# Simplifying the modal mu-calculus alternation hierarchy

J. C. Bradfield

*Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, Edinburgh, United Kingdom, EH9 3JZ (email: jcb@dcs.ed.ac.uk)*

**Abstract:** In [Bra96], the strictness of the modal mu-calculus alternation hierarchy was shown by transferring a hierarchy from arithmetic; the latter was a minor corollary of a highly technical analysis of [Lub93]. In this paper, we give an entirely elementary proof of that hierarchy; further, simple examples of strict alternation depth $n$ formulae can be constructed, which in turn give very simple examples to separate the modal hierarchy. In addition, the winning strategy formulae of parity games are shown to be such examples.

## 1 Introduction

The modal mu-calculus, or Hennessy–Milner logic with fixpoints, is a popular logic for expressing temporal properties of systems. It was first studied by Kozen in [Koz83], and since then there has been much work on both theoretical and practical aspects of the logic. The feature of the logic that gives it both its simplicity and its power is that it is possible to have mutually dependent minimal and maximal fixpoint operators. This makes it simple, as the fixpoints are the only non-first-order operators, and powerful, as by such nesting one can express complex properties such as 'infinitely often' and fairness. A measure of the complexity of a formula is the *alternation depth*, that is, the number of alternating blocks of minimal/maximal fixpoints. Formulae of alternation depth higher than 2 are notoriously hard to understand, and in practice one rarely produces them—not least because they are so hard to understand. For some years, it was not even known whether formulae of high alternation depth were necessary, that is, whether the *alternation hierarchy* was indeed a strict hierarchy of expressive power—a problem with several interesting ramifications, as well as its intrinsic interest. In 1996 the strictness of the hierarchy was established by the present author [Bra96], and independently by Lenzi [Len96]. The proof technique in [Bra96] relied on the existence of a similar fixpoint alternation hierarchy in arithmetic with fixpoints (mu-arithmetic). Mu-arithmetic appears, somewhat surprisingly, not to have been studied in the recursion-theoretic literature, apart from the deep and technical recursion-theoretic study of mu-arithmetic by Robert Lubarsky [Lub93], which incidentally implies the hierarchy. Thus the proof of [Bra96] was not self-contained; furthermore, it was apparently not feasible to exhibit examples of strict alternation depth $n$ formulae, as the strict mu-arithmetic formulae of [Lub93] are not constructible with any reasonable amounts of paper, ink and patience—there is only a high level description of the complex coding required, which not only establishes the hierarchy but also gives a precise characterization in terms of large admissible ordinals.

The purpose of this paper is to mend these drawbacks of [Bra96]. The first result is an elementary proof of the alternation hierarchy in mu-arithmetic. This proof uses the standard technique for recursion-theoretic hierarchies; thus we remove the reliance on [Lub93]. Furthermore, the proof constructs very simple examples of strict level $n$ formulae of mu-arithmetic; and by using a simplified version of the techniques of [Bra96], we are able to construct even simpler examples of strict level $n$ modal mu-calculus formulae. These examples are of just the form that one expects, if one is a modal mu-calculus hacker. In addition, we can also show that the formulae defining the existence of a winning strategy in a parity game are examples of strict formulae—indeed, a referee has observed that this can be shown already from [Bra96].

## 2 Preliminaries

### *2.1 Modal mu-calculus*

We assume some familiarity with the modal mu-calculus, so in this section we give brief definitions to establish notations and conventions. Expository material on the modal mu-calculus may be found in [Bra91,Sti91].

The modal mu-calculus, with respect to some countable set $\mathscr{L}$ of *labels*, has formulae $\Phi$ defined inductively thus: variables $Z$ and the boolean constants $\mathtt{tt}$, $\mathtt{ff}$ are formulae; if $\Phi_1$ and $\Phi_2$ are formulae, so are $\Phi_1 \vee \Phi_2$ and $\Phi_1 \wedge \Phi_2$; if $\Phi$ is a formula and $l$ a label, then $[l]\Phi$ and $\langle l\rangle\Phi$ are formulae; and if $\Phi$ is a formula and $Z$ a variable, then $\mu Z.\,\Phi$ and $\nu Z.\,\Phi$ are formulae.

Note that we adopt the convention that the scope of the binding operators $\mu$ and $\nu$ extends as far as possible. For consistency, we also apply this convention to the $\forall$ and $\exists$ of first-order logic, writing $\forall x.\,(\exists y.\,P) \vee Q$ rather than the logicians' traditional $\forall x\,[\exists y\,[P] \vee Q]$.

Observe that negation is not in the language, but any closed mu-formula can be negated by using the usual De Morgan dualities—$\mu$ and $\nu$ are dual by $\neg\mu Z.\,\Phi(Z) = \nu Z.\,\neg\Phi(\neg Z)$. Where necessary, we shall assume that free variables can be negated just by adjusting the valuation. We shall use $\Rightarrow$ etc. freely, though we must ensure that bound variables only occur positively.

We use the symbol $\mu\!\!\nu$ to mean '$\mu$ or $\nu$ as appropriate'.

Given a labelled transition system $\mathscr{T} = (\mathscr{S}, \mathscr{L}, \longrightarrow)$, where $\mathscr{S}$ is a set of states, $\mathscr{L}$ a set of labels, and $\longrightarrow \subseteq \mathscr{S} \times \mathscr{L} \times \mathscr{S}$ is the transition relation (we write $s \xrightarrow{l} s'$), and given also a valuation $\mathscr{V}$ assigning subsets of $\mathscr{S}$ to variables, the denotation $\|\Phi\|^{\mathscr{T}}_{\mathscr{V}} \subseteq \mathscr{S}$ of a mu-calculus formula $\Phi$ is defined in the obvious way for the variables and booleans, for the modalities by

$$\|[l]\Phi\|^{\mathscr{T}}_{\mathscr{V}} = \{\, s \mid \forall s'.\, s \xrightarrow{l} s' \;\Rightarrow\; s' \in \|\Phi\|^{\mathscr{T}}_{\mathscr{V}} \,\}$$

$$\|\langle l\rangle\Phi\|^{\mathscr{T}}_{\mathscr{V}} = \{\, s \mid \exists s'.\, s \xrightarrow{l} s' \;\wedge\; s' \in \|\Phi\|^{\mathscr{T}}_{\mathscr{V}} \,\}\,,$$

and for the fixpoints by

$$\|\mu Z.\,\Phi\|^{\mathcal{T}}_{\mathcal{V}} = \bigcap\{\, S \subseteq \mathcal{S} \mid \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z:=S]} \subseteq S \,\}$$
$$\|\nu Z.\,\Phi\|^{\mathcal{T}}_{\mathcal{V}} = \bigcup\{\, S \subseteq \mathcal{S} \mid S \subseteq \|\Phi\|^{\mathcal{T}}_{\mathcal{V}[Z:=S]} \,\}\,.$$

It is often useful to think of $\mu Z$ and $\nu Z$ as meaning respectively finite and infinite looping from $Z$ back to $\mu Z$ ($\nu Z$) as one 'follows a path of the system through the formula'. Examples of properties expressible by the mu-calculus are 'always (on $a$-paths) $P$', as $\nu Z.\, P \wedge [a]Z$, 'eventually (on $a$-paths) $P$', as $\mu Z.\, P \vee \langle a\rangle Z$, and 'there is an $\{a, b\}$-path along which $b$ happens infinitely often', as $\nu Y.\, \mu Z.\, \langle b\rangle Y \vee \langle a\rangle Z$. (For the latter, we can loop around $Y$ for ever, but each internal loop round $Z$ must terminate.)

There are several notions of alternation. The naive notion is simply to count syntactic alternations of $\mu$ and $\nu$, resulting in the following definition: A formula $\Phi$ is said to be in the classes $\Sigma^{S\mu}_0$ and $\Pi^{S\mu}_0$ iff it contains no fixpoint operators ('$S$' for 'simple' or 'syntactic'). The class $\Sigma^{S\mu}_{n+1}$ is the least class containing $\Sigma^{S\mu}_n \cup \Pi^{S\mu}_n$ and closed under the following operations: (i) application of the boolean and modal combinators; (ii) the formation of $\mu Z.\Phi$, where $\Phi \in \Sigma^{S\mu}_{n+1}$. Dually, to form the class $\Pi^{S\mu}_{n+1}$, take $\Sigma^{S\mu}_n \cup \Pi^{S\mu}_n$, and close under (i) boolean and modal combinators, (ii) $\nu Z.\Phi$, for $\Phi \in \Pi^{S\mu}_{n+1}$. Thus the examples above are in $\Pi^{S\mu}_1$, $\Sigma^{S\mu}_1$, and $\Pi^{S\mu}_2$ (but not $\Sigma^{S\mu}_2$) respectively. We shall say a formula is *strict* $\Sigma^{S\mu}_n$ if it is in $\Sigma^{S\mu}_n - \Pi^{S\mu}_n$.

For the modal mu-calculus, it is usual to define stronger notions of alternation [EmL86, Niw86], which capture the true interdependency of alternating fixpoints, rather than just their syntactic position. In [Bra96], the analysis is carried out for the strongest notion, that of [Niw86], giving the classes called $\Sigma^{N\mu}_n$ in [Bra96], as well as for the simple notion. In this paper, we shall not worry about the distinction, as the arguments apply whichever notion is used. Hence we shall just write $\Sigma^{\mu}_n$.

### *2.2 The arithmetic mu-calculus*

In [Lub93] Robert Lubarsky studies the logic given by adding fixpoint constructors to first-order arithmetic. Precisely, the logic ('mu-arithmetic' for short) has as basic symbols the following: function symbols $f, g, h$; predicate symbols $P, Q, R$; first-order variables $x, y, z$; set variables $X, Y, Z$; and the symbols $\vee, \wedge, \exists, \forall, \mu, \nu, \neg, \in$. As with the modal mu-calculus, $\neg$ can be pushed inwards to apply only to atomic formulae, by De Morgan duality.

The language has expressions of three kinds, individual terms, set terms, and formulae. The individual terms comprise the usual terms of first-order logic. The set terms comprise set variables and expressions $\mu(x, X).\,\phi$ and $\nu(x, X).\,\phi$, where $X$ occurs positively in $\phi$. Here $\mu$ binds both an individual variable and a set variable; henceforth we shall write just $\mu X.\,\phi$, and assume that the individual variable is the lower-case of the set variable. The formulae are built by the usual first-order construction, together with the rule that if $\tau$ is an individual term and $\Xi$ is a set term, then $\tau \in \Xi$ is a formula.

This language is interpreted over the structure $\mathbb{N}$ of first-order arithmetic with all recursive functions and predicates—in particular, let $\langle -, -\rangle$, $(-)_0$ and $(-)_1$ be standard pairing and unpairing functions. The semantics of the first-order connectives is as usual;

$\tau \in \Xi$ is interpreted naturally; and the set term $\mu X.\,\phi(x,X)$ is interpreted as the least fixpoint of the functional $\mathbf{X} \mapsto \{\, m \in \mathbb{N} \mid \phi(m,\mathbf{X}) \,\}$ (where $\mathbf{X} \subseteq \mathbb{N}$).

The simplest examples of mu-arithmetic just use least fixpoints to represent an inductive definition. For example, $\mu X.\, x = 0 \vee (x > 1 \wedge (x-2) \in X)$ is the set of even numbers. Of course, the even numbers are also the complement of the odd numbers: the odd numbers are defined by $\mu X.\, x = 1 \vee (x > 1 \wedge (x-2) \in X)$, so by negating we can express the even numbers as a maximal fixpoint $\nu X.\, x \neq 1 \wedge (x > 1 \Rightarrow (x-2) \in X)$. To produce natural examples involving alternating fixpoints is rather difficult, since even one induction is already very powerful, and most natural mathematical objects are simple.

One can define the syntactic alternation classes for arithmetic just as for the modal mu-calculus: First-order formulae are $\Sigma^\mu_0$ and $\Pi^\mu_0$, as are set variables. The $\Sigma^\mu_{n+1}$ formulae and set terms are formed from the $\Sigma^\mu_n \cup \Pi^\mu_n$ formulae and set terms by closing under (i) the first-order connectives and (ii) forming $\mu X.\,\phi$ for $\phi \in \Sigma^\mu_{n+1}$.

A crucial lemma is the following:

**Lemma 1** [Lub93,Bra96] *A $\Sigma^\mu_n$ formula of mu-arithmetic can be put into a normal form of the following shape:*

$$\tau_n \in \mu X_n.\, \tau_{n-1} \in \nu X_{n-1}.\, \tau_{n-2} \in \mu X_{n-2}.\, \ldots \tau_1 \in \mu\nu X_1.\, \phi$$

*where $\phi$ is first-order—that is, a string of alternating fixpoint quantifiers, and a first-order body.* □

(See [Bra96] for detailed definitions and proof.)

The analysis of [Lub93] provides the following

**Theorem 2** [Lub93] *The hierarchy of the sets of integers definable by $\Sigma^\mu_n$ formulae of the arithmetic mu-calculus is a strict hierarchy.* □

*2.3 Summary of [Bra96]*

The results of this paper require the results and proof techniques of [Bra96], so we now give a summary of these, skipping the fine details.

We define a *recursively presented transition system (r.p.t.s.)* to be a labelled transition system $(\mathscr{S}, \mathscr{L}, \longrightarrow)$ such that $\mathscr{S}$ is (recursively codable as) a recursive set of integers, $\mathscr{L}$ likewise, and $\longrightarrow$ is recursive. Henceforth we consider only recursively presented transition systems, with recursive valuations for the free variables. We have the following theorem, which is proved by a trivial translation of the semantics of the modal mu-calculus into mu-arithmetic:

**Theorem 3** [Bra96] *For a modal mu-calculus formula $\Phi \in \Sigma^\mu_n$, the denotation $\|\Phi\|$ in any r.p.t.s. is a $\Sigma^\mu_n$ definable set of integers.* □

We also have the converse

**Theorem 4** [Bra96] *Let $\phi(z)$ be a $\Sigma^\mu_n$ formula of mu-arithmetic. There is a r.p.t.s. $\mathscr{T}$ with recursive valuation $\mathscr{V}$ and a $\Sigma^\mu_n$ formula $\Phi$ of the modal mu-calculus such that $\phi((s)_0)$ iff $s \in \|\Phi\|^{\mathscr{T}}_{\mathscr{V}}$. (Thus if $\phi$ is not $\Sigma^\mu_{n-1}$-definable, neither is $\|\Phi\|$.)* □

This theorem is not inherently difficult; it is established by coding the evaluation of mu-arithmetic formulae into a r.p.t.s. and a modal mu-calculus formula, in such a way that arithmetic computation is handled by the transitions of the system, and the fixpoints of $\phi$ are handled by the fixpoints of $\Phi$. The proof is then a fairly straightforward induction. In this paper, we shall see a simplified version of this technique.

These two theorems establish the modal alternation hierarchy: use Theorem 4 to code an arithmetic strict $\Sigma^\mu_n$ set of integers by a strict $\Sigma^\mu_n$ modal mu-formula $\Phi$ on a r.p.t.s. $\mathscr{T}$; by Theorem 3, no $\Sigma^\mu_{n-1}$ modal formula can have the same denotation in $\mathscr{T}$, and so no $\Sigma^\mu_{n-1}$ modal formula is logically equivalent to $\Phi$.

## 3 A simple proof of the alternation hierarchy in mu-arithmetic

The first result of this paper is to observe that the alternation hierarchy theorem in mu-arithmetic can be proved simply along the lines of the proof of the strictness of the Kleene arithmetic hierarchy. The technique is to show that the truth of $\Sigma^\mu_n$ formulae can itself be expressed by a $\Sigma^\mu_n$ formula, and to use a diagonalization argument to show that this formula cannot be equivalent to any $\Pi^\mu_n$ formula.

Firstly, take a suitable Gödel numbering of mu-arithmetic. We consider only formulae without free set variables; wlog, we may assume that all encoded formulae are in normal form, and are normalized so that the free individual variables are $\mathsf{z}_0, \dots, \mathsf{z}_k$, the first-order quantifiers bind $\mathsf{z}_{k+1}, \dots$, and for a formula of alternation depth $n$, the fixpoint variables are $\mathsf{X}_n, \dots, \mathsf{X}_1$, with associated individual variables $\mathsf{x}_n, \dots, \mathsf{x}_1$. We use sans-serif type to indicate that the variable is being seen as part of an encoded object-level formula; normal italic type indicates a meta-level variable. We use corner quotes to denote the Gödel coding. We also need coded *assignments* which map an encoded variable to a value: we write $[v/\mathsf{z}]$ for the assignment that maps $\mathsf{z}$ (strictly, the code $\ulcorner \mathsf{z} \urcorner$) to the integer $v$, and $a[v/\mathsf{z}]$ for the updating of $a$ by $[v/\mathsf{z}]$. We use double quotes to indicate the appropriate meta-language formalization of the informal statement inside the quotes.

Now suppose that $\mathrm{Sat}_n(x, y)$ is a formula of mu-arithmetic expressing the truth of $\Sigma^\mu_n$ formulae, so that if $\phi$ is a formula and $a$ an assignment of values $\vec{v}$ to the free variables $\vec{\mathsf{z}}$ of $\phi$, then $\mathrm{Sat}_n(\ulcorner \phi \urcorner, a)$ is true just in case $\phi(\vec{v}/\vec{\mathsf{z}})$ is true. We have the

**Lemma 5** $\mathrm{Sat}_n(z_0, [z_0/\mathsf{z}_0])$ *is not equivalent to any* $\Pi^\mu_n$ *formula.*

**Proof.** The proof is exactly as for the arithmetical hierarchy. Suppose the contrary, i.e. that $\neg\mathrm{Sat}_n(z_0, [z_0/\mathsf{z}_0])$ is equivalent to some $\Sigma^\mu_n$ formula $\theta(z_0)$. Then we have $\theta(\ulcorner\theta\urcorner)$ iff $\neg\mathrm{Sat}_n(\ulcorner\theta\urcorner, [\ulcorner\theta\urcorner/\mathsf{z}_0])$ iff $\neg\theta(\ulcorner\theta\urcorner)$, which is a contradiction. □

It remains to show that $\mathrm{Sat}_n$ exists and is indeed a $\Sigma^\mu_n$ formula.

**Theorem 6** $\mathrm{Sat}_n$ *is a* $\Sigma^\mu_n$ *formula of mu-arithmetic, for* $n > 0$.

**Proof.** We start by constructing $\mathrm{Sat}_0$, truth in first-order arithmetic, both as a $\Sigma^\mu_1$ formula and as a $\Pi_1^\mu$ formula. $\mathrm{Sat}_0(x,y)$ is defined as:

$$\begin{aligned}\langle x,y\rangle \in \mu(w,W).\ &\text{“}(w)_0 = \ulcorner P(\tau)\urcorner \text{ and } \mathrm{pred}(\ulcorner P\urcorner, \mathrm{eval}(\ulcorner\tau\urcorner,(w)_1))\text{”}\\ &\vee \text{“}(w)_0 = \ulcorner\phi_1 \wedge \phi_2\urcorner \text{ and } (\langle\ulcorner\phi_1\urcorner,(w)_1\rangle \in W \wedge \langle\ulcorner\phi_2\urcorner,(w)_1\rangle \in W)\text{”}\\ &\vee \text{“}(w)_0 = \ulcorner\phi_1 \vee \phi_2\urcorner \text{ and } (\langle\ulcorner\phi_1\urcorner,(w)_1\rangle \in W \vee \langle\ulcorner\phi_2\urcorner,(w)_1\rangle \in W)\text{”}\\ &\vee \text{“}(w)_0 = \ulcorner\exists \mathsf{z}_i.\,\phi_1\urcorner \text{ and } \exists v.\,\langle\ulcorner\phi_1\urcorner,(w)_1[v/\mathsf{z}_i]\rangle \in W\text{”}\\ &\vee \text{“}(w)_0 = \ulcorner\forall \mathsf{z}_i.\,\phi_1\urcorner \text{ and } \forall v.\,\langle\ulcorner\phi_1\urcorner,(w)_1[v/\mathsf{z}_i]\rangle \in W\text{”}\end{aligned}$$

where $\mathrm{eval}(t,a)$ is the recursive function which evaluates a coded term $t = \ulcorner\tau\urcorner$ under the variable assignment $a$, and $\mathrm{pred}(p,x)$ is the computable predicate which is true if the value $x$ satisfies the predicate coded by $p = \ulcorner P\urcorner$.

We have here skipped the details of the coding, which are standard. For example, if we look in more detail at the clause for $\forall$, it actually says:

$$f((w)_0) = \ulcorner\forall\urcorner \wedge \forall v.\,\langle g((w)_0), h((w)_1, v, g'((w)_0))\rangle \in W$$

where $f$ extracts the top-level connective of a coded formula, $g$ extracts the body of a $\forall$ formula and $g'$ extracts the bound variable, and $h(a,v,z)$ takes the variable assignment $a$ and updates the variable whose code is $z$ by the value $v$. The fact that these functions $f,g,h$ are recursive is obvious, and since we allow ourselves all recursive functions as primitives, that is sufficient; but explicit definitions in standard arithmetic may be found in references such as [Kay91].

It is clear that this fixpoint formula simply encodes directly the definition of truth in arithmetic. The formula is $\Sigma_1^\mu$, but since the encoded recursive function terminates on all arguments—it is just a definition by induction on the structure of formulae—it does not matter whether we use a minimal or maximal fixpoint to achieve the recursion. Thus we may also obtain $\mathrm{Sat}_0$ as a $\Pi_1^\mu$ formula.

In order to encode within mu-arithmetic the evaluation of formulae with fixpoints, it is necessary to have the same fixpoint structure in the Sat formula as in the formula it's evaluating. Recall that we assume pair-normal form, and suppose that we wish to evaluate $\Sigma_n^\mu$ formulae where $n$ is odd, that is, formulae of the form

$$\tau_n \in \mu\mathsf{X}_n.\,\tau_{n-1} \in \nu\mathsf{X}_{n-1}\ldots.\,\tau_2 \in \nu\mathsf{X}_2.\,\tau_1 \in \mu\mathsf{X}_1.\,\phi \qquad (*)$$

where $\phi$ is first-order. The interpretation of the pure first-order part of $\phi$ may be done with the $\Sigma_1^\mu$ version of $\mathrm{Sat}_0$—but $\phi$ may also now contain formulae $\tau \in \mathsf{X}_i$. We cannot code as integers the sets referred to by the $\mathsf{X}_i$, so they must be represented by set variables in the meta-language. We use the meta-level variable $X_i$ to represent the object variable $\mathsf{X}_i$, and extend the body of $\mathrm{Sat}_0$ by the clauses (for each $1 \le i \ge n$)

$$\vee \text{ “}(w)_0 = \ulcorner\tau \in \mathsf{X}_i\urcorner \text{ and } \mathrm{eval}(\ulcorner\tau\urcorner,(w)_1)) \in X_i\text{”}.$$

Let $\mathrm{Sat}_0'$ denote the adjusted $\mathrm{Sat}_0$.

With these adjustments, we have that $(*)$ is true with free variable assigment $a$ just in case

$$\begin{aligned}
&\mathrm{eval}(\ulcorner\tau_n\urcorner, a) \in \mu X_n.\\
&\quad \mathrm{eval}(\ulcorner\tau_{n-1}\urcorner, a[x_n/\mathsf{x}_n]) \in \nu X_{n-1}.\ \ldots\\
&\qquad \mathrm{eval}(\ulcorner\tau_1\urcorner, a[x_n,\ldots,x_2/\mathsf{x}_n,\ldots,\mathsf{x}_2]) \in \mu X_1.\\
&\qquad\quad \mathrm{Sat}_0'(\ulcorner\phi\urcorner, a[x_n,\ldots,x_1/\mathsf{x}_n,\ldots,\mathsf{x}_1])
\end{aligned}$$

Now we just parametrize on $(*)$: let $f_1(x,y)$ be the function that given $x$ encoding a $\Sigma_n^\mu$ formula $(*)$ and an assignment $y$, computes $\mathrm{eval}(\ulcorner\tau_n\urcorner, y)$, and so on, and let $g(x)$ extract the body of $(*)$. Then we have $\mathrm{Sat}_n(x,y)$ in the form

$$\begin{aligned}
&f_n(x,y) \in \mu X_n.\, f_{n-1}(x, y[x_n/\mathsf{x}_n]) \in \nu X_{n-1}.\ \ldots\\
&\qquad f_1(x,y,[x_n,\ldots,x_2/\mathsf{x}_n,\ldots,\mathsf{x}_2]) \in \mu X_1.\, \mathrm{Sat}_0'(g(x), y[x_n,\ldots,x_1/\mathsf{x}_n,\ldots,\mathsf{x}_1])
\end{aligned}$$

which is $\Sigma_n^\mu$ as required. If $n$ is even, we use the $\Pi_1^\mu$ version of $\mathrm{Sat}_0$ instead.

The fact that $\mathrm{Sat}_n$ does indeed code truth is easily shown: show by induction on $i$ that each meta-level fixpoint set $X_i$ coincides with the object-level set $\mathsf{X}_i$. The base case follows from the correctness of $\mathrm{Sat}_0'$, and the induction step is easy.

It may be noted that we have also skipped details of what the functions $f_1$ etc. should do if given ill-formed arguments. Any convenient trick may be used; the details are unimportant. $\square$

## 4 The simple examples in the modal mu-calculus

To construct examples of strict $\Sigma_n^\mu$ formulae in the modal mu-calculus, it would suffice to apply the general construction of Theorem 4 to $\mathrm{Sat}_n$. However, $\mathrm{Sat}_n$ contains a large number of function symbols, and the translation would contain many labels. By specializing the general construction, we can eliminate most of these labels, and obtain very simple examples. The following presentation is self-contained, but terse; for a longer explanation of the technique, see [Bra96].

We aim to construct a transition system $\mathscr{T}$ and a $\Sigma_n^\mu$ modal mu-calculus formula $\Phi$ such that the set of states satisfying $\Phi$ is defined by the strict $\Sigma_n^\mu$ arithmetic formula $\mathrm{Sat}_n$.

The transition system $\mathscr{T}$ should be viewed as a machine for evaluating arithmetic expressions in the same way that $\mathrm{Sat}_n$ does: the computation happening in the body of $\mathrm{Sat}_0'$ will be dealt with by the definition of the transitions of $\mathscr{T}$, and the arithmetic fixpoints are translated into modal fixpoints in $\Phi$.

The states of $\mathscr{T}$ encode several pieces of information. Namely, a state $s$ contains: a formula $\psi_s$ of the form $(*)$, and a variable assignment $a_s$, and a pointer $p_s$ into $\psi_s$ which keeps track of where we are in the evaluation. We use the notation of $(*)$ to refer to parts of $\psi_s$.

The labels of $\mathscr{T}$ are used to distinguish various steps of computation; we shall start with enough labels to make the construction clear, and then argue the number down a little.

The transitions of $\mathscr{T}$ from a state $s$ are defined thus:

- If $p_s$ points at $\tau_i$ (or after $\mu\nu\mathsf{X}_{i+1}.$, which we consider to be the same), then $s \xrightarrow{\mathrm{x}_i} s'$ where $\psi_{s'} = \psi_s$, and $a_{s'} = a_s[\mathrm{eval}(\tau_i, a_s)/\mathsf{x}_i]$ and $p_{s'}$ points after $\mu\nu\mathsf{X}_i$. That is, the term $\tau_i$ is evaluated in the current assignment, $\mathsf{x}_i$ is set to its value, and we start evaluating the inner fixpoint.

Otherwise, $p_s$ points at a subformula of $\phi$. The transition from $s$ mimics the appropriate clause of $\mathrm{Sat}_0'$. The $\psi$ component is not altered by any transition, and the $a$ component is unchanged unless otherwise stated.

- If $p_s$ points at $P(\tau)$, then $s \xrightarrow{\mathrm{a}} s_{\mathrm{a}}$ ('a' for atom), where $s_{\mathrm{a}}$ is a special state with no structure, only if $P(\tau)$ is true with variable assignment $a_s$; otherwise there are no transitions from $s$.
- If $p_s$ points at $\phi_1 \wedge \phi_2$, then $s \xrightarrow{\mathrm{c}} s_k$ ('c' for conjunction) for $k = 1, 2$, where $p_{s_k}$ points at $\phi_k$.
- If $p_s$ points at $\forall \mathsf{z}_i.\, \phi_1$, then $s \xrightarrow{\mathrm{c}} s_k$ (universal quantification is treated as conjunction) for $k \in \mathbb{N}$, where $p_{s_k}$ points at $\phi_1$, and $a_{s_k} = a_s[k/\mathsf{z}_i]$.
- If $p_s$ points at $\phi_1 \vee \phi_2$, then $s \xrightarrow{\mathrm{d}} s_k$ ('d' for disjunction) for $k = 1, 2$, where $p_{s_k}$ points at $\phi_k$.
- If $p_s$ points at $\exists \mathsf{z}_i.\, \phi_1$, then $s \xrightarrow{\mathrm{d}} s_k$ (existential quantification is treated as disjunction) for $k \in \mathbb{N}$, where $p_{s_k}$ points at $\phi_1$, and $a_{s_k} = a_s[k/\mathsf{z}_i]$.
- If $p_s$ points at $\tau \in \mathsf{X}_i$, then $s \xrightarrow{\mathrm{x}_i} s'$, where $p_{s'}$ points after $\mu\nu\mathsf{X}_i.$, and $a_{s'} = a_s[\mathrm{eval}(\tau, a_s)/\mathsf{x}_i]$. That is, the term $\tau$ is evaluated, copied to the input variable $\mathsf{x}_i$ of the fixpoint $\mathsf{X}_i$, and evaluation of the fixpoint started.

It is clear that $\mathscr{T}$ is a recursively presented transition system.

Now consider the following *modal* mu-calculus formula:

$$\begin{aligned}\mathrm{MuSat}_n \stackrel{\mathrm{def}}{=} \langle \mathrm{x}_n \rangle \mu X_n.\, \langle \mathrm{x}_{n-1} \rangle \nu X_{n-1}.\, \ldots \langle \mathrm{x}_1 \rangle \mu X_1.\, \mu W.& \\ \langle \mathrm{a} \rangle \mathrm{tt} \;\vee\; (\langle \mathrm{c} \rangle \mathrm{tt} \wedge [\mathrm{c}] W) \;\vee\; \langle \mathrm{d} \rangle W& \\ \vee\; \langle \mathrm{x}_1 \rangle X_1 \;\vee\; \ldots \;\vee\; \langle \mathrm{x}_n \rangle X_n&\end{aligned}$$

By the construction of $\mathscr{T}$, we have:

**Theorem 7** *$s \models \mathrm{MuSat}_n$ just in case $p_s$ points at $\psi_s$, and $\mathrm{Sat}_n(\ulcorner \psi_s \urcorner, a_s)$. Hence $\mathrm{MuSat}_n$ is a strict $\Sigma^\mu_n$ modal formula.*

**Proof.** The proof is a special case of the proof of Theorem 4, the details of which are in [Bra96]. □

$\mathrm{MuSat}_n$ is already quite a simple formula, but it is interesting to try to simplify it further, which we shall do in stages.

Firstly, is it necessary to have the double occurrence of $\langle \mathrm{x}_i \rangle$, or can we remove the guards from the fixpoint formulae? Yes, we can: consider the formula

$$\mathrm{MuSat}'_n \stackrel{\mathrm{def}}{=} \mu X'_n.\, \nu X'_{n-1}.\, \ldots \mu X'_1.\, \mu W.\, \Psi$$

where $\Psi$ is formed from the body of $\mathrm{MuSat}_n$ by priming the $X_i$s. The relation between $\mathrm{MuSat}_n$ and $\mathrm{MuSat}'_n$ is that $X'_n = \ldots = X'_1 = X_1$) (note that in $\mathrm{MuSat}_n$, we have $X_1 \supseteq X_2 \cup \ldots \cup X_n$), and conversely $X_i = \langle \mathrm{x}_i \rangle X'_i$ for $i = n, \ldots, 2$. The denotation of $\mathrm{MuSat}'_n$ is still a strict $\Sigma^\mu_n$ set, since the denotation of $\mathrm{MuSat}_n$ is a projection of it.

Next, the occurrence of $\langle \mathrm{c} \rangle \mathtt{tt}$ is irritating. Its purpose is to assert that $p_s$ is indeed pointing at an $\wedge$-subterm of $\psi_s$—of course, $[\mathrm{c}]W$ is true at any state with no c-transitions from it. However, we can render it unnecessary by modifying $\mathscr{T}$: if $s$ is any state *other than* an $\wedge$-subterm state, then add a transition $s \xrightarrow{\mathrm{c}} s$. Since $W$ is a least fixpoint variable, if $W$ is true at a state with a c-loop, it is true by virtue of some other disjunct than $[\mathrm{c}]W$, and it is not true if it was not true before the loop was added.

We can also eliminate the requirement for a separate a-transition, by modifying the modification: for all those states $s$ with an a-transition, remove the c-loop added in the previous paragraph; now $[\mathrm{c}]W$ is true at those states, so we can discard the $\langle \mathrm{a} \rangle \mathtt{tt}$ clause.

Finally, we note that $W = X'_1$, and they are adjacent least fixpoints, so we can amalgamate them; further, the job of the d transition can as well be done by $\mathrm{x}_1$, since they work on disjoint sets of states.

Hence we arrive at the following very simple example of a strict $\Sigma^\mu_n$ modal formula (replacing $X'$ by $X$ again):

$$\mathrm{MuSat}''_n \stackrel{\mathrm{def}}{=} \mu X_n.\, \nu X_{n-1}.\, \ldots \mu X_1.\, [\mathrm{c}]X_1 \vee \langle \mathrm{x}_1 \rangle X_1 \vee \ldots \vee \langle \mathrm{x}_n \rangle X_n$$

## 5 Relation to parity games.

In the earliest version of this paper, we showed that if one chooses to look at models with no action labels, but with atomic propositions, the above formula appears in a form that is the same as the formula describing the existence of a winning strategy in a parity game of rank $n$, and hence that formula is strict $\Sigma^\mu_n$.

One of the STACS referees has pointed out that the strictness of the winning strategy formula can be shown directly from [Bra96] and the game interpretation of modal mu-calculus [EmJ91], without requiring the explicit use of the arithmetic formula $\mathrm{Sat}_n$. As this is an elegant proof, we outline it here, and then comment on the similarities to MuSat.

A parity game of rank $n$ [EmJ91] is played on a directed graph with the following properties: every vertex belongs either to Player or Opponent, and every vertex has an index between 1 and $n$. If the current vertex belongs to player $A$, then $A$ moves by choosing a successor vertex. (In [EmJ91], the graph is bipartite so that Player and Opponent alternate, but this is not essential.) In a given play, Player wins if either Opponent gets stuck, or if the greatest index occurring infinitely often is even. ('greatest' is sometimes replaced by 'least', e.g. in [Wal96].) For simplicity, assume henceforth that $n$ is odd.

Given such a graph, let $P$ be true at Player vertices, $O$ be true at Opponent vertices, and $R_i$ true at vertices of index $i$. It is easy to show [EmJ91,Wal96] that the modal mu-calculus formula

$$\begin{aligned}\mathrm{Par}_n \stackrel{\mathrm{def}}{=} & \ \mu X_n.\, \nu X_{n-1}.\, \ldots \mu X_1. \\ & \big(P \Rightarrow \langle\rangle \textstyle\bigwedge_{1\le i\le n}(R_i \Rightarrow X_i)\big) \wedge \big(O \Rightarrow [\,] \textstyle\bigwedge_{1\le i\le n}(R_i \Rightarrow X_i)\big)\end{aligned}$$

defines exactly the set of vertices from which Player has a winning strategy.

Now take a strict $\Sigma_n^\mu$ formula $\phi(z)$ of mu-arithmetic, and construct the r.p.t.s. $\mathscr{T}$ and modal formula $\Phi$ of Theorem 4. Given a $\Sigma_n^\mu$ modal formula, one can easily, and recursively, construct a parity game $G$ of rank $n$, whose vertices are pairs $(s, \Psi)$ of states of $\mathscr{T}$ and subformulae of $\Phi$, such that Player wins from $(s, \Psi)$ iff $s \vDash^{\mathscr{T}} \Psi$. Hence $(s, \Phi) \vDash^G \mathrm{Par}_n$ iff $\phi(s_0)$. Therefore $\|\mathrm{Par}_n\|^G$ is an arithmetic $\Sigma_n^\mu$-hard set, and so by Theorem 3 we conclude

**Theorem 8** $\mathrm{Par}_n$ *is a strict* $\Sigma_n^\mu$ *modal formula.* □

The alternative approach for showing the strictness of $\mathrm{Par}_n$ is to work from the transition system $\mathscr{T}$ of Theorem 7, and replace the action labels by atomic propositions, so that $P$ is true at disjunctive states, $O$ at conjunctive states, and $R_i$ at $\mathsf{X}_i$ states. With a little manipulation along the lines of the construction of $\mathrm{MuSat}_n''$ from $\mathrm{MuSat}_n$, one obtains exactly the formula $\mathrm{Par}_n$ as the modal encoding of $\mathrm{Sat}_n$. Thus we use $\mathrm{Sat}_n$ explicitly, and use a specialization of the proof of Theorem 4. The STACS referee's suggestion avoids this work, and so produces the $\mathrm{Par}_n$ examples from [Bra96] without using the simple proof of the mu-arithmetic hierarchy.

## 6 Conclusion

In this paper, we have made our previous proof of the modal mu-calculus alternation hierarchy entirely self-contained, and provided simple examples of formulae which separate the hierarchy, including the winning strategy formulae for parity games. It is natural to wonder whether it is possible to avoid the use of arithmetic with fixpoints altogether, and somehow carry out the entire diagonalization process within the framework of modal mu-calculus, or perhaps modal mu-calculus and parity automata. So far, such a proof has not presented itself, but we hope that it may.

Another avenue for investigation is the relationship of these techniques with Lenzi's proof [Len96] of the alternation hierarchy in the mu-calculus of deterministic trees. Lenzi's technique is entirely different, and uses topological analyses of certain structures derived from formulae and their models. It also gives examples of strict formulae; these examples are not of the form we have here, but rather look more like the formulae of [Niw86], where an alternation hierarchy for a restricted calculus on trees was established. The exact connexions between this approach and ours, and between it and automata-theoretic approaches, is far from clear.

## 7 Acknowledgments

I especially thank Alex Simpson, who suggested to me that there must be a simple proof of the mu-arithmetic hierarchy along these lines. Thanks also to Igor Walukiewicz, who pointed me at the parity game formulae. In addition to the referee who provided the improved proof for parity game formulae, other STACS referees provided helpful suggestions; I am grateful to them.

A previous version of the paper appeared as [Bra98].

I am supported by an EPSRC Advanced Research Fellowship.

## References

[Bra91] J. C. Bradfield, *Verifying Temporal Properties of Systems* (Birkhäuser, Boston, 1991).

[Bra96] J. C. Bradfield, The modal mu-calculus alternation hierarchy is strict, in: U. Montanari and V. Sassone, eds., *Proc. CONCUR '96*, LNCS **1119** (Springer, Berlin, 1996) 233–246.

[Bra98] J. C. Bradfield, Simplifying the modal mu-calculus alternation hierarchy, in: M. Morvan, C. Meinel and D. Krob, eds., *Proc. STACS 98*, LNCS **1373** (Springer, Berlin, 1998) 39–49.

[EmJ91] E. A. Emerson and C. S. Jutla, Tree automata, mu-calculus and determinacy, in: *Proc. FOCS 91.* (1991)

[EmL86] E. A. Emerson and C.-L. Lei, Efficient model checking in fragments of the propositional mu-calculus, in: *Proc. 1st LICS* (IEEE, Los Alamitos, CA, 1986) 267–278.

[Kay91] R. Kaye, *Models of Peano Arithmetic.* (Oxford University Press, Oxford, 1991).

[Koz83] D. Kozen, Results on the propositional mu-calculus, *Theoret. Comput. Sci.* **27** (1983) 333–354.

[Len96] G. Lenzi, A hierarchy theorem for the mu-calculus, in: F. Meyer auf der Heide and B. Monien, eds., *Proc. ICALP '96*, LNCS **1099** (Springer, Berlin, 1996) 87–109.

[Lub93] R. S. Lubarsky, $\mu$-definable sets of integers, *J. Symbolic Logic* **58** (1993) 291–313.

[Mos84] A. W. Mostowski, Regular expressions for infinite trees and a standard form of automata, in: A. Skowron, ed., *Fifth Symp. on Computation Theory*, LNCS **208** (Springer, Berlin, 1984) 157–168.

[Niw86] D. Niwiński, On fixed point clones, in: L. Kott, ed., *Proc. 13th ICALP*, LNCS **226** (Springer, Berlin, 1986) 464–473.

[Sti91] C. P. Stirling, Modal and temporal logics, in: S. Abramsky, D. Gabbay and T. Maibaum, eds., *Handbook of Logic in Computer Science*, Vol. 2 (Oxford University Press, 1991) 477–563.

[Wal96] I. Walukiewicz, Monadic second order logic on tree-like structures, in: C. Puech and Rüdiger Reischuk, eds., *Proc. STACS '96*, LNCS **1046** (Springer, Berlin, 1996) 401–414.