu[2]$.

With one fixed point variable, only alternation free formulae can be written. Though this suffices to state non-local properties beyond the expressive power of plain modal logic, e.g., $\nu X.\langle a \rangle X$ (the model has an infinite a-path), $L_\mu[1]$ remains below $L_\mu[2]$ which contains formulae with genuine fixed point alternation. For example, $\mu X.\nu Y.\langle a \rangle X \vee \langle b \rangle Y$ (there is an $\{a, b\}$-path with infinitely many b's) is strictly on the second level of the L_μ alternation hierarchy. Moreover, hard sentences of any level of the alternation hierarchy can be expressed with two variables, as it was shown in [4].

Simultaneous Fixed Points. There is a variant of L_μ that admits simultaneous fixed points of several formulae. This does not increase the expressive power but often allows for more modular and easier to read formalizations. The mechanism for building simultaneous fixed point formulae is the following: Given formulae $\varphi_1, \ldots, \varphi_k$ and variables $X_1, \ldots, X_k$

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

is called a *system of rules*, which can be used to build the formulae $(\mu X_i : S)$ and $(\nu X_i : S)$.

Semantics: On every Kripke structure $\mathcal{K}$, the system S defines an operator $S^{\mathcal{K}}$ mapping a k-tuple $\overline{X} = (X_1, \dots, X_k)$ of sets of states to $S_1^{\mathcal{K}}(\overline{X}), \dots, S_k^{\mathcal{K}}(\overline{X})$ with

$$S_i^{\mathcal{K}}(\overline{X}) := [\![\varphi_i]\!]^{(\mathcal{K},\overline{X})}.$$

As $S^{\mathcal{K}}$ is monotone, there exist the least and greatest fixed points $\mathrm{lfp}(S) = (X_1^\mu, \dots, X_k^\mu)$ and $\mathrm{gfp}(S) = (X_1^\nu \dots, X_k^\nu)$. Now set

$$[\![(\mu X_i : S)]\!]^{\mathcal{K}} := X_i^\mu \text{ and } [\![(\nu X_i : S)]\!]^{\mathcal{K}} := X_i^\nu.$$

Examples of simultaneous fixed point formulae will be given in Section 4. It is known that simultaneous least fixed points can be eliminated in favor of nested individual fixed points (see, e.g., [3, page 27]). Indeed,

$$\big(\mu X : X \leftarrow \psi(X,Y),\ Y \leftarrow \varphi(X,Y)\big) \equiv \mu X.\psi(X, \mu Y.\varphi(X,Y)),$$

and this equivalence generalizes to larger systems in the obvious way. Note that the translation does neither increase the number of variables nor the alternation depth of the formula.

Proposition 11. *Every formula in L_μ with simultaneous fixed points can be translated into an equivalent formula with the same number of fixed point variables and the same alternation depth.*

3 The $\Box$-free fragment

We write $\mathcal{K} \subseteq \mathcal{K}'$ to denote that $\mathcal{K}'$ is an extension of $\mathcal{K}$ (or equivalently, that $\mathcal{K}$ is a substructure of $\mathcal{K}'$). A formula ψ is *closed under extensions* if, whenever $\mathcal{K} \models \psi$ and $\mathcal{K} \subseteq \mathcal{K}'$, then also $\mathcal{K}' \models \psi$. In most logics there is a natural notion of *existential formulae* (i.e., formulae where all first-order quantifiers are existential) and it is obvious that existential formulae are closed under extensions. In many cases, also the converse holds. For instance, it is a classical result in model theory, due to Tarski [19] and Łos [14], that every first-order sentence which is closed under extensions is equivalent to an existential sentence. Results of this kind are called *existential preservation theorems*, or also *Łos-Tarski Theorems* (often stated in the dual form, in terms of universal formulae and closure under substructures).

It should be pointed out that there are many scenarios where the Łos-Tarski Theorem fails. In particular this is the case for the k-variable fragments of first-order logic, for every $k \geq 2$ [1, 9], and for first-order logic on finite structures [18].

In the μ-calculus $\langle a \rangle$-operators correspond to existential quantifiers, and $[a]$-operators to universal ones. Therefore, the existential formulae are those in the $\Box$-free fragment $L_\mu^\Diamond$, which is defined by the grammar

$$\varphi ::= \mathit{false} \mid \mathit{true} \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid \mu X.\varphi \mid \nu X.\varphi.$$

We denote the k-variable fragment of this logic by $L_\mu^\Diamond[k]$. It has recently been shown that the existential preservation theorem does hold for the μ-calculus [7].

The proof makes use of the μ-automata of Janin and Walukiewicz [10] and it is not clear whether it carries over to the k-variable fragments of L_μ. It turns out that this question is related to the strictness of the variable hierarchy and we discuss it again at the end of this paper.

One technically useful property of fixed point formulae is guardedness. In terms of games this guarantees that plays cannot get hung on a state of the transition system while the second component scans the syntax tree forever.

Definition 12. *An L_μ-sentence ψ is* guarded *if each path in the syntax tree of ψ from a fixed point definition $\lambda X.\varphi$ to an occurrence of X passes through a modality.*

In [12], Kupferman, Vardi, and Wolper give a procedure to transform every L_μ-sentence into an equivalent guarded formula. This procedure does not increase the number of variables and preserves $\Box$-free formulae.

Proposition 13. *Every formula in $L_\mu^\Diamond[k]$ is equivalent to a guarded formula in $L_\mu^\Diamond[k]$.*

We conclude this section with an observation on strategy trees for evaluation games of $\Box$-free formulae. Let $\mathcal{K}, u$ be a transition system and ψ a guarded $L_\mu^\Diamond$-formula. For every strategy f of Verifier in the game $\mathcal{G}(\mathcal{K},\psi)$, we define a tree $\mathcal{T}_f$, called the *strategy tree* for f, as follows.

The root of $\mathcal{T}_f$ is the initial position (u,ψ). Besides this, the domain of $\mathcal{T}_f$ comprises all initial plays according to f that end with a modal move:

$$\pi = (u,\psi),\dots,(v,\langle a\rangle\varphi),(w,\varphi).$$

We call the elements $\pi \in \mathcal{T}_f$ *initial segments* of the plays against f. Let $\mathrm{node}(\pi)$ denote the last node w in $\mathcal{K}$ visited by π, and $\mathrm{path}(\pi)$ the sequence of all actions occurring in π. The edges of $\mathcal{T}_f$ are labelled by actions $a \in \textsc{act}$. The a-successors of a segment π are those segments which prolong π by an a-action, i.e., those $\pi' \in \mathcal{T}_f$ with $\pi \leq \pi'$ and $\mathrm{path}(\pi') = \mathrm{path}(\pi)\cdot a$.

Observe that while Verifier moves according to its fixed strategy f in $\mathcal{G}(\mathcal{K},\psi)$, Falsifier can lead the play to any initial segment $\pi \in \mathcal{T}_f$, by means of some strategy which is not necessarily positional. Further, he can prolong the initial play π to any successor segment π' by adding to his strategy a collection of rules that do not involve any move in $\mathcal{K}$. We call this collection of rules a *local strategy.*

Hence, every edge (π,π') in $\mathcal{T}_f$ corresponds to a local strategy of Falsifier and every (maximal) path in $\mathcal{T}_f$ corresponds to a complete play in $\mathcal{G}(\mathcal{K},\psi)$ which Falsifier can enforce by composing the local strategies along its edges.

Lemma 14. *If f is a winning strategy for Verifier in $\mathcal{G}(\mathcal{K},\psi)$, then $\mathcal{T}_f \models \psi$.*

Proof. We can replicate f as a Verifier strategy for $\mathcal{G}(\mathcal{T}_f,\psi)$ in such a way that the resulting plays correspond to plays against f in the original game $\mathcal{G}(\mathcal{K},\psi)$.

To accomplish this, at every disjunction $(\pi,\varphi_1\vee\varphi_2)$ with $\mathrm{node}(\pi)=v$, Verifier applies the advice $f(v,\varphi_1\vee\varphi_2)=(v,\varphi_i)$ and moves to (π,φ_i). This ensures that for any reachable modal position $(\pi, \langle a \rangle \varphi)$ there is a play $\pi \ldots (v, \langle a \rangle \varphi)$ in $\mathcal{G}(\mathcal{K}, \psi)$, consistent with f, which Verifier then prolongs to a next initial segment π' by moving to the position $(w, \varphi) = f(v, \langle a \rangle \varphi)$. Note that in $\mathcal{T}_f$, the segment π' is an a-successor of π. Hence, in $\mathcal{G}(\mathcal{T}_f, \psi)$ Verifier can move from $(\pi, \langle a \rangle \varphi)$ to (π', φ) and proceed.

A play according to this strategy can end only at positions (π, true) in which case Verifier wins. Otherwise, if the play is infinite, the sequence of initial plays $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_n \leq \ldots$ visited on $\mathcal{T}_f$ converges to an infinite play π_∞ of $\mathcal{G}(\mathcal{K}, \psi)$. Clearly, π_∞ is consistent with f and thus winning. On the other hand, in the considered play of $\mathcal{G}(\mathcal{T}_f, \psi)$ the sequence of occurring subformulae is the same as in π_∞, so Verifier wins in this case as well.

4 The clique formulae

We now define the formulae that witness the strictness of the variable hierarchy in $L_\mu^\Diamond$. The formulae $\psi^{(k)}$ do not contain propositional atoms and have actions ij, for all $i, j = 0, \ldots, k-1$.

Definition 15. *For any $k \in \mathbb{N}$, let $\psi^{(k)} := \nu X_0 \, . \, S$ where S is the system of rules*

$$X_i \leftarrow \bigwedge_{j=0}^{k-1} \langle ij \rangle X_j$$

for $i = 0, \ldots, k-1$.

Obviously, $\psi^{(k)}$ belongs to $L_\mu^\Diamond[k]$ and we will prove that no formula in $L_\mu^\Diamond[k-1]$ is equivalent to $\psi^{(k)}$.

As a model of this formula, consider the transition system $\mathcal{C}^k$ with nodes $\{0, \ldots, k-1\}$ and with transition relations $E_{ij} = \{(i,j)\}$, that is, the k-clique with edge labels that indicate the source and the destination of every edge. Clearly, $\mathcal{C}^k, 0 \models \psi^{(k)}$.

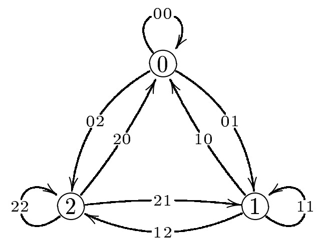


Fig. 1. The transition system $\mathcal{C}^3$

To describe the whole class of models, let $\mathcal{T}_i^k$ be the tree unravelling of $\mathcal{C}^k$ from node i. Further, let $\psi_i^{(k)}$ be the formula $(\nu X_i : S)$.

Lemma 16. *For every $i < k$, $\mathcal{T}_i^k \models \psi_i^{(k)}$. In fact, for any tree $\mathcal{T}$ we have that $\mathcal{T} \models \psi_i^{(k)}$ if and only if $\mathcal{T}$ contains a substructure (with the same root) that is isomorphic to $\mathcal{T}_i^k$.*

Hence, $\psi^{(k)}$ just expresses that its models are extensions of (the unravelling of) the clique $\mathcal{C}^k$. What makes this formula hard? To understand this, consider the following simpler variant of $\psi^{(k)}$ which only takes care of the destination of an action and which needs only one variable: $\varphi^{(k)} := \nu X. \bigvee_{i<k} \bigwedge_{j<k} \langle ij \rangle X$. Obviously $\psi^{(k)} \models \varphi^{(k)}$. The models of $\varphi^{(k)}$ are still extensions of the clique, up to renaming of edges. However, the naming of edges makes a crucial difference. Note that $\psi^{(k)}$ requires of its models to contain all paths of $\mathcal{C}^k$, and along these paths, the destination of an action is the source of the next one. We call such a pair of actions ij and jm matching. Intuitively, a formula needs memory to store the last label to be able to check whether all actions along paths are matching. In $\psi^{(k)}$ this is the role of the the k fixed point variables. Instead, $\varphi^{(k)}$ fails to distinguish matching from non matching paths, thus being unable to establish whether all matching paths are present. This leads $\varphi^{(k)}$ to accept structures that $\psi^{(k)}$ would reject.

5 The hierarchy theorem

We are now ready to prove our hierarchy theorem. Towards this goal we first analyse strategies in the game $\mathcal{G}(\mathcal{C}^k, \psi)$ for any guarded, $\Box$-free formula $\psi \equiv \psi^{(k)}$. Although Falsifier loses this game, he nevertheless has control over important aspects, notably over which node of $\mathcal{C}^k$ will be reached next.

The full branching power. Fix a winning strategy f for the game $\mathcal{G}(\mathcal{C}^k, \psi)$ and consider any initial segment

$$\pi = (0, \psi), \ldots, (i, \langle ij \rangle \varphi), (j, \varphi) \in \mathcal{T}_f$$

of a play against f. Recall that in the strategy tree $\mathcal{T}_f$, a segment π' is a (jm)-successor of π, if Falsifier has a local strategy in $\mathcal{G}(\mathcal{C}^k, \psi)$ to prolong the play from π to π' through a (jm)-action, so that $\mathrm{path}(\pi') = \mathrm{path}(\pi)(jm)$.

Definition 17. *We say that Falsifier has* full branching power *from $\pi \in \mathcal{T}_f$ if for every $m < k$, he has a strategy to lead the play from π to a successor segment π' with node$(\pi') = m$, so that Falsifier again has full branching power from π'. More formally, this means that*

$$\mathcal{T}_f, \pi \models \nu X. \bigwedge_{m<k} \bigvee_{j<k} \langle jm \rangle X,$$

or, using the notation of the previous section, $\mathcal{T}_f, \pi \models \varphi^{(k)}$.

In other words, whenever the play enters a new node i, i.e, at the completion of an initial segment, Verifier must allow Falsifier to turn the play towards a successor along any action ij he chooses. Falsifier does not need to commit himself to more than one modal move at each initial segment. We call a strategy that maintains the full branching power of Falsifier a *full branching strategy.* Observe that in $\mathcal{G}(\mathcal{C}^k, \psi)$ Falsifier may also have strategies that are not full branching. For example, when $\psi \equiv \psi^{(k)} \wedge \langle 00\rangle \mathit{true}$, Falsifier may choose the conjunct $\langle 00\rangle \mathit{true}$ which would of course scotch his branching power.

The construction of a full branching strategy in the game $\mathcal{G}(\mathcal{C}^k, \psi)$ against f can again be viewed as a game, played on $\mathcal{T}_f$. At any position π, Challenger selects a node $m < k$, and then Pathfinder moves from π to some successor π' with $\mathrm{node}(\pi') = m$. Challenger wins if Pathfinder cannot move, i.e., if he can force the play to remain finite, otherwise Pathfinder wins. With every move Pathfinder reveals a local Falsifier strategy to prolong π through the action chosen by Challenger. Composing these local strategies yields the desired full branching strategy for Falsifier. Thus, Falsifier has full branching power in the game $\mathcal{G}(\mathcal{C}^k, \psi)$ iff Pathfinder has a winning strategy in the game on $\mathcal{T}_f$.

Lemma 18. *For every guarded, $\Box$-free formula $\psi \equiv \psi^k$ and against every winning strategy f of Verifier, Falsifier has full branching power from the initial position in $\mathcal{G}(\mathcal{C}^k, \psi)$*

Proof. By Lemma 14 we know that $\mathcal{T}_f \models \psi^{(k)}$. Hence $\mathcal{T}_f \models \varphi^{(k)}$ which proves the full branching property.

Non-ambiguous formulae. We call a formula ψ *non-ambiguous* on the transition system $\mathcal{K}$, if for every subformula $\eta \in \mathrm{cl}(\psi)$ with $\eta \neq \mathit{true}$, there exists at most one node v of $\mathcal{K}$ such that $\mathcal{K}, v \models \eta$.

Lemma 19. *Any formula $\psi \in L_\mu^\Diamond$ equivalent to $\psi^{(k)}$ can be transformed, without increasing the number of variables, into an equivalent formula $\psi' \in L_\mu^\Diamond$ that is non-ambiguous on $\mathcal{C}^k$.*

Proof. We eliminate the subformulae that hold at more than one clique node. Assume that $\mathcal{C}^k, j_1 \models \eta$ and $\mathcal{C}^k, j_2 \models \eta$ for some $j_1 \neq j_2$ and some $\eta \in \mathrm{cl}(\psi)$, and let ψ' be the formula obtained by replacing every occurrence of η in ψ by *true*. We claim that $\psi' \equiv \psi$. By the tree model property of L_μ it suffices to establish this on trees.

It is obvious that ψ implies ψ'. For the converse, consider a tree model $\mathcal{T}$ of ψ'. We can partition the universe T of $\mathcal{T}$ according to the label of the incoming transition: $T = T_0 \,\dot\cup \cdots \dot\cup\, T_{k-1}$ such that the root of $\mathcal{T}$ belongs to T_0 and every node whose incoming edge is labelled ij belongs to T_j. Next, we define the extension $\mathcal{T}'$ of $\mathcal{T}$ obtained by adding at every node $v \in T_i$ the unravelling $\mathcal{T}_j^k$ of $\mathcal{C}^k$ from the node $j := j_1$ if $i \neq j_1$ and $j := j_2$ otherwise. In this way, every subtree of $\mathcal{T}'$ rooted at a node $v \in \mathcal{T}$ extends a clique unravelling $\mathcal{T}_j^k$, where η holds. Since η is $\Box$-free and hence closed under extensions, it follows that also $\mathcal{T}'_v$, the subtree of $\mathcal{T}'$ rooted at v, is a model of η. Moreover, if f_j is a winning strategy for Verifier in $\mathcal{G}(\mathcal{T}_j^k, \eta)$ it will also be a winning strategy in $\mathcal{G}(\mathcal{T}'_v, \eta)$.

By means of this, we can extend any winning strategy f of Verifier in $\mathcal{G}(\mathcal{T}, \psi')$ to a strategy in $\mathcal{G}(\mathcal{T}', \psi)$ as follows. At every position (v, φ) where $v \in \mathcal{T}$ and $\varphi \neq \eta$ choose according to f. As Falsifier cannot move in the tree, the play will stay on nodes of $\mathcal{T}$ unless a position (v, η) is reached. When this occurs, Verifier drops f and proceeds with the strategy f_j which is winning in $\mathcal{G}(\mathcal{T}_j^k, \eta)$ and thus in $\mathcal{G}(\mathcal{T}_v', \eta)$. In that way, every play of $\mathcal{G}(\mathcal{T}', \psi)$ is won by Verifier which means that $\mathcal{T}' \models \psi$ or, equivalently, $\mathcal{T}' \models \psi^{(k)}$.

Observe that in any game on $\psi^{(k)}$ modal moves, i.e., moves of the form $(v, \varphi) \to (v', \varphi')$ where $v \neq v'$, are possible only if $v \in T_i$, and v' is an ij-successor of v, for some pair ij. In particular, $\mathcal{G}(\mathcal{T}', \psi^{(k)})$ does not allow any moves between any node $v \in T_i$ of $\mathcal{T}$ and their jj'-successors $v' \in \mathcal{T}' \setminus \mathcal{T}$. Accordingly, $\mathcal{G}(\mathcal{T}', \psi^{(k)})$ is restricted to positions (v, φ) with $v \in \mathcal{T}$ and therefore just the same game as $\mathcal{G}(\mathcal{T}, \psi^{(k)})$. Since Verifier wins this game we obtain $\mathcal{T} \models \psi^{(k)}$ and can conclude that $\psi' \equiv \psi$.

We are now ready for the final step.

Theorem 20. *No $\Box$-free formula in $L_\mu[k-1]$ is equivalent to $\psi^{(k)}$.*

Proof. Towards a contradiction, suppose that $\psi \in L_\mu^\Diamond[k-1]$ is equivalent to $\psi^{(k)}$. Without loss of generality, we can assume that ψ is guarded and non-ambiguous on $\mathcal{C}^k$.

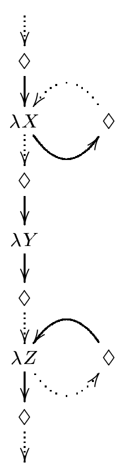
Fix a winning strategy f for Verifier in the game $\mathcal{G}(\mathcal{C}^k, \psi)$. Such a strategy must exist since $\mathcal{C}^k \models \psi$. We construct a full branching strategy g for Falsifier and prove that it forces the play $f\hat{\ }g$ (defined by the two strategies f and g) to be finite. But this is absurd since a full branching strategy necessarily leads to an infinite play.

For every initial play in $\mathcal{G}(\mathcal{C}^k, \psi)$ we define a function $S : \{X_1, \ldots, X_{k-1}\} \to \{0, \ldots, k-1\}$, mapping each variable X_i to the node j at which it has last been opened (or to 0 if it has not been opened yet). Intuitively, the strategy of Falsifier is to always force the play towards a node that is *not* in the range of S. At the initial position, set $S(X_i) = 0$ for all X_i. As the game proceeds, S is changed only at positions of form $(j, \lambda X_i \varphi)$ at which it is updated by the rule $S(X_i) := j$.

The strategy g for Falsifier is defined as a concatenation of local strategies. At the initial position, and after any initial segment of the play, Falsifier selects a node $m \leq k$ that is not in the range of S (which must exist as the range of S has size at most $k-1$) and plays according to a local strategy g_m by which he forces the game to a segment π' with $\mathrm{node}(\pi') = m$. There he again selects a value m' not in the range of the current S, and continues with a local strategy $g_{m'}$ forcing the play towards node m'.

Since ψ is non-ambiguous on $\mathcal{C}^k$ we already know that for every fixed point formula $\lambda X_\ell.\varphi \in \mathrm{cl}(\psi)$ there is at most one node j such that the position $(j, \lambda X_\ell.\varphi)$ appears in plays won by Verifier. We claim that in the play $f\hat{\ }g$ any such position appears at most twice, which means that the play is finite.

To prove this, let π be the minimal initial segment of the play $f\hat{\ }g$ in which the position $(j, \lambda X_\ell.\varphi)$ occurs. At this position, S is updated by the rule $S(X_\ell) := j$. Since ψ is guarded, the play must go through a modality before it can reach this position again. The segment π ends with a move from $(j, \langle jm \rangle \eta)$ to (m, η), with $m = \text{node}(\pi)$. We distinguish two cases.



(1) If $m \neq j$, then the position $(j, \lambda X.\varphi)$ will not occur anymore in the play $f\hat{\ }g$. Indeed at the end of the segment π Falsifier selects a local strategy $g_{m'}$ with $m' \neq j$ (as j is now in the range of S) and keeps the play away from node j until j has been removed from the range of S. But this only happens when a new fixed point definition with the same variable X_ℓ appears on the play, after which the regeneration of $(j, \lambda X_\ell.\varphi)$ is impossible anyway.

(2) If $m = j$ then it is possible to hit position $(j, \lambda X_\ell.\varphi)$ in the following segment. (For instance, it could be the case that $\eta = \lambda X_\ell.\varphi$). However, this can happen only once, since after this regeneration the play must again go through a modality before hitting position $(j, \lambda X_\ell.\varphi)$ a third time. But this is impossible because at position (m, η), Falsifier has decided that the game will now go to a node $m' \neq j$ and stay away from node j until j is removed from the range of S.

Corollary 21. *The variable hierarchy for $L_\mu^\Diamond$ is strict.*

6 Conclusion

We have established that the variable hierarchy is strict in the $\Box$-free fragment of the μ-calculus. A more ambitious goal, of course, is to determine whether the variable hierarchy is strict for the full μ-calculus. We believe that this is the case, and conjecture that the strictness is witessed by the formulae $\psi^{(k)}$.

Conjecture 22. For every $k \geq 1$, the formula $\psi^{(k)}$ is not equivalent to any formula in $L_\mu[k-1]$.

This conjecture is related to the question whether the existential preservation theorem holds for the bounded-variable fragments $L_\mu[k]$. Indeed, if any formula equivalent to ψ^k can be translated into a $\Box$-free formula without increasing the number of variables, then the above conjecture holds as a consequence of

Corollary 23. *If the existential preservation theorem holds for $L_\mu[k]$, then no formula of $L_\mu[k]$ is equivalent to $\psi^{(k+1)}$.*

As mentioned in the second section, the proof of the existential preservation theorem for L_μ in [7] is based on the μ-automata of Janin and Walukiewicz [10]. Every formula of L_μ can be translated to an equivalent μ-automaton. These automata differ from the more common alternating tree automata used, e.g., in [12], and there is no direct, inductive, translation from formulae to μ-automata. The advantage of this detour is that any μ-automaton that is closed under extensions can be transformed in a relatively straightforward way to an equivalent automaton whose transition function is existential. This modified automaton can then be translated back to a $\Box$-free formula of the μ-calculus. However, the construction of the μ-automaton involves a powerset construction and it is not clear whether the number of variables can be preserved. In fact it might well be the case that the existential preservation theorem fails for the bounded variable fragments of L_μ, as it happens for other logics, for instance first-order logic [1, 9].

References

[1] H. Andréka, J. van Benthem, and I. Németi, *Modal languages and bounded fragments of predicate logic*, Journal of Philosophical Logic, 27 (1998), 217–274.

[2] A. Arnold, *The mu-calculus alternation-depth is strict on binary trees*, RAIRO Informatique Théorique et Applications, 33 (1999), 329–339.

[3] A. Arnold and D. Niwiński, *Rudiments of μ-calculus*, North Holland, 2001.

[4] D. Berwanger, *Game logic is strong enough for parity games*, Studia Logica. Special issue on Game Logic and Game Algebra, (2002).

[5] D. Berwanger and E. Grädel, *Games and model checking for guarded logics*, in Proceedings of LPAR 2001, Lecture Notes in Computer Science Nr. 2250, Springer, 2001, 70–84.

[6] J. Bradfield, *The modal μ-calculus alternation hierarchy is strict*, Theoretical Computer Science, 195 (1998), 133–153.

[7] G. d'Agostino and M. Hollenberg, *Logical questions concerning the μ-calculus: interpolation, Lyndon, and Los-Tarski*, Journal of Symbolic Logic, 65 (2000), 310–332.

[8] A. Emerson and C. Jutla, *Tree automata, mu-calculus and determinacy*, in Proc. 32nd IEEE Symp. on Foundations of Computer Science, 1991, 368–377.

[9] E. Grädel and E. Rosen, *Preservation theorems for two-variable logic*, Mathematical Logic Quarterly, 45 (1999), 315–325.

[10] D. Janin and I. Walukiewicz, *Automata for the modal μ-calculus and related results*, in Proceedings of MFCS 95, Lecture Notes in Computer Science Nr. 969, Springer-Verlag, 1995, 552–562.

[11] M. Jurdziński, *Small progress measures for solving parity games*, in STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings, vol. 1770 of Lecture Notes in Computer Science, Springer, 2000, 290–301.

[12] O. Kupferman, M. Vardi, and P. Wolper, *An automata-theoretic approach to branching-time model checking*, Journal of the ACM, 47 (2000), 312–360.

[13] G. Lenzi, *A hierarchy theorem for the mu-calculus*, in Proceedings of the 23rd International Colloquium on Automata, Languages and Programming, ICALP '96, F. Meyer auf der Heide and B. Monien, eds., vol. 1099 of Lecture Notes in Computer Science, Springer-Verlag, July 1996, 87–97.

[14] J. Los, *On the extending of models (I)*, Fundamenta Mathematicae, 42 (1955), 38–54.

[15] R. Parikh, *The logic of games and its applications*, Annals of Discrete Mathematics, 24 (1985), 111–140.

[16] M. Pauly, *Logic for Social Software*, PhD thesis, University of Amsterdam, 2001.

[17] C. Stirling, *Bisimulation, model checking and other games.* Notes for the Mathfit instructional meeting on games and computation. Edinburgh, 1997.

[18] W. W. Tait, *A counterexample to a conjecture of Scott and Suppes*, Journal of Symbolic Logic, 24 (1959), 15–16.

[19] A. Tarski, *Contributions to the theory of models I, II*, Indagationes Mathematicae, 16 (1954), 572–588.