A Proof System for the Linear Time μ-Calculus

Christian Dax[1], Martin Hofmann[2], and Martin Lange[2]

[1] Department of Computer Science, ETH Zürich
[2] Institut für Informatik, LMU München

Abstract. The linear time μ-calculus extends LTL with arbitrary least and greatest fixpoint operators. This gives it the power to express all ω-regular languages, i.e. strictly more than LTL. The validity problem is PSPACE-complete for both LTL and the linear time μ-calculus. In practice it is more difficult for the latter because of nestings of fixpoint operators and variables with several occurrences.

We present a simple sound and complete infinitary proof system for the linear time μ-calculus and then present two decision procedures for provability in the system, hence validity of formulas. One uses nondeterministic Büchi automata, the other one a generalisation of size-change termination analysis (SCT) known from functional programming.

The main novelties of this paper are the connection with SCT and the fact that both decision procedures have a better asymptotic complexity than earlier ones and have been implemented.

1 Introduction

The linear time μ-calculus (L_μ^{lin}) [1,14] extends Pnueli's *Linear Time Temporal Logic* (LTL) with extremal fixpoints quantifiers. This increases its expressive power: L_μ^{lin} captures exactly the ω-regular languages, while the class of LTL-definable properties is only that of star-free ω-languages. L_μ^{lin} can also be seen as the modal μ-calculus which is only interpreted over infinite linear time structures, i.e. Kripke structures in which every state has exactly one successor.

The main decision problems for LTL and L_μ^{lin} have the same complexity: model checking, satisfiability and validity are all PSPACE-complete for both logics [11,14]. By *model checking* we denote, as usual, the problem to decide whether all paths of a given Kripke structure satisfy a given specification. Note that these three problems are all interreducible for linear time logics. For instance, validity is the same as model checking in a *universal Kripke structure* that has the shape of a full clique; model checking can be reduced to validity checking by modeling the given structure in a formula which is linear in the size of the structure, etc. Since these reductions do not interfere with the main difficulty in each decision problem – to find infinite regenerations of least or greatest fixpoint – we will simply refer to *decision problems*. Here we focus on the validity problem but stress the point that this approach is applicable to the other problems without major alterations, too.

The presence of nested and possibly alternating fixpoint constructs makes L_μ^{lin}'s decision problems much harder in practice than those of LTL. The fact

S. Arun-Kumar and N. Garg (Eds.): FSTTCS 2006, LNCS 4337, pp. 273–284, 2006.

that LTL formulas only contain very simple unnested fixpoints is certainly one of the reasons for LTL being well supported by successfully working tools like SPIN and NUSMV, etc.

Some decision procedures for L_μ^{lin} have been presented so far. Vardi [14] uses nondeterministic Büchi automata to decide an extension of L_μ^{lin} with temporal past operators. The time complexity of his algorithm is $2^{O(n^4)}$ where n is the size of the input formula. Stirling and Walker subsequently gave a tableau characterisation for L_μ^{lin}'s decision problems but were not concerned with complexity issues.

Bradfield, Esparza and Mader defined tableaux with simpler termination conditions. Their algorithm runs in time $2^{O(n^2 \log n)}$ but this appeals to general complexity theorems about nondeterministic space vs. deterministic time. Hence, their result is of theoretical rather than – as they say – practical use. The same holds for Kaivola's procedure [4] which runs in time $2^{O(n^2 \log n)}$ when transformed into a deterministic procedure. We remark that it was designed to be nondeterministic in the first place – the user is supposed to provide Hintikka structures manually. To the best of our knowledge, none of these existing suggestions to solve L_μ^{lin}'s decision problems have ever seen any serious attempt to be put into practice.

Here we present a simple proof system for L_μ^{lin}. A proof is an infinite tree in which each branch satisfies an additional *global* condition concerning the existence of *threads* – similar to the internal paths of [2]. Our proof system and in particular the characterisation of valid proof branches is related to the notion of pre-models and models in Streett and Emerson's work on deciding the modal μ-calculus [13], adapted to L_μ^{lin} by Vardi [14]. Indeed, a formula is invalid iff its negation is satisfiable and in this case the offending path in the generic pre-proof amounts to a model in their sense when we negate all formulas and understand a sequent as the conjunction of its formulas whereas any infinite path in a pre-proof can be extended to a pre-model.

There are some subtle differences though. States of a pre-model are always maximally consistent sets of formulas (Hintikka sets) whereas our proofs contain arbitrary sequents. Second, by considering the whole proof tree rather than individual paths in isolation the need for the perhaps mysterious concept of choice functions disappears. Of course they come back in Section 4 where we show that a simple nondeterministic parity (or Büchi) automaton (NPA/NBA) is able to accept all valid paths in a proof. They return in the form of a condensed description of rule instances fed to the NPA in addition to the sequents. We claim though that the concept is more naturally explained by arguing that the automaton must check every path in the pre-proof.

We present two different approaches to decide validity. The first one reduces this to the inclusion problem for nondeterministic Büchi automata. Depending on which complementation procedure is used we obtain an algorithm that runs in time $2^{O(n^2 \log n)}$ for example. This is easily implementable since it does not use any theorems from complexity theory. Alternatively, there is a procedure running in time $2^{O(n^4)}$ that can be implemented symbolically.

The second approach is an iterative algorithm inspired by the size-change termination (SCT) method introduced by Jones et al. [8] in the context of termination analysis. There is, effectively, a fundamental connection between termination of functional programs and decision problems for ω-automata which we will elaborate elsewhere. Here we adapt and substantially generalise the SCT method in an ad hoc fashion to our situation at hand. Validity then reduces to the problem of finding an idempotent morphism satisfying a certain property in a category generated by a finite number of morphisms. Rule applications in the proof system are regarded as morphisms with successive applications as morphism composition. Systematically exploring the set of morphisms can be done in time $2^{O(n^3)}$ but is in practice better than the automata-theoretic method as some experimental results suggest.

2 Preliminaries

Let $\mathcal{P} = \{p, q, \ldots\}$ be a set of atomic propositions, and $\mathcal{V} = \{X, Y, \ldots\}$ an infinite set of monadic second-order variables. Formulas of the linear time μ-calculus L_μ^{lin} in positive normal form are given by the following grammar.

$$\varphi \quad ::= \quad q \mid \neg q \mid X \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where $q \in \mathcal{P}$, and $X \in \mathcal{V}$. We will write σ for either μ or ν and use $l, \overline{l}$ etc. to denote literals $q, \neg q$ and their complements. We assume the reader to be familiar with the standard notions of syntactic subformulas $Sub(\varphi)$, free variables, well-named formulas, substitution $\varphi[\psi/X]$ of all occurrences of a variable, etc.

The Fischer-Ladner closure $FL(\varphi_0)$ of a L_μ^{lin}-formula φ_0 is the least set of formulas that contains φ_0 and satisfies: if $\varphi \in FL(\varphi_0)$ and

- $\varphi = \psi_1 \vee \psi_2$ or $\psi = \psi_1 \wedge \psi_2$ then $\{\psi_1, \psi_2\} \subseteq FL(\varphi)$;
- $\varphi = \bigcirc\psi$ then $\psi \in FL(\varphi)$;
- $\varphi = \sigma X.\psi$ then $\psi[\sigma X.\psi/X] \in FL(\varphi)$.

Define $|\varphi_0| := |FL(\varphi_0)|$. Note that $|FL(\varphi_0)|$ is bounded by the syntactical length of φ_0.

A *linear time structure* over $\mathcal{P}$ is a function $\mathcal{K} : \mathbb{N} \to 2^{\mathcal{P}}$ or, equally, an ω-word over the alphabet $2^{\mathcal{P}}$. The semantics of a L_μ^{lin}-formula φ, relative to $\mathcal{K}$ and an environment $\rho : \mathcal{V} \to 2^{\mathbb{N}}$ is a subset of $\mathbb{N}$, inductively defined using the Knaster-Tarski-Theorem.

$$\begin{array}{ll}
[\![q]\!]_\rho^{\mathcal{K}} := \{n \in \mathbb{N} \mid q \in \mathcal{K}(n)\} & [\![X]\!]_\rho^{\mathcal{K}} := \rho(X) \\
[\![\neg q]\!]_\rho^{\mathcal{K}} := \{n \in \mathbb{N} \mid q \notin \mathcal{K}(n)\} & [\![\bigcirc\varphi]\!]_\rho^{\mathcal{K}} := \{n \in \mathbb{N} \mid n+1 \in [\![\varphi]\!]_\rho^{\mathcal{K}}\} \\
[\![\mu X.\varphi]\!]_\rho^{\mathcal{K}} := \bigcap\{T \subseteq \mathbb{N} \mid [\![\varphi]\!]_{\rho[X \mapsto T]}^{\mathcal{K}} \subseteq T\} & [\![\varphi \vee \psi]\!]_\rho^{\mathcal{K}} := [\![\varphi]\!]_\rho^{\mathcal{K}} \cup [\![\psi]\!]_\rho^{\mathcal{K}} \\
[\![\nu X.\varphi]\!]_\rho^{\mathcal{K}} := \bigcup\{T \subseteq \mathbb{N} \mid T \subseteq [\![\varphi]\!]_{\rho[X \mapsto T]}^{\mathcal{K}}\} & [\![\varphi \wedge \psi]\!]_\rho^{\mathcal{K}} := [\![\varphi]\!]_\rho^{\mathcal{K}} \cap [\![\psi]\!]_\rho^{\mathcal{K}}
\end{array}$$

We write $\mathcal{K}, i \models_\rho \varphi$ if $i \in [\![\varphi]\!]_\rho^{\mathcal{K}}$, and $\mathcal{K} \models_\rho \varphi$ if $0 \in [\![\varphi]\!]_\rho^{\mathcal{K}}$. If φ is closed we may also drop ρ.

By deMorgan's laws and duality of μ and ν, negation $\neg$ – and then of course $\rightarrow$ and $\leftrightarrow$ – can be defined in L_μ^{lin}.

A formula φ is *valid*, written $\models \varphi$, if for all linear time structures $\mathcal{K}$, and all environments ρ: $\mathcal{K} \models_\rho \varphi$ holds. Two formulas φ and ψ are equivalent, $\varphi \equiv \psi$, if for all ρ, and all $\mathcal{K}$ we have $\mathcal{K} \models_\rho \varphi$ iff $\mathcal{K} \models_\rho \psi$.

A formula is guarded if every occurrence of a variable X is in the scope of a $\bigcirc$-operator under its quantifier μ or ν. Every L_μ^{lin} formula can be transformed into guarded form.

Approximants of a fixpoint formula $\nu X.\varphi$ are defined in the usual way: $\nu^0 X.\varphi := \mathtt{tt}$, $\nu^{k+1} X.\varphi := \varphi[\sigma^k X.\varphi/X]$, and $\nu^\omega X.\varphi := \bigwedge_{k\in\mathbb{N}} \nu^k X.\varphi$. The next result about approximants uses the fact that the semantics of a L_μ^{lin} formula is a monotone and continuous function (for infinite unions of directed sets) of type $2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ in each variable, c.f. [3].

Lemma 1. *For all linear time structures $\mathcal{K}$, all $i \in \mathbb{N}$, all environments ρ, and all $\varphi(X)$ we have: $\mathcal{K}, i \not\models_\rho \nu X.\varphi$ iff there is a $k \in \mathbb{N}$ s.t. $\mathcal{K}, i \not\models_\rho \nu^k X.\varphi$.*

3 A Proof System for the Linear Time μ-Calculus

Let φ_0 be fixed. A *sequent* is a subset Γ of $FL(\varphi_0)$. Semantically, a sequent stands for the disjunction of its members; the empty sequent is always false. We extend satisfaction by structures and validity to sequents accordingly.

A *pre-proof* for φ_0 is a possibly infinite tree whose nodes are labeled with sequents, whose root is labeled with $\vdash \varphi_0$ and which is built according to the following proof rules, later referred to as $(\vee)$, $(\wedge)$, (σ), and $(\bigcirc)$. We write $\bigcirc\Gamma$ to abbreviate $\bigcirc\gamma_1, \ldots, \bigcirc\gamma_n$ if $\Gamma = \gamma_1, \ldots, \gamma_n$.

$$\frac{\vdash \varphi, \psi, \Gamma}{\vdash \varphi \vee \psi, \Gamma} \qquad \frac{\vdash \varphi, \Gamma \qquad \vdash \psi, \Gamma}{\vdash \varphi \wedge \psi, \Gamma} \qquad \frac{\vdash \varphi[\sigma X.\varphi/X], \Gamma}{\vdash \sigma X.\varphi, \Gamma} \qquad \frac{\vdash \Gamma}{\vdash \bigcirc\Gamma, \Delta}$$

A *principal formula* in a rule application is a formula that gets transformed by this rule, e.g. $\varphi \vee \psi$ in rule $(\vee)$. Note that rule $(\bigcirc)$ can have several principal formulas.

For all sequents Γ, Δ and all rules r occurring in a pre-proof for φ_0, s.t. Γ is the conclusion of r and Δ is a premiss of r we define the *connection relation* $Con_r(\Gamma, \Delta) \subseteq FL(\varphi_0) \times FL(\varphi_0)$ as follows.

$$(\varphi, \psi) \in Con_r(\Gamma, \Delta) \quad \text{iff} \quad \begin{array}{l} \text{either } r \text{ does not transform } \varphi \text{ and } \varphi = \psi \\ \text{or } \psi \text{ results from } \varphi \text{ in the application of } r \end{array}$$

We drop the index r if the actual rule is irrelevant. Let $\pi = \Gamma_0, \Gamma_1, \ldots$ be an infinite branch in a pre-proof for φ_0 resulting from the rule applications $r_0, r_1, \ldots$. A *thread* in π is a sequence of formulas $\varphi_0, \varphi_1, \ldots$ s.t. for all $i \in \mathbb{N}$: $(\varphi_i, \varphi_{i+1}) \in Con_{r_i}(\Gamma_i, \Gamma_{i+1})$ holds. Such a thread is called a *ν-thread* if there is a $\nu X.\psi \in FL(\varphi_0)$ s.t. $\varphi_i = \nu X.\psi$ for infinitely many $i \in \mathbb{N}$, and for all $\mu Y.\psi'$ s.t. $\nu X.\psi \in Sub(\mu Y.\psi')$: there are only finitely many $i \in \mathbb{N}$ s.t. $\varphi_i = \mu Y.\psi'$. A μ-thread is defined accordingly.

The following facts about threads are not hard to see.

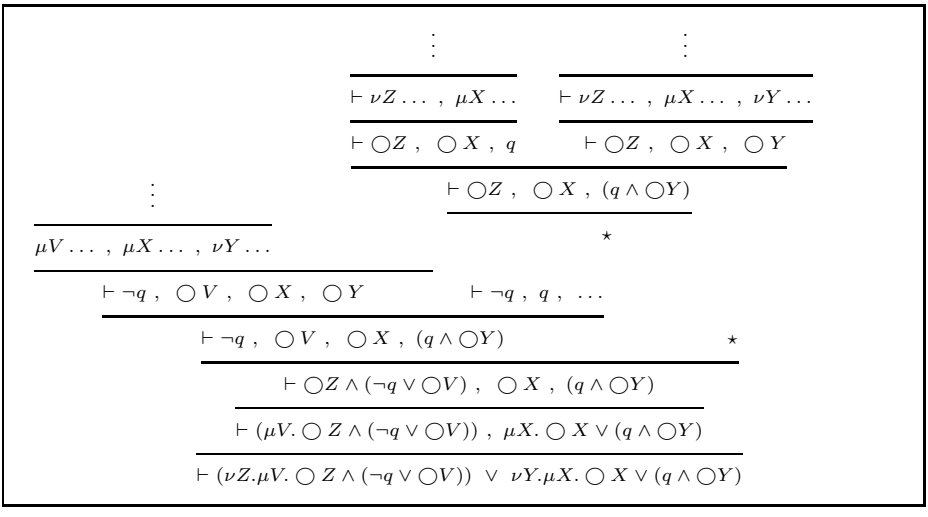

Fig. 1. Example of a proof

1. If a $\sigma X.\psi$ and a $\sigma' X'.\psi'$ occur infinitely often in a thread then $\sigma X.\psi \in Sub(\sigma' X'.\psi')$ or vice-versa.
2. Every thread is either a ν-thread or a μ-thread.

A *proof* for φ_0 is a pre-proof s.t. every finite branch ends in a sequent $l, \overline{l}, \Gamma$, and every infinite branch contains a ν-thread. We also write $\vdash \varphi_0$ to indicate that there is a proof for φ_0.

Example 1. Consider the quantifier swapping theorem

$$\models \big(\mu Z.\nu V. \bigcirc Z \vee (q \wedge \bigcirc V)\big) \rightarrow \big(\nu Y.\mu X. \bigcirc X \vee (q \wedge \bigcirc Y)\big)$$

This can be shown to be valid using principles from fixpoint theory. It is also intuitively valid: the premiss of the implication expresses "after some point, q always holds" and the conclusion says "q holds infinitely often".

In positive normal form this is written as $\varphi = (\nu Z.\mu V. \bigcirc Z \wedge (\neg q \vee \bigcirc V)) \vee (\nu Y.\mu X. \bigcirc X \vee (q \wedge \bigcirc Y))$. A proof for φ is sketched in Fig. 1. In order to save space, not all rule applications are listed explicitly and a variable is used to denote its unique fixpoint formula. On each infinite branch of this proof, either $\nu Z. \ldots$ or $\nu X. \ldots$ can be followed along a thread.

Theorem 1. *For all closed and guarded $\varphi \in L_\mu^{\mathrm{lin}}$: if $\models \varphi$ then $\vdash \varphi$.*

Proof. Suppose $\models \varphi$. Let us replace $(\bigcirc)$ by the following restriction.

$$\frac{\vdash \Gamma}{\vdash \bigcirc \Gamma, l_1, \ldots, l_k} \; \nexists i, j : l_i = \overline{l_j}$$

Now all rules preserve and reflect validity. Therefore, systematic backwards application of the rules leads to a pre-proof P of φ comprising valid sequents only.

We claim that P is a proof. Note that guardedness means that all fixpoints must have been unfolded prior to an application of restricted ($\bigcirc$) so no "round-robin" policy or similar is needed in the construction of P.

Take any infinite branch $\pi = \Delta_0, \Delta_1, \ldots$ of P. We will now exhibit a ν-thread in π. For every $i \in \mathbb{N}$ let $f(i)$ be the number of applications of rule ($\bigcirc$) in π before Δ_i. We construct a linear time structure $\mathcal{K}$ as follows:

$$\forall i \in \mathbb{N} \text{ with } f(i) \neq f(i+1): \quad \mathcal{K}(f(i)) = \{\bar{l}_1, \ldots, \bar{l}_k\} \text{ iff } \Delta_i = \bigcirc\Gamma, l_1, \ldots, l_k$$

Consider for each Δ_i the formulas of Δ_i satisfied by $\mathcal{K}, f(i)$. Call them "true formulas". For each Δ_i there is at least one such true formula, because each Δ_i is a valid disjunction.

Every true formula is linked by the connection relation to a true formula in the preceding sequent; König's lemma thus delivers a thread comprising true formulas only. More formally, we obtain a sequence $\varphi_i \in \Delta_i$ such that $\varphi_0 = \varphi$ and $(\varphi_i, \varphi_{i+1}) \in \mathit{Con}(\Delta_i, \Delta_{i+1})$ and $\mathcal{K}, f(i) \models \varphi_i$. Assume that $(\varphi_i)_i$ is a μ-thread. There is an $i \in \mathbb{N}$ and a $\mu X.\psi \in \mathit{Sub}(\varphi)$ s.t. $\mathcal{K}, f(i) \models \mu X.\psi$. Furthermore, no greater $\nu Y.\psi'$ occurs on this thread after position i. According to Lemma 1 there is a $k \in \mathbb{N}$ s.t. $\mathcal{K}, f(i) \models \mu^k X.\psi$. Now note that the connection relation follows the definition of the approximants. Hence, by preservation of satisfaction along this thread, there must be a $i' > i$, s.t. $\mathcal{K}, f(i') \models \mu^0 X.\psi$ which is impossible. So, the thread $(\varphi_i)_i$ is a ν-thread as required. □

We remark without proof that the proof system is also complete for non-guarded formulas.

Let $\varphi_0 \in L_\mu^{\mathrm{lin}}$ and $\nu X_1.\psi_1, \ldots, \nu X_n.\psi_n$ all ν-quantified formulas in $FL(\varphi_0)$, ordered s.t. $\nu X_i.\psi_i \in \mathit{Sub}(\psi_j)$ implies $i > j$. A *ν-signature* is a tuple $\zeta = (k_1, \ldots, k_n) \in (\mathbb{N} \cup \{\omega\})^n$. Note that the lexicographic ordering $<$ on ν-signatures is well-founded. We write $\zeta(i)$ for the i-th component of ζ, and $\mathcal{K}, i \models_\zeta \varphi$ if $\mathcal{K}, i$ is a model of the formula that results from φ when every $\nu X_i.\psi_i$ is interpreted by $\nu^{\zeta(i)} X_i.\psi_i$.

Theorem 2. *For all closed $\varphi \in L_\mu^{\mathrm{lin}}$: if $\not\models \varphi$ then $\not\vdash \varphi$.*

Proof. Suppose $\not\models \varphi$ but P is a proof for φ. Then there is a $\mathcal{K}$ s.t. $\mathcal{K}, 0 \not\models \varphi$. This can be used to construct a path $\pi = \Gamma_0, \Gamma_1, \ldots$ with inferences $r_0, r_1, \ldots$ in P, and a sequence $t_0 \leq t_1 \leq \ldots$ of positions in $\mathcal{K}$, s.t. $\mathcal{K}, t_i \not\models \Gamma_i$ (I), and whenever $(\alpha, \beta) \in \mathit{Con}_{r_i}(\Gamma_i, \Gamma_{i+1})$ and $\mathcal{K}, t_i \not\models_\zeta \alpha$ then $\mathcal{K}, t_{i+1} \not\models_\zeta \beta$ (II).

Let $\Gamma_0 := \varphi$ and $t_0 := 0$. If Γ_i and t_i have been constructed we regard the inference r_i leading to Γ_i (note that Γ_i cannot be an axiom). If $r_i = (\bigcirc)$ then $t_{i+1} := t_i + 1$. We put $t_{i+1} := t_i$ in all other cases. If $r_i \neq (\wedge)$ then Γ_i has a unique premiss $\Delta =: \Gamma_{i+1}$. In the case of $(\wedge)$ let $\psi_1 \wedge \psi_2 \in \Gamma_i$ be the principal formula of r_i. Let ζ be the least ν-signature s.t. $\mathcal{K}, t_i \not\models_\zeta \psi_1 \wedge \psi_2$ (it exists by Lemma 1 and (I)). Let Γ_{i+1} be the j-th premiss of r_i where $j \in \{1, 2\}$ s.t. $\mathcal{K}, t_i \not\models_\zeta \psi_j$. Clearly, this construction guarantees condition (II).

Since P is a proof, π must contain a ν-thread $(\varphi_i)_i$. For each $i \in \mathbb{N}$ let ζ_i be the minimal ν-signature s.t. $\mathcal{K}, t_i \not\models_{\zeta_i} \varphi_i$. Since $(\varphi_i, \varphi_{i+1}) \in \mathit{Con}_{r_i}(\Gamma_i, \Gamma_{i+1})$

we have $\zeta_{i+1} \leq \zeta_i$. Since there is an outermost fixpoint formula $\nu Z.\psi$ that gets unfolded infinitely often in this thread, there are infinitely many i s.t. $\zeta_i > \zeta_{i+1}$ which is a contradiction to the wellfoundedness of $<$. □

4 Deciding Validity I: Automata-Theoretic Method

We regard rule applications in a pre-proof for a L_μ^{lin} formula φ_0 as symbols of a finite alphabet. Formally, let $\Sigma_{\varphi_0} := \{\ \mathtt{L}(\varphi \wedge \psi), \mathtt{R}(\varphi \wedge \psi) \ \mid\ \varphi \wedge \psi \in FL(\varphi_0)\ \} \cup \{\mathtt{N}\} \cup \{\ \mathtt{P}(\varphi) \ \mid\ \varphi \in FL(\varphi_0)$ is of the form $\psi_1 \vee \psi_2, \sigma X.\psi$, or $\bigcirc\psi\ \}$.

An infinite branch $\pi = \Gamma_0, \Gamma_1, \ldots$ in a pre-proof for φ_0 induces a word $\pi' = r_0, r_1, \ldots \in \Sigma_{\varphi_0}^\omega$ in a straight-forward way:

$$r_i \ := \ \begin{cases} \mathtt{L}(\varphi \wedge \psi), & \text{if } \varphi \wedge \psi \text{ is principal in } \Gamma_i, \Gamma_{i+1} \text{ is left premiss of } \Gamma_i \\ \mathtt{R}(\varphi \wedge \psi), & \text{if } \varphi \wedge \psi \text{ is principal in } \Gamma_i, \Gamma_{i+1} \text{ is right premiss of } \Gamma_i \\ \mathtt{P}(\varphi), & \text{if } \varphi \text{ is principal in } \Gamma_i \text{ and not of the form } \bigcirc\varphi' \\ \mathtt{N}, & \text{if } (\Gamma_i, \Gamma_{i+1}) \text{ is an instance of } (\bigcirc) \end{cases}$$

We will not distinguish formally between a branch π and its induced ω-word π' over Σ_{φ_0}.

Next we define an NPA that accepts exactly those branches which contain a ν-thread. Let $\varphi_0 \in L_\mu^{\text{lin}}$, and define $\mathcal{A}_{\varphi_0} := (Q, \Sigma_{\varphi_0}, q_0, \delta, \Omega)$ where $Q := FL(\varphi_0)$ is the set of states with starting state $q_0 := \varphi_0$. The priority function $\Omega : Q \to \mathbb{N}$ is defined inductively as $\Omega(\psi_1 \vee \psi_2) = \Omega(\psi_1 \wedge \psi_2) := \max\{\Omega(\psi_1), \Omega(\psi_2)\}$; $\Omega(\bigcirc\varphi) := \Omega(\varphi)$; $\Omega(\sigma X.\varphi) := \Omega(\varphi)$ if $\Omega(\varphi)$ is odd and $\sigma = \mu$, or $\Omega(\varphi)$ is even and $\sigma = \nu$, and $\Omega(\sigma X.\varphi) := \Omega(\varphi) + 1$ otherwise; and $\Omega(\psi) := 0$ in all other cases. Here we assume that an NPA accepts a word if it has an accepting run in which the *greatest* priority occurring infinitely often is *even*.

Intuitively, $\mathcal{A}_{\varphi_0}$ traces a thread. The priority function ensures that the underlying word is accepted only if the guessed thread is a ν-thread. The transition relation therefore simply resembles the connection relation:

$$\begin{array}{lll} \delta(\psi, r) & := \{\psi\} & \text{if } r \notin \{\mathtt{P}(\psi), \mathtt{L}(\psi), \mathtt{R}(\psi)\} \\ \delta(\psi_1 \vee \psi_2, \mathtt{P}(\psi_1 \vee \psi_2)) & := \{\psi_1, \psi_2\} & \delta(\bigcirc\psi, \mathtt{N}) := \{\psi\} \\ \delta(\psi_1 \wedge \psi_2, \mathtt{L}(\psi_1 \wedge \psi_2)) & := \{\psi_1\} & \delta(\bigcirc\psi, r) := \{\bigcirc\psi\} \quad \text{if } r \neq \mathtt{N} \\ \delta(\psi_1 \wedge \psi_2, \mathtt{R}(\psi_1 \wedge \psi_2)) & := \{\psi_2\} & \delta(\sigma X.\varphi, \mathtt{P}(\sigma X.\varphi)) := \{\varphi[\sigma X.\varphi / X]\} \end{array}$$

Clearly, $|\mathcal{A}_{\varphi_0}|$, the number of states of $\mathcal{A}_{\varphi_0}$ is $|\varphi_0|$.

Lemma 2. *For all closed $\varphi_0 \in L_\mu^{\text{lin}}$ and all infinite branches π of a pre-proof for φ_0: $\pi \in L(\mathcal{A}_{\varphi_0})$ iff π contains a ν-thread.*

Proof. Let $\pi = \Gamma_0, \Gamma_1, \ldots$ be an infinite branch in a pre-proof for φ_0.

($\Leftarrow$) Suppose $\varphi_0, \varphi_1, \ldots$ is a ν-thread in π. Since δ is defined in accordance to the connection relation, this thread is also a run of $\mathcal{A}_{\varphi_0}$. By assumption, the outermost subformula of the form $\sigma X.\psi$ that occurs infinitely often in this thread is of type ν. Now note that the priority of an automaton state $\sigma X.\psi$ is even iff

$\sigma = \nu$, and outer formulas have greater priorities than inner ones. Hence, the greatest priority occurring infinitely often in this run is even, i.e. $\pi \in L(\mathcal{A}_{\varphi_0})$.

($\Rightarrow$) This is proved in the same way as the other direction. □

Furthermore, we define a (deterministic) Büchi automaton $\mathcal{B}_\varphi$ that accepts all the words which form a branch in a pre-proof for φ. In order to avoid notational clutter we simply assume that every branch in a pre-proof is infinite. Note that finite branches can be modeled by introducing a new final state in the automaton with a self-loop under any alphabet symbol.

Let $\mathcal{B}_\varphi := (2^{FL(\varphi)}, \Sigma_\varphi, \{\varphi\}, \delta, F)$ with $F := 2^{FL(\varphi)}$, and $\Delta \in \delta(\Gamma, r)$ iff Δ is a premiss of Γ in an application of rule r. The following is a direct consequence of the definition of a proof and Lemma 2.

Proposition 1. *For all $\varphi \in L_\mu^{\mathrm{lin}}$: $\vdash \varphi$ iff $L(\mathcal{B}_\varphi) \subseteq L(\mathcal{A}_\varphi)$.*

This shows that validity in L_μ^{lin} can be decided using this proof system in an optimal way matching the known PSPACE lower bound [11].

Theorem 3. *Deciding whether or not $\vdash \varphi$ holds for a given $\varphi \in L_\mu^{\mathrm{lin}}$ is in PSPACE.*

Proof. According to Proposition 1 it suffices to check the language $L(\mathcal{B}_\varphi) \cap \overline{L(\mathcal{A}_\varphi)}$ for non-emptiness. Let $n := |\varphi|$. Note that $|\mathcal{B}_\varphi| \leq 2^n$ and $|\mathcal{A}_\varphi| \leq n$. Using well-known automata-theoretic constructions and Savitch's Theorem this boils down to the emptiness test of an automaton $\mathcal{B} \times \overline{\mathcal{A}}$ which can be done in PSPACE. □

Proposition 1 yields a generic automata-theoretic method for deciding validity. We will compare various complementation and non-emptiness procedures for NBAs w.r.t. the incurring complexities. Note that every state of $\mathcal{B}_\varphi$ is final. Hence, the automaton $\mathcal{B}_\varphi \times \overline{\mathcal{A}_\varphi}$ can always be built in a simple product construction and is of the same type as $\overline{\mathcal{A}_\varphi}$.

construction	type of $\overline{\mathcal{A}_\varphi}$	$\lvert\mathcal{B}_\varphi \times \overline{\mathcal{A}_\varphi}\rvert$	emptiness test
Safra [10]	det. Streett	$2^{O(n^2 \log n)}$	$2^{O(n^2 \log n)}$
Sistla/Vardi/Wolper [12]	nondet. Büchi	$2^{O(n^4)}$	$2^{O(n^4)}$
Klarlund [5]	nondet. Büchi	$2^{O(n^2 \log n)}$	$2^{O(n^2 \log n)}$
Kupferman/Vardi [7]	weak alt. Büchi	$O(n^4)$	$2^{O(n^4)}$
Kupferman/Vardi [6]	weak alt. Büchi	$O(n^n)$	$2^{O(n^n)}$
Piterman [9]	det. parity	$2^{O(n^2 \log n)}$	$2^{O(n^2 \log n)}$

The index of the Streett or parity automaton is $O(n^2)$ in both cases. Note that the table lists the running times of a deterministic algorithm not using general theorems from complexity theory. We remark that using either of Safra's, Klarlund's or Piterman's construction improves asymptotically over Vardi's decision procedure for L_μ^{lin}. It also improves over the other $2^{O(n^2 \log n)}$ procedures mentioned in the introduction by being *a priori* deterministic. Furthermore, the procedures using Kupferman and Vardi's complementation can be implemented symbolically.

5 Deciding Validity II: Category-Theoretic Method

Let P be the (finite) set of priorities assigned to subformulas of φ_0 by the function Ω in Section 4. Let Γ and Δ be sequents. A morphism f from Γ to Δ written $f : \Gamma \to \Delta$ is a subset of $\Gamma \times \Delta \times P$. In this case, Γ is the domain of f and Δ is the codomain of f. If $f : \Gamma \to \Delta$ and $\Delta \to \Theta$ then the composition $f; g : \Gamma \to \Theta$ is the morphism defined by

$$f; g = \{(\gamma, \theta, p) \mid \exists \delta\ p_1\ p_2.(\gamma, \delta, p_1) \in f \wedge (\delta, \theta, p_2) \in g \wedge p = \max(p_1, p_2)\}$$

The identity morphism $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ is given by $\mathrm{id}_\Gamma = \{(\gamma, \gamma, 0) \mid \gamma \in \Gamma\}$.

It is clear that composition is associative with identities as neutral elements and that therefore the sequents with morphisms form a category. If M is a set of morphisms we denote by $C(M)$ the set of morphisms obtained by closing M under composition and adding identities, i.e., the category generated by M. Notice that if M is a finite set so is $C(M)$ because there is only a finite number of sequents and morphisms.

A morphism $f : \Gamma \to \Delta$ is idempotent if $\Gamma = \Delta$ and $f; f = f$. An idempotent morphism f is *bad* if it does not contain a link of the form (φ, φ, p) with p even. A morphism should be viewed as a connection relation whose links are labeled with priorities.

Suppose that r is an instance of a rule occurring in a pre-proof of φ_0 with conclusion Γ and Δ one of its premisses. We define the morphism $f_{(\Gamma, r, \Delta)} : \Gamma \to \Delta$ by

$$f_{\Gamma, r, \Delta} = \{(\gamma, \delta, \Omega(\gamma)) \mid (\gamma, \delta) \in Con_r(\Gamma, \Delta)\}$$

If π is a finite branch occurring in a pre-proof then we obtain a morphism f_π by composing the morphisms $f_{\_, r, \_}$ that are associated with the sequents and rules occurring along π. If π begins at sequent Γ and ends at Δ then $(\gamma, \delta, p) \in f_\pi$ iff there is a run of $\mathcal{A}_{\varphi_0}$ on π beginning in state γ, ending in state δ and exhibiting p as the highest priority along this run.

Now let P be the generic pre-proof obtained as in the proof of Theorem 3. Note that in this pre-proof any sequent uniquely determines the proof rule which is (backwards-)applied to it.

Theorem 4. *Let M be the set of morphisms of the form $f_{\Gamma, r, \Delta}$ where r is a rule instance contained in the generic pre-proof P of φ_0. The following are equivalent.*
(a) φ_0 is valid.
(b) P is a proof.
(c) The closure $C(M)$ of M by composition contains no bad idempotent.

Proof. The equivalence between (a) and (b) is a direct consequence of Lemma 2 and Theorem 1. The interesting part is the equivalence between (b) and (c) and it is here that we draw inspiration from the graph-theoretic algorithm for size-change termination in [8] and in particular closely follow their proof idea.

(b)⇒(c) by contraposition: suppose that $C(M)$ contains a bad idempotent $f : \Delta \to \Delta$. Let π be the finite path in the generic proof P that led to f's being in $C(M)$. We use here the fact that every sequent uniquely determines its

proof rule. Let ρ be a finite path in P leading from $\{\varphi_0\}$ to Δ and consider the infinite path $\rho;\pi;\pi;\pi;\ldots$ which is contained in P. We claim that this path is not accepted by $\mathcal{A}_{\varphi_0}$. Assume for a contradiction that there is an accepting run with n the highest (even) priority. Since Δ is finite there must exist $\delta \in \Delta$ such that the accepting run goes through δ and after consuming $\pi^i := \pi;\pi;\ldots;\pi$ (i times) for some $i > 0$ goes through δ again and moreover, the highest priority encountered along π^i is n. But this means that $(\delta, \delta, n) \in f$ contradicting the assumption that f was bad.

(c)⇒(b) Assume that $C(M)$ does not contain a bad idempotent. Let π be an infinite path in P. For $i < j$ let $\pi_{i,j}$ be the finite portion of π from i to j. By Ramsey's theorem there exists an infinite subset $U \subseteq \mathbb{N}$ and a morphism f such that $f_{\pi_{i,j}} = f$ whenever $i, j \in U$. It follows that f is idempotent. If f contains a link (δ, δ, n) with n even then we get a successful run on π with highest priority n simply by going through δ at each position in U and following the construction of $f_{\pi_{i,j}}$ in between. □

Theorem 4 directly leads to an algorithm for deciding validity of formulas: simply compute the set of morphisms M occurring in the generic pre-proof of φ_0, then iteratively calculate $C(M)$ and then look for a bad idempotent in $C(M)$. Of course, in practice one checks for bad idempotents already during the construction of M and $C(M)$ terminating the process immediately upon encountering one. The resulting algorithm is not in PSPACE since the size of $C(M)$ is exponential: $|C(M)| \leq 2^{n^2 \cdot p + 2n}$ where n is the size of the input formula and p is the highest priority of any subformula. Since $p \leq n$, and the runtime of our algorithm is quadratic in $|C(M)|$, it is also bounded by $2^{O(n^3)}$. We note that this also improves asymptotically on the runtime of Vardi's decision procedure for L_μ^{lin} [14].

6 Experimental Results

We have implemented two exponential time algorithms – the one based on an explicit computation of $C(M)$ and the one testing emptiness of a deterministic parity automaton using Piterman's determinisation procedure – in OCAML. In the following we present some experimental results obtained on three families of formulas.

$$\begin{aligned} \mathit{Include}_n \;:=\; & \nu X.\big(\underbrace{q \wedge \bigcirc(q \wedge \bigcirc(\ldots \bigcirc (q}_{2n \text{ times}} \wedge \bigcirc\,(\neg q \wedge \bigcirc X))\ldots))) \\ & \rightarrow\; \nu Z.\mu Y.(\neg q \wedge \bigcirc Z) \vee (q \wedge \bigcirc(q \wedge \bigcirc Y)) \end{aligned}$$

describes the valid statement $((aa)^n b)^\omega \subseteq ((aa)^* b)^\omega$, where the alphabet symbol a is the label $\{q\}$ and b is $\emptyset$. $\mathit{Include}_n$ is not LTL-definable for any $n \in \mathbb{N}$.

The next family is $\mathit{Nester}_n := \psi \vee \neg\psi$ where

$$\psi := \mu X_1.\nu X_2.\mu X_3.\ldots.\sigma X_n.q_1 \vee \bigcirc\Big(X_1 \wedge \big(q_2 \vee \bigcirc(X_2 \wedge \ldots (q_n \vee \bigcirc X_n)\ldots)\big)\Big)$$

	$Include_n$		$Nester_n$		$Counter_n$	
n	$\|C(M)\|$	search	$\|C(M)\|$	search	$\|C(M)\|$	search
0	2,545	1,285	—	—	5	638
1	11,965	17,203	9	79	299	18,564
2	28,866	44,903	2,154	23,589	1,333	195,989
3	50,057	83,864	2,030,259	†	34,401	1,666,281
4	77,189	135,220	†		379,356	12,576,760
5	110,242	198,971			†	†

Fig. 2. Complexity measures for some example formulas

It is clearly valid and is chosen as an example with several alternating fixpoint constructs.

$$Counter_n \;:=\; (\bigvee_{i=0}^{n} \neg c_i) \;\vee\; \Big(\mu X.\bigcirc X \;\vee\; (c_0 \nleftrightarrow \bigcirc\neg c_0) \vee \bigvee_{i=1}^{n} \bigcirc c_i \;\leftrightarrow\; (c_i \wedge \neg c_{i-1}) \vee (c_{i-1} \wedge (\bigcirc c_{i-1} \leftrightarrow c_i))\Big)$$

is not valid and is chosen because its smallest countermodel has size 2^{n+1}. Note that $\neg Counter_n$ formalises an $(n+1)$-bit counter.

Figure 2 presents empirical measures for the complexity of both procedures on the example formulas above. The columns labeled $|C(M)|$ contain the number of examined morphisms. Note that this is the number of all possible morphisms unless the input formula is not valid. The columns labeled "search" contain the number of search steps done in the emptiness test on the DPA $\mathcal{B} \times \overline{\mathcal{A}}_\varphi$. This is in general quadratic in the size of the automaton. The runtime was always around a few minutes but of course space is the limiting resource here. A dagger marks the tasks where our 1GB PCs ran out of memory.

7 Further Work

We would like to carry out a more systematic study of the practical usefulness of either algorithm. The examples from Section 6 were deliberately chosen to stress-test our approach. It may well be that formulas of the form $\varphi \Rightarrow \psi$ where φ describes an implementation of a system, e.g., a Mutex algorithm and where ψ is a specification of low quantifier nesting depth are feasible up to a much larger size. Should such experiments turn out promising one could then consider improving the treatment of the propositional part using BDDs or SAT-solvers, as well as using a symbolic implementation of the automata-theoretic algorithm. Notice namely that our decision procedures deal with propositional tautologies or consequences rather inefficiently basically by proof search in sequent calculus.

Also of interest could be optimisations using heuristics to guide the search for a countermodel as well as improvements on the theoretical side that reduce the size of the entire search space.

Furthermore, the proof system of Section 3 can be quite straightforwardly extended to capture validity of the modal μ-calculus: simply replace rule ($\bigcirc$) with

$$\frac{\varphi, \Gamma}{[a]\varphi, \langle a\rangle\Gamma, \Delta}$$

This, however, introduces non-determinism (if a sequent contains several $[a]$-formulas), and countermodels become genuine trees. The automata-theoretic decision procedure of Section 4 can, in theory, easily be extended. Using Piterman's construction to determinise the automaton that recognises ν-threads, the product of this and the proof system becomes a parity game. Hence, validity in the modal μ-calculus can be solved using parity game solvers. On the other hand, whether or not the category-theoretical method of Section 5 can be extended to this framework is a nontrivial question.

References

1. H. Barringer, R. Kuiper, and A. Pnueli. A really abstract concurrent model and its temporal logic. In *Conf. Record of the 13th Annual ACM Symp. on Principles of Programming Languages, POPL'86*, pages 173–183. ACM, 1986.
2. J. C. Bradfield, J. Esparza, and A. Mader. An effective tableau system for the linear time μ-calculus. In *Proc. 23rd Int. Coll. on Automata, Languages and Programming, ICALP'96*, volume 1099 of *LNCS*, pages 98–109. Springer, 1996.
3. C. Dax. Games for the linear time μ-calculus. Master's thesis, Dep. of Computer Science, University of Munich, 2006. available from `http://www.tcs.ifi.lmu.de/lehre/da_fopra/Christian_Dax.pdf`.
4. R. Kaivola. A simple decision method for the linear time μ-calculus. In *Proc. Int. Workshop on Structures in Conc. Theory, STRICT'95*, pages 190–204, 1995.
5. N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd Annual Symp. on Foundations of Computer Science, FOCS'91*, pages 358–367. IEEE, 1991.
6. O. Kupferman and M. Y. Vardi. Weak alternating automata and tree automata emptiness. In *Proc. 30th Annual ACM Symp. on Theory of Computing, STOC'98*, pages 224–233. ACM Press, 1998.
7. O. Kupferman and M. Y. Vardi. Weak alternating automata are not that weak. *ACM Transactions on Computational Logic*, 2(3):408–429, 2001.
8. C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. *j-SIGPLAN*, 36(3):81–92, 2001.
9. N.Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Proc. 21st Ann. IEEE Symp. on Logic in Computer Science, LICS'06*. IEEE Computer Society Press, 2006. To appear.
10. S. Safra. On the complexity of ω-automata. In *Proc. 29th Symp. on Foundations of Computer Science, FOCS'88*, pages 319–327. IEEE, 1988.
11. A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the Association for Computing Machinery*, 32(3):733–749, 1985.
12. A. P. Sistla, M. Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *TCS*, 49(2–3):217–237, 1987.
13. R. S. Streett and E. A. Emerson. An automata theoretic decision procedure for the propositional μ-calculus. *Information and Computation*, 81(3):249–264, 1989.
14. M. Y. Vardi. A temporal fixpoint calculus. In *Proc. Conf. on Principles of Programming Languages, POPL'88*, pages 250–259. ACM Press, 1988.