Countdown μ-Calculus

Jędrzej Kołodziejski
University of Warsaw, Poland

Bartek Klin
University of Oxford, UK

Abstract

We introduce the countdown μ-calculus, an extension of the modal μ-calculus with ordinal approximations of fixpoint operators. In addition to properties definable in the classical calculus, it can express (un)boundedness properties such as the existence of arbitrarily long sequences of specific actions. The standard correspondence with parity games and automata extends to suitably defined countdown games and automata. However, unlike in the classical setting, the scalar fragment is provably weaker than the full vectorial calculus and corresponds to automata satisfying a simple syntactic condition. We establish some facts, in particular decidability of the model checking problem and strictness of the hierarchy induced by the maximal allowed nesting of our new operators.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics; Theory of computation → Automata over infinite objects

Keywords and phrases countdown μ-calculus, games, automata

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.64

Related Version *Full Version*: https://arxiv.org/abs/2208.00536

Funding *Jędrzej Kołodziejski*: National Science Center (NCN) grant 2021/41/B/ST6/00535

Acknowledgements We are grateful to Mikołaj Bojańczyk for numerous helpful suggestions.

1 Introduction

The modal μ-calculus [14] is a well-known logic for defining and verifying behavioural properties of state-and-transition systems. It extends propositional logic with basic next-step modalities and fixpoint operators to describe long-term behaviour. It is expressive enough to include other temporal logics such as CTL* as fragments, but it has good computational properties, and its simple syntax and semantics makes it a convenient formalism to study.

The μ-calculus has a straightforward inductively-defined semantics, but it is often useful to consider an alternative (but equivalent) semantics based on parity games. A formula φ together with a model $\mathcal{M}$ define a game between two players called ∀dam and ∃ve. Positions in the game are of the form (m, ψ) where m is a point in $\mathcal{M}$ and ψ is a subformula of φ, and moves are defined so that ∃ve has a winning strategy from (m, φ) if and only if φ holds in m. Among other advantages, the game-based semantics provides more efficient algorithms for model checking of μ-calculus formulas than an inductive computation of fixpoints [9].

The model component can be abstracted away from parity games. Indeed, a formula φ itself gives rise to an alternating parity automaton $\mathcal{A}_\varphi$ that recognizes models. The behaviour of an automaton on a model is defined in terms of a parity game, states of $\mathcal{A}_\varphi$ are subformulas of φ, and the transition relation is defined so that it accepts a model $\mathcal{M}$ rooted in a point m if and only if φ holds in m. The advantage of this is that $\mathcal{A}_\varphi$, while conceptually closer to a parity game, is a finite structure even if it is then applied to infinite models.

The modal μ-calculus is a rather expressive formalism: it can define all bisimulation-invariant properties definable in monadic second-order logic (MSO) [13], such as "there is an infinite path of τ-labeled edges". However, there are some properties of interest which are not definable even in MSO. Notable examples include (un)boundedness properties such as "for every number n, there is a path with at least n consecutive τ-labeled edges". An extension of

47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022).
Editors: Stefan Szeider, Robert Ganian, and Alexandra Silva; Article No. 64; pp. 64:1–64:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

MSO called MSO+U, aimed at defining such properties, has been considered [6]. However, the satisfiability problem of MSO+U turned out to be undecidable even for word models [4]. Since the modal μ-calculus is a fragment of MSO, it is worthwhile to extend it with a mechanism for defining (un)boundedness properties, in the hope of retaining decidability.

In this paper we propose such an extension: the *countdown μ-calculus* μ^α-ML. In addition to μ-calculus operators, it features countdown operators μ^α and ν^α parametrized by ordinal numbers α. Instead of least and greatest fixpoints, they define ordinal approximations of those fixpoints. Intuitively, while the meaning of classical μ-calculus formulas $\mu x.\varphi(x)$ and $\nu x.\varphi(x)$ is defined by infinite unfolding of the formula φ until a fixpoint is reached, for $\mu^\alpha x.\varphi(x)$ and $\nu^\alpha x.\varphi(x)$ the unfolding stops after α steps (which makes a difference if α is smaller than the *closure ordinal* of φ). The classical fixpoint operators are kept but renamed to μ^∞ and ν^∞, to make clear the lack of any restrictions on the unfolding process.

An inductive definition of the semantics of countdown formulas is just as straightforward as in the classical case. With some more effort, we are able to formulate game-based semantics as well. We introduce *countdown games* and *countdown automata*, which are similar to parity games and alternating automata known from the classical setting, but are additionally equipped with counters that are decremented and reset by the two players according to specific rules. Intuitively, the counters say how many more times various ranks can be visited, in similar manner to the signatures introduced by Walukiewicz [17, Section 3]. A player responsible for decrementing a counter may lose the game if the value of that counter is zero, just as a player responsible for finding the next position in a game may lose if there is no position to go to. The key mechanism of countdown games is implicit in [11], where the authors investigate a nonstandard semantics for the scalar fragment of the μ-calculus equivalent to replacing every μ and ν by our countdown operators μ^α and ν^α, respectively. However, the authors do not abstract from formulas in their definition of games, nor consider the full vectorial calculus that corresponds to automata.

A correspondence between countdown formulas, automata and games is as tight as for the classical μ-calculus. However, complications arise: the distinction between *vectorial* and *scalar* formulas, which in the classical case disappears to a large extent due to the so-called Bekić principle, now becomes pronounced. We prove that vectorial countdown calculus is more expressive than its scalar fragment. We also prove that the countdown operator nesting hierarchy of formulas is proper.

We conjecture that the satisfiability problem is decidable for μ^α-ML. Unfortunately, the lack of positional determinacy in countdown games prevents us from using proof techniques known from parity automata (where one can transform an alternating automaton into a nondeterministic one that guesses the positional strategy). Nevertheless, the existence of an automata model equivalent to logic is encouraging. Apart from allowing us to solve some fragments of the logic, it implies that μ^α-ML does not share some of the troublesome properties of MSO + U that result in undecidability. In particular, it can be used to show that all languages definable in μ-ML have *bounded topological complexity* (i.e. at most Σ^1_2, see [15] for an introduction to topological methods in computer science). Since MSO + U defines a Σ^1_n-complete language for every $n < \omega$ [12, Theorem 2.1], [15, Theorem 7], it follows that some MSO + U-definable languages are not expressible in μ^α-ML (whether μ^α-ML-definability implies MSO + U-definability remains an open question). Since by [8, Theorem 1.3] every logic closed under boolean combinations, projections and defining the language U from Example 4 contains MSO + U, this means that our calculus is *not closed under projections*. This is an arguably good news, as in the light of [3, Theorem 1.4], giving up closure under projections is the only way to go if one wants to design a decidable extension of MSO closed under boolean operations. Decidability of the weak variant $\mathsf{WMSO}+\mathsf{U}$ of $\mathsf{MSO}+\mathsf{U}$ over infinite words [2] and infinite (ranked) trees [5] shows that such extensions are possible. In fact, both results are obtained by establishing a correspondence with equivalent automata models, namely deterministic max-automata [2, Theorem 1] and nested limsup automata [5, Theorem 2]. Since the existence of accepting runs for such automata can be expressed in μ^α-ML, we get that μ^α-ML contains $\mathsf{WMSO}+\mathsf{U}$ on infinite words and trees. The opposite inclusion is false (due to topological reasons), at least for the trees. The relation between μ^α-ML and the ωB-, ωS- and ωBS-automata of [7] remains unclear, as these models do not admit determinization. Also, the relation between our logic and regular cost functions (see e.g. [10]) is less immediate than it could seem at first glance and requires further research.

2 Preliminaries

Fixpoints. Let Ord be the class of all ordinals, and Ord_∞ the class Ord extended with an additional element ∞ greater than all ordinals.

Knaster-Tarski theorem says that every monotonic function $F : A \to A$ on a complete lattice A has the least and the greatest fixpoint, which we denote F^∞_μ and F^∞_ν. Moreover:

- F^∞_μ is the limit of the increasing sequence $F^\alpha_\mu = \bigvee_{\beta<\alpha} F(F^\beta_\mu)$
- F^∞_ν is the limit of the decreasing sequence $F^\alpha_\nu = \bigwedge_{\beta<\alpha} F(F^\beta_\nu)$

where $\alpha \in \mathsf{Ord}$ and $\bigvee, \bigwedge$ are the join and meet operations in A.

Parity games. A *parity game* is played between two players $\exists$ve and $\forall$dam (or simply $\exists$ and $\forall$). It consists of a set of *positions* $V = V_\exists \sqcup V_\forall$ divided between both players, an edge relation $E \subseteq V \times V$, and a labeling $\mathsf{rank} : V \to \mathcal{R}$ for some finite linear order $\mathcal{R} = \mathcal{R}_\exists \sqcup \mathcal{R}_\forall$ divided between the two players.

A *play* is a sequence of positions. After a play $\pi = v_1 \dots v_n \in V^*$, the owner of v_n chooses $(v_n, v_{n+1}) \in E$ and the game moves to v_{n+1}. A player who has no legal moves loses immediately. To determine the winner of an infinite play, we look at the highest $r \in \mathcal{R}$ such that positions with rank r appear infinitely often in the play, and the owner of r loses.

A *strategy* for a player $P \in \{\exists, \forall\}$ is a partial map $\sigma : V^*V_P \to E$ that tells the player how to move. A play $v_1v_2\dots$ is *consistent* with σ if for every n such that $v_n \in V_P$ we have $\sigma(v_1 \dots v_n) = v_{n+1}$. A strategy σ is *winning* from a position v if every play that begins in v and is consistent with σ is a win for P. A strategy is *positional* if $\sigma(\pi)$ depends only on the last position in π. Parity games are *positionally determined*: if a player has a winning strategy from v then (s)he has a winning positional strategy.

Modal μ-calculus. A model $\mathcal{M}$ for a fixed set Act of atomic *actions* consists of a set of *points* $M \ni \mathsf{m}, \mathsf{n}, \cdots$ together with a binary relation $\xrightarrow{\tau} \subseteq M \times M$ for every $\tau \in \mathsf{Act}$.

Formulas of the modal μ-calculus μ-ML are given by the grammar:

$$\varphi ::= x \mid \top \mid \bot \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mu x.\varphi \mid \nu x.\varphi \mid \langle\tau\rangle\varphi \mid [\tau]\varphi \tag{1}$$

where x ranges over a fixed infinite set Var of variables and $\tau \in \mathsf{Act}$. Given a valuation $\mathsf{val} : \mathsf{Var} \to \mathcal{P}(M)$, the semantics $[\![\varphi]\!]^{\mathsf{val}} \subseteq M$ for all formulas φ is defined inductively, with $\mu x.\varphi$ and $\nu x.\varphi$ denoting the least and greatest fixpoints, respectively, of the monotonic function $H \mapsto [\![\varphi]\!]^{\mathsf{val}[x\mapsto H]}$ on the complete lattice $\mathcal{P}(M)$. More details can be found e.g. in [1, 16], but they can also be discerned from Section 3 below, where the semantics of countdown μ-calculus is presented in detail.

The above syntax does not include negation, but μ-calculus formulas are semantically closed under negation. For every formula φ there is a formula $\widetilde{\varphi}$ that acts as the negation of φ on every model, defined by induction in a straightforward way:

$$\widetilde{\varphi_1 \vee \varphi_2} = \widetilde{\varphi_1} \wedge \widetilde{\varphi_2}, \qquad \widetilde{\langle\tau\rangle\varphi} = [\tau]\widetilde{\varphi}, \qquad \widetilde{\mu x.\varphi} = \nu x.\widetilde{\varphi}, \qquad \text{etc.} \tag{2}$$

Vectorial μ-calculus. A syntactically richer version of the modal μ-calculus admits mutual fixpoint definitions of multiple properties, in formulas such as $\mu_1(x_1, x_2).(\varphi_1, \varphi_2)$, where variables x_1 and x_2 may occur both in φ_1 and φ_2. Given a valuation val as before, this formula is interpreted as the least fixpoint of the monotonic function $(H_1, H_2) \mapsto (\llbracket\varphi_1\rrbracket^{\mathsf{val}[x_i \mapsto H_i]}, \llbracket\varphi_1\rrbracket^{\mathsf{val}[x_i \mapsto H_i]})$ on the complete lattice $\mathcal{P}(M)^2$; the resulting pair of sets is then projected to the first component as dictated by the subscript in μ_1. Tuples of any size are allowed. This *vectorial* calculus is expressively equivalent to the scalar version described before, thanks to the so-called *Bekić principle* which says that the equality:

$$\mu\begin{pmatrix}x_1\\x_2\end{pmatrix}.\begin{pmatrix}f_1(x_1,x_2)\\f_2(x_1,x_2)\end{pmatrix} = \begin{pmatrix}\mu x_1.f_1(x_1,\ \mu x_2.f_2(x_1,x_2))\\ \mu x_2.f_2(\mu x_1.f_1(x_1,x_2),\ x_2)\end{pmatrix} \tag{3}$$

holds for every pair of monotone operations $f_i : A_1 \times A_2 \to A_i$ on complete lattices A_1, A_2, and similarly for the greatest fixpoint operator ν in place of μ.

3 Countdown μ-calculus

We now introduce the *countdown μ-calculus* μ^α-ML. We begin with the scalar version.

3.1 The scalar fragment

As before, fix an infinite set Var of variables and a set Act of actions. The syntax of *(scalar) countdown μ-calculus* is defined as follows:

$$\varphi ::= x \mid \top \mid \bot \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mu^\alpha x.\varphi \mid \nu^\alpha x.\varphi \mid \langle\tau\rangle\varphi \mid [\tau]\varphi \tag{4}$$

for $x \in \mathsf{Var}$, $\tau \in \mathsf{Act}$ and $\alpha \in \mathsf{Ord}_\infty$; the presence of ordinal numbers α is the only syntactic difference with (1). A formula with no free variables is called a *sentence*. In case $|\mathsf{Act}| = 1$, we may skip the labels and write $\Diamond$ and $\Box$ instead of $\langle\tau\rangle$ and $[\tau]$. In statements that apply both to least and greatest fixpoints, we will sometimes use η^α to denote either μ^α or ν^α.

Given a model $\mathcal{M}$, for every valuation $\mathsf{val} : \mathsf{Var} \to \mathcal{P}(M)$, the *semantics* $\llbracket\varphi\rrbracket^{\mathsf{val}} \subseteq M$ is defined inductively as follows:

$$\begin{aligned}
\llbracket x\rrbracket^{\mathsf{val}} &= \mathsf{val}(x);\\
\llbracket\top\rrbracket^{\mathsf{val}} = M \quad &\text{and} \quad \llbracket\bot\rrbracket^{\mathsf{val}} = \emptyset\\
\llbracket\varphi_1 \vee \varphi_2\rrbracket^{\mathsf{val}} = \llbracket\varphi_1\rrbracket^{\mathsf{val}} \cup \llbracket\varphi_2\rrbracket^{\mathsf{val}} \quad &\text{and} \quad \llbracket\varphi_1 \wedge \varphi_2\rrbracket^{\mathsf{val}} = \llbracket\varphi_1\rrbracket^{\mathsf{val}} \cap \llbracket\varphi_2\rrbracket^{\mathsf{val}};\\
\llbracket\langle\tau\rangle\varphi\rrbracket^{\mathsf{val}} = \{\mathsf{m} \in M \mid \exists_{\mathsf{n}\in\llbracket\varphi\rrbracket^{\mathsf{val}}}\ \mathsf{m} \xrightarrow{\tau} \mathsf{n}\} \quad &\text{and} \quad \llbracket[\tau]\varphi\rrbracket^{\mathsf{val}} = \{\mathsf{m} \in M \mid \forall_{\mathsf{n}\in\llbracket\varphi\rrbracket^{\mathsf{val}}}\ \mathsf{m} \xrightarrow{\tau} \mathsf{n}\};\\
\llbracket\mu^\alpha x.\varphi\rrbracket^{\mathsf{val}} = F^\alpha_\mu \quad &\text{and} \quad \llbracket\nu^\alpha x.\varphi\rrbracket^{\mathsf{val}} = F^\alpha_\nu
\end{aligned}$$

where in the last clause $F(H) = \llbracket\varphi\rrbracket^{\mathsf{val}[x \mapsto H]}$. We will skip the index val if it is immaterial or clear from the context.

This obviously contains the classical μ-calculus, but is capable of capturing *boundedness* and *unboundedness* properties which are not expressible in the classical setting:

▶ **Example 1.** For $|\mathsf{Act}| = 1$, consider the formula $\nu^\alpha x.\Diamond x$. In a model $\mathcal{M}$, for $\alpha < \omega$ the set $\llbracket\nu^\alpha x.\Diamond x\rrbracket$ consists of the points from which there is a path of length at least α. Hence, $\nu^\omega x.\Diamond x$ holds in a point if there are arbitrarily long finite paths starting from there.

3.2 The vectorial calculus

The (full) *countdown μ-calculus* is defined as for its scalar fragment, except that fixpoint operators act on tuples (vectors) of formulas rather than on single formulas.

▶ **Definition 2.** *The syntax of* countdown μ-calculus *is given as follows:*

$$\varphi ::= x \mid \top \mid \bot \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \mu_i^\alpha \overline{x}.\overline{\varphi} \mid \nu_i^\alpha \overline{x}.\overline{\varphi} \mid \langle\tau\rangle\varphi \mid [\tau]\varphi$$

for $1 \le i \le n < \omega$, $\overline{x} = \langle x_1, ..., x_n\rangle \in \mathsf{Var}^n$, $\overline{\varphi} = \langle \varphi_1, ..., \varphi_n\rangle$ a tuple of formulas, $\tau \in \mathsf{Act}$ and $\alpha \in \mathsf{Ord}_\infty$.

▶ **Definition 3.** *The meaning $[\![\varphi]\!]^{\mathsf{val}} \subseteq M$ of a formula φ in a model $\mathcal{M}$ under valuation* val *is defined by induction the same way as for the scalar formulas except for the operators μ_i^α and ν_i^α, in which case:*

$$[\![\mu_i^\alpha \overline{x}.\overline{\varphi}]\!]^{\mathsf{val}} = \pi_i(F_\mu^\alpha) \quad \text{and} \quad [\![\nu_i^\alpha \overline{x}.\overline{\varphi}]\!]^{\mathsf{val}} = \pi_i(F_\nu^\alpha)$$

where the monotone map $F : (\mathcal{P}(M))^n \to (\mathcal{P}(M))^n$ is given as:

$$F(H_1, ..., H_n) = ([\![\varphi_1]\!]^{\mathsf{val}'}, ..., [\![\varphi_n]\!]^{\mathsf{val}'})$$

for $\mathsf{val}' = \mathsf{val}[x_1 \mapsto H_1, ..., x_n \mapsto H_n]$ and $\pi_i : (\mathcal{P}(M))^n \to \mathcal{P}(M)$ is the i-th projection.

Note that operators μ^∞ and ν^∞ are equivalent to μ and ν from the classical μ-calculus. Furthermore, for every ordinal α, the formula $\mu_i^{\alpha+1}\overline{x}.\overline{\psi}$ is equivalent to

$$\psi_i[x_1 \mapsto \mu_1^\alpha \overline{x}.\overline{\psi}, \dots, x_n \mapsto \mu_n^\alpha \overline{x}.\overline{\psi}]$$

and similarly for $\nu^{\alpha+1}$. As a result, without loss of generality we may assume that in countdown operators μ^α and ν^α only limit ordinals α are used.

The countdown μ-calculus is semantically closed under negation in the same way as the classical calculus, extending (2) with the straightforward $\widetilde{\mu_i^\alpha \overline{x}.\overline{\varphi}} = \nu_i^\alpha \overline{x}.\overline{\widetilde{\varphi}}$ and $\widetilde{\nu_i^\alpha \overline{x}.\overline{\varphi}} = \mu_i^\alpha \overline{x}.\overline{\widetilde{\varphi}}$.

In Section 6 we will compare the expressive power of the vectorial and scalar countdown μ-calculus in detail. For now, let us show that Bekić principle (3) fails for countdown operators:

▶ **Example 4.** An infinite word $W \in \Gamma^\omega$ over the alphabet $\Gamma = \{\mathsf{a}, \mathsf{b}\}$ can be seen as a model for $\mathsf{Act} = \Gamma$ with ω as the set of points and with transition relations defined by:

$$n \xrightarrow{\tau} m \iff m = n + 1 \text{ and } W_n = \tau.$$

For every regular language $K \subseteq \Gamma^*$ and $x \in \mathsf{Var}$, it is straightforward to define a fixpoint formula (in the classical μ-calculus, so without countdown operators) $\langle K\rangle x$ that holds in a point n, for a valuation val, if and only if there exists a word $w \in K$ and a path in W labelled with w that starts in n and ends in a point that belongs to $\mathsf{val}(x)$. Then, the formula:

$$\varphi = \nu_1^\omega(x_1, x_2).(\langle\Gamma^*\rangle x_2, \langle\mathsf{a}\rangle x_2)$$

is true in a word W iff it contains arbitrarily long blocks of consecutive a's. To see this, observe that at the i-th step of approximation: (i) the second component (x_2) contains a point n iff the next i transitions are all labelled with a, and (ii) the first component (x_1) contains a point n iff the second component contains at least one point after n.

However, the following scalar formula constructed by analogy to the Bekić principle:

$$\psi = \nu^\omega x_1.\langle\Gamma^*\rangle(\nu^\omega x_2.\langle\mathsf{a}\rangle x_2)$$

is equivalent to $\langle\Gamma^*\rangle(\nu^\omega x_2.\langle\mathsf{a}\rangle x_2)$, and the formula under $\langle\Gamma^*\rangle$ holds in a point iff all the future transitions from that point are labelled with a. Thus, ψ holds (in any point) iff the word W is of the form $\Gamma^*\mathsf{a}^\omega$, and so ψ is not equivalent to φ.

4 Countdown Games

The notion of a countdown game extends that of a parity game. As for parity games, it assumes a fixed finite linear order of ranks $\mathcal{R} = \mathcal{R}_\exists \sqcup \mathcal{R}_\forall$. In addition, we fix a subset $\mathcal{D} \subseteq \mathcal{R}$ of *nonstandard* ranks; at positions with these ranks countdowns will occur. Denote $\mathcal{D}_\exists = \mathcal{D} \cap \mathcal{R}_\exists$ and $\mathcal{D}_\forall = \mathcal{D} \cap \mathcal{R}_\forall$.

A *countdown game* consists of a set of *positions* $V = V_\exists \sqcup V_\forall$ divided between players $\exists$ve and $\forall$dam, an edge relation $E \subseteq V \times V$, a labelling $\mathsf{rank} : V \to \mathcal{R}$, and an initial counter valuation $\mathsf{ctr}_I : \mathcal{D} \to \mathsf{Ord}$. Each nonstandard rank has an associated counter.

Each game *configuration* consists of a position $v \in V$ together with a counter valuation $\mathsf{ctr} : \mathcal{D} \to \mathsf{Ord}$. We consider *positional* and *countdown* configurations, denoted respectively $\langle v, \mathsf{ctr}\rangle$ and $[v, \mathsf{ctr}]$, with the following moves allowed:

- From a positional configuration $\langle v, \mathsf{ctr}\rangle$, the owner of v chooses an edge $(v, w) \in E$ and the game proceeds from the countdown configuration $[w, \mathsf{ctr}]$;
- From a countdown configuration $[v, \mathsf{ctr}]$, the owner of $r = \mathsf{rank}(v)$ chooses a counter valuation ctr' such that:
  - $\mathsf{ctr}'(r') = \mathsf{ctr}_I(r')$ for $r' < r$,
  - $\mathsf{ctr}'(r) < \mathsf{ctr}(r)$ (if r is nonstandard),
  - $\mathsf{ctr}'(r') = \mathsf{ctr}(r')$ for $r' > r$,

  and the game proceeds from the positional configuration $\langle v, \mathsf{ctr}'\rangle$. In words: counters for ranks lower than r are reset, the counter for r (if any) is decremented, and counters for higher ranks are left unchanged. Note that if r is standard then there is no real choice here: ctr' is determined by ctr. And if r is nonstandard then the move amounts to choosing an ordinal $\alpha < \mathsf{ctr}(r)$.

Every play of the game alternates between positional and countdown configurations, and in each move only one component of the configuration is modified. Therefore, although a play is formally a sequence of configurations, it can be more succinctly represented as an alternating sequence of positions and counter valuations:

$$\pi = v_1 \mathsf{ctr}_2 v_2 \mathsf{ctr}_2 v_3 \mathsf{ctr}_3 \cdots \tag{5}$$

This has the same length as the sequence of configurations, and we will call it the length of the play. A *phase* of a game is a set of its finite plays that is convex with respect to the prefix ordering.

In any configuration, if the player responsible for making the next move is stuck, (s)he looses immediately. Otherwise, in an infinite play, the owner of the greatest rank appearing infinitely often looses, as in parity games. Strategies and winning strategies are defined as for classical parity games, as partial functions from finite plays to moves.

Given configuration γ, we denote the game *initialized in the configuration* γ by $\mathcal{G}, \gamma$. The default initial counter assignment is ctr_I and the default initial mode is the positional one, meaning that $\mathcal{G}, v$ stands for $\mathcal{G}, \langle v, \mathsf{ctr}_I\rangle$.

Note that the only way the counters may interfere with a play is when a counter has value 0 and so its owner cannot decrement it. It is therefore beneficial for a player to have greater ordinals at his/her counters.

Countdown games are not positionally determined, in the sense that the players may need to look at the counter values in order to choose a winning move (although they are *configurationally determined*, since a countdown game $\mathcal{G}$ can be seen as a parity game with configurations of $\mathcal{G}$ as its positions).

5 Countdown Automata

Countdown automata are a stepping stone between formulas and games. A countdown formula will define an automaton, which will then recognize a model in terms of a countdown game. Since formulas can have free variables, for technical reasons we will also consider automata with free variables. These variables resemble terminal states in that they can be targets of transitions, but no transitions originate in them, and whether they accept or not depends on an external valuation.

▶ **Definition 5.** *A* countdown automaton *consists of:*

- *a finite set of states* $Q = Q_\exists \sqcup Q_\forall$ *divided between two players;*
- *an initial state* $q_I \in Q$*;*
- *a transition function* $\delta : Q \to \mathcal{P}(Q \sqcup \mathsf{Var}) \sqcup (\mathsf{Act} \times (Q \sqcup \mathsf{Var}))$ *(we call the left part* ϵ-transitions *and the right one* modal transitions*);*
- *an assignment of ranks* $\mathsf{rank} : Q \to \mathcal{R}$ *and an assignment of initial counter values* $\mathsf{ctr}_I : \mathcal{D} \to \mathsf{Ord}$*, as in a countdown game.*

The language of an automaton is defined in terms of a countdown game, analogously to parity games and parity automata.

▶ **Definition 6.** *Fix an automaton* $\mathcal{A} = (Q, q_I, \delta, \mathsf{rank}, \mathsf{ctr}_I)$*. Given a model* $\mathcal{M}$*, a valuation* $\mathsf{val} : \mathsf{Var} \to \mathcal{P}(M)$ *and a point* $\mathsf{m}_I \in M$*, we define the semantic game* $\mathcal{G}^{\mathsf{val}}(\mathcal{A})$ *to be the countdown game* $(V, E, \mathsf{rank}', \mathsf{ctr}_I)$ *where positions are of the form* $V = M \times (Q \sqcup \mathsf{Var})$ *and the edge relation* E *is defined as follows. In a position* (m, q) *for* $q \in Q$*:*

- *if* $\delta(q) \subseteq Q \sqcup \mathsf{Var}$*, outgoing edges (called* ϵ*-edges, or* ϵ*-moves) are* $\{((\mathsf{m}, q), (\mathsf{m}, z)) \mid z \in \delta(q)\}$*,*
- *if* $\delta(q) = (\tau, p)$*, outgoing edges (modal edges, modal moves) are* $\{((\mathsf{m}, q), (\mathsf{n}, p)) \mid \mathsf{m} \xrightarrow{\tau} \mathsf{n}\}$*.*

There are no outgoing edges from positions (m, x) *for* $x \in \mathsf{Var}$*.*

For $q \in Q$*, the owner of the position* (m, q) *is the owner of the state* q*, and* $\mathsf{rank}'(\mathsf{m}, q) = \mathsf{rank}(q)$*. For* $x \in \mathsf{Var}$*, the position* (m, x) *belongs to* ∀*dam if* $\mathsf{m} \in \mathsf{val}(x)$ *and to* ∃*ve otherwise. The rank* $\mathsf{rank}'(\mathsf{m}, x)$ *can be set arbitrarily, as it does not affect the outcome of the game. The initial counter assignment* ctr_I *is kept the same.*

The language $[\![\mathcal{A}]\!]^{\mathsf{val}} \subseteq M$ *of an automaton* $\mathcal{A}$ *is the set of all points* $\mathsf{m} \in M$ *for which the configuration* $\langle(\mathsf{m}, q_I), \mathsf{ctr}_I\rangle$ *in the game* $\mathcal{G}^{\mathsf{val}}(\mathcal{A})$ *is winning for* ∃*ve.*

It is worth to mention that although in general countdown games are not positional, one can show a much weaker but still useful fact: in the particular case of semantic games, the winning player always has a strategy that does not look at the counters in the initial *pre-modal* phase of the game (that is, *before the first modal move*).

The countdown calculus and countdown automata have the same expressive power, i.e. there are language-preserving translations $\varphi \mapsto \mathcal{A}_\varphi$ and $\mathcal{A} \mapsto \varphi_\mathcal{A}$ between formulas and automata. As in the classical setting, the link between formulas and automata is very useful in establishing facts about the logic. For example, one can use game semantics to show that every formula of the standard μ-ML can be transformed into an equivalent guarded one. Thanks to the equivalence between *countdown* formulas and *countdown* automata, the same is true for μ^α-ML.

We will now explain the translations between logic and automata in turn.

5.1 From formulas to automata – Game Semantics

Every countdown formula $\varphi \in \mu^\alpha$-ML gives rise to a countdown automaton $\mathcal{A}_\varphi$ such that $[\![\varphi]\!]^{\mathsf{val}} = [\![\mathcal{A}_\varphi]\!]^{\mathsf{val}}$ for every model $\mathcal{M}$ and valuation val. Specifically, given a formula φ (with some free variables), we define an automaton $\mathcal{A}_\varphi = (Q, q_I, \delta, \mathsf{rank}, \mathsf{ctr}_I)$ (over the same free variables) as follows:

- $Q = \mathsf{SubFor}(\varphi) - \mathsf{FreeVar}(\varphi)$ is the set of all subformulas other than the free variables of φ (*without* identifying different occurrences of identical subformulas, i.e., here a subformula means a path in the syntactic tree of φ from the root of φ to the root node of the subformula). Ownership of a state in Q depends on the topmost connective, with $\exists$ve owning $\vee$ and $\langle\tau\rangle$ and $\forall$dam owning $\wedge$ and $[\tau]$; ownership of fixpoint subformulas, countdown subformulas and variables can be set arbitrarily as it will not matter;
- $q_I = \varphi$;
- the transition function is defined by cases:
  - $\delta(\theta_1 \vee \theta_2) = \delta(\theta_1 \wedge \theta_2) = \{\theta_1, \theta_2\}$,
  - $\delta(\langle\tau\rangle\theta) = \delta([\tau]\theta) = (\tau, \theta)$,
  - $\delta(\eta_i^\alpha \overline{x}.\overline{\theta}) = \{\theta_i\}$ (for $\eta = \mu$ or $\eta = \nu$),
  - $\delta(x) = \{\theta_i\}$, where $\eta_j^\alpha(x_1, ..., x_n).(\theta_1, ..., \theta_n)$ is the (unique) subformula of φ binding x with $x = x_i$.
- For the ranking function, assume that the lowest rank in $\mathcal{R}$ is standard and call it 0 (ownership of this rank does not matter). Then let rank assign 0 to all subformulas of φ except for immediate subformulas of fixpoint operators. To those, assign ranks in such a way that subformulas have strictly smaller ranks than their superformulas, and for every subformula $\eta_i^\alpha \overline{x}.\overline{\varphi}$:
  - all formulas in the tuple $\overline{\varphi}$ have the same rank r,
  - r belongs to $\exists$ve if $\eta = \mu$ and to $\forall$dam if $\eta = \nu$, and
  - if $\alpha = \infty$ then r is standard, otherwise it is nonstandard and $\mathsf{ctr}_I(r) = \alpha$.

We denote $\mathcal{G}^{\mathsf{val}}(\varphi) = \mathcal{G}^{\mathsf{val}}(\mathcal{A}_\varphi)$.

▶ **Theorem 7** (Adequacy). *For every model $\mathcal{M}$ and valuation* val, $[\![\varphi]\!]^{\mathsf{val}} = [\![\mathcal{A}_\varphi]\!]^{\mathsf{val}}$.

Proof (sketch). As with the classical *mu*-calculus, the proof proceeds by induction on the complexity of the formula. The only new cases of $\mu^\alpha \overline{x}.\overline{\varphi}$ and $\nu^\alpha \overline{x}.\overline{\varphi}$ are proven by transfinite induction on α. ◀

▶ **Example 8.** For $\mathsf{Act} = \{\tau\}$, consider the formula $\varphi = \nu^\omega x.\Diamond x$ from Example 1. The automaton $\mathcal{A}_\varphi$ has three states: $Q = \{\varphi, \Diamond x, x\}$, with φ the initial state, and the transition function comprises two deterministic ϵ-transitions and one modal transition:

$$\delta(\varphi) = \{\Diamond x\}, \qquad \delta(\Diamond x) = (\tau, x), \qquad \delta(x) = \{\Diamond x\}.$$

The state $\Diamond x$ is owned by $\exists$ve; ownership of the other two states does not matter. The automaton uses two ranks, $0 < 1$, where 0 is standard and 1 is nonstandard, assigned to states by: $\mathsf{rank}(\varphi) = \mathsf{rank}(x) = 0$ and $\mathsf{rank}(\Diamond x) = 1$. Rank 1 is owned by $\forall$dam; ownership of rank 0 does not matter. (Note how the state $\Diamond x$ is owned by $\exists$ve, but its rank is owned by $\forall$dam). The initial counter value is $\mathsf{ctr}_I(1) = \omega$.

Now consider any model $\mathcal{M}$. Since Act has only one element, $\mathcal{M}$ is simply a directed graph. The semantic game $\mathcal{G}(\varphi)$ on $\mathcal{M}$ (φ has no free variables, so neither has $\mathcal{A}_\varphi$ and we need not consider valuations val) has positions of the form (m, q) where $\mathsf{m} \in M$ and $q \in Q$, with ownership and rank inherited from q. Edges are of the form:

- $((\mathsf{m}, \varphi), (\mathsf{m}, \Diamond x))$ and $((\mathsf{m}, x), (\mathsf{m}, \varphi))$ – the ϵ-edges,
- $((\mathsf{m}, \Diamond x), (\mathsf{n}, x))$ such that $\mathsf{m} \to \mathsf{n}$ is an edge in $\mathcal{M}$ – the modal edges.

Configurations of the game arise from positions together with counter valuations; there is only one nonstandard rank, so a counter valuation is simply an ordinal.

For a point $\mathsf{m} \in \mathcal{M}$, the default initial configuration of the game is the positional configuration $\langle(\mathsf{m}, \varphi), \omega\rangle$. A play that begins in this configuration proceeds as follows:

1. The first move is deterministic, to the countdown configuration $[(\mathsf{m}, \Diamond x), \omega]$.
2. ∀dam, as the owner of the rank of $\Diamond x$, makes the next move: he chooses a number $k < \omega$, and the games moves to the positional configuration $\langle(\mathsf{m}, \Diamond x), k\rangle$.
3. ∃ve owns the position, so she makes the next move: she chooses a point $\mathsf{n} \in M$ such that $\mathsf{m} \xrightarrow{\tau} \mathsf{n}$, and the game moves to the countdown configuration $[(\mathsf{n}, x), k]$.
4. The rank of x is standard, so in the next move the counter does not change and the game moves to $\langle(\mathsf{n}, x), k\rangle$. The next move is also deterministic, to the countdown configuration $[(\mathsf{n}, \Diamond x), k]$. The game then goes back to step 2. above, with k in place of ω.

From this it is clear that ∃ve wins from $\langle(\mathsf{m}, \varphi), \omega\rangle$ if and only if $\mathcal{M}$ has arbitrarily long paths that begin in m, as stated in Example 1.

5.2 From automata to formulas

▶ **Theorem 9.** *For every countdown automaton $\mathcal{A}$ there exists a countdown formula $\varphi_\mathcal{A}$ s.t. $[\![\mathcal{A}]\!]^{\mathsf{val}} = [\![\varphi_\mathcal{A}]\!]^{\mathsf{val}}$ for every model $\mathcal{M}$ and valuation* val.

Proof (sketch). We sketch the construction of $\varphi_\mathcal{A}$. For an automaton $\mathcal{A} = (Q, q_I, \delta, \mathsf{rank}, \mathsf{ctr}_I)$, by induction on $r \in \mathcal{R}$ we build a formula $\psi_{r,q}$ for each $q \in Q$. Then we put $\varphi_\mathcal{A} = \psi_{r_{\mathsf{max}},q_I}$. Thus for the base case of the lowest rank $r = 0$:

- if $\delta(s) = (\tau, p)$ then for $\psi_{0,s}$ we put $\langle\tau\rangle x_p$ if q belongs to ∃ve and $[\tau]x_p$ if q belongs to ∀dam,
- if $\delta(s) \subseteq Q$ then for $\psi_{0,s}$ we put $\bigvee_{p\in\delta(s)} x_p$ if q belongs to ∃ve and $\bigwedge_{p\in\delta(s)} x_p$ if q belongs to ∀dam.

For the inductive step, let $q_1, \ldots, q_d$ be all states in Q with rank r. For every q_i define the vectorial formula:

$$\theta_i = \eta^\alpha_{q_i}(x_{q_1}, \ldots, x_{q_d}).(\psi_{r,q_1}, \ldots, \psi_{r,q_d})$$

with $\alpha = \mathsf{ctr}_I(r)$ and $\eta = \mu$ if r belongs to ∃ve and $\eta = \nu$ if r belongs to ∀dam. Then put $\psi_{r+1,q} = \psi_{r,q}[x_{q_1} \mapsto \theta_1, \ldots, x_{q_d} \mapsto \theta_d]$. ◀

6 Vectorial vs. scalar calculus

In this section we investigate the relation between scalar and vectorial formulas. We have already seen with Example 4 that unlike with standard fixpoints, the Bekić principle is not valid in the countdown setting. Interestingly, scalar formulas correspond to automata with a simple syntactic restriction.

▶ **Proposition 10.** *Scalar countdown formulas and automata where every two states have different ranks have equal expressive power.*

Proof (sketch). Inspecting the translations between formulas and automata from Sections 5.1 and 5.2, it is evident that injectively ranked automata are translated to scalar formulas, and that, although in our translation the choice of the assignment of ranks is not deterministic, every scalar formula can be translated to an injectively ranked automaton. ◀

Since the Bekić principle fails, a natural question is whether there is another way of transforming vectorial formulas to scalar form (or, equivalently, arbitrary countdown automata to injectively ranked ones). We shall give a negative answer in Theorem 11. However, before we proceed, let us analyse the following example, which shows that scalar formulas are more expressive than they may seem, covering in particular the property from Example 4.

6.1 Languages of unbounded infixes

Fix a regular language of finite words $L \subseteq \Gamma^*$. Let $\mathcal{U}(L) \subseteq \Gamma^\omega$ be the language of all infinite words that contain arbitrarily long infixes from L. For instance, the language from Example 4 is $\mathcal{U}(\mathsf{a}^*)$. We shall now show that $\mathcal{U}(L)$ can be defined in the countdown μ-calculus, first by a vectorial formula, then by a scalar one.

Consider a finite deterministic automaton $\mathcal{A} = (Q, \delta, q_I, F)$ that recognizes L. Let $\delta^+ : \Gamma^+ \times Q \to Q$ be the unique inductive extension of the transition function $\delta : \Gamma \times Q \to Q$ to nonempty words. Define $K_{p,q} = \{w \in \Gamma^+ \mid \delta^+(w, p) = q\}$ the (regular) language of nonempty words leading from p to q in $\mathcal{A}$, and let $K_{p,F}$ denote the union $\bigcup_{q \in F} K_{p,q}$. By the pigeonhole principle we have $\mathcal{U}(L) = \bigcup_{q \in Q} \mathcal{U}_q(L)$, where $\mathcal{U}_q(L) \subseteq \Gamma^\omega$ consists of words such that for every $n < \omega$, w has an infix $w_n = v_I u_1 ... u_n v_F \in L$ s.t. (i) $v_I \in K_{q_I,q}$, (ii) $u_1, ..., u_n \in K_{q,q}$, and (iii) $v_F \in K_{q,F}$. Then $\mathcal{U}_q(L)$ can be defined by a vectorial formula:

$$\mathcal{U}_q(L) = [\![\nu_1^\omega(x_1, x_2).(\langle \Gamma^* K_{q_I,q} \rangle x_2, \langle K_{q,q} \rangle x_2 \wedge \langle K_{q,F} \rangle \top)]\!]$$

where $\langle K \rangle \psi$ is the formula as explained in Example 4. Indeed, the corresponding semantic game on a word w proceeds as follows:

1. $\forall$dam chooses a number $n < \omega$ as the value of his only counter,
2. $\exists$ve skips a prefix $v_0 v_I \in \Gamma^* K_{q_I,q}$ of w,
3. $\forall$dam decrements his counter;
4. $\exists$ve keeps moving through $u_1, u_2, ... \in K_{q,q}$ so that after each step, some state in F is reachable from q by some prefix of the remaining word. After each such choice of u_i $\forall$dam has to decrement his counter, and so $\exists$ve wins iff she can make at least $n - 1$ such steps.

The two different stages in which $\forall$dam's counter is decremented reflect the two-phase dynamics of the game: first $\forall$dam challenges $\exists$ve with a number, and then $\exists$ve shows that she can provide an infix long enough.

It is more tricky to define the language $\mathcal{U}_q(L)$ with a scalar formula, but it turns out to be possible. To this end, observe that without loss of generality we may restrict attention to words w such that:

1. the infixes $w_n \in L$ start arbitrarily far in w;
2. each w_n can be decomposed as $v_I u_1 ... u_n v_F \in L$ s.t. (i) $v_I \in K_{q_I,q}$, (ii) $u_1, ..., u_n \in K_{q,q}$, (iii) $v_F \in K_{q,F}$, and additionally *(iv) all u_i begin with the same letter* $\mathsf{a} \in \Gamma$;
3. there are at least two distinct letters $\mathsf{a}, \mathsf{b} \in \Gamma$ that appear infinitely often in w;
4. the first letter of w is b.

Indeed, for (1) note that otherwise w_n start in the same position k for all n large enough. But then even the stronger property "There exists a position k such that the run of $\mathcal{A}$ from k visits q and F infinitely often" holds, and this is easily definable by a fixpoint formula.

Item (2) follows from the pigeonhole principle and the observation that in $w_{n \times |\Gamma|} = v_I u_1 ... u_{n \times |\Gamma|} v_F$ at least n u_i's begin with the same letter.

For (3) observe that otherwise w has a suffix a^ω for some $\mathsf{a} \in \Gamma$, in which case membership in $\mathcal{U}_q(L)$ is definable by a fixpoint formula. This is because an ultimately periodic word is bisimilar to a finite model, and so every monotone map reaches its fixpoints in finitely many steps, meaning that the countdown operator ν^ω is equivalent to ν^∞.

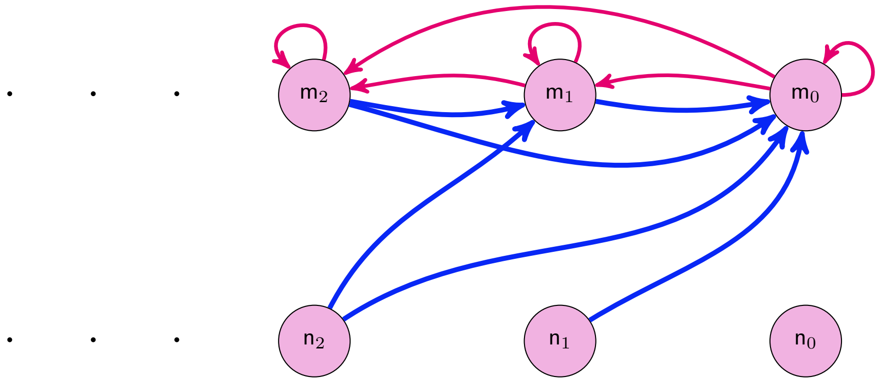

Figure 1 The model $\mathcal{M}$. Blue arrows represent edges labeled both with a and b, and pink arrows are edges labeled only with b.

Finally, for (4) note that the language $\mathcal{U}_q(L)$ is closed under adding and removing finite prefixes, and so if a formula φ defines $\mathcal{U}_q(L) \cap \mathsf{b}\Gamma^\omega$, then the formula $\langle \Gamma^* \rangle(\langle \mathsf{b} \rangle \top \wedge \varphi)$ defines $\mathcal{U}_q(L)$.

With this in mind, define:

$$\varphi = \nu^\omega x.(\langle \mathsf{b} \rangle \top \wedge \langle \Gamma^* K_{q_I,q} \rangle(\langle \mathsf{a} \rangle \top \wedge x)) \vee (\langle K_{q,q} \rangle(\langle \mathsf{a} \rangle \top \wedge x) \wedge \langle \mathsf{a} \rangle \top \wedge \langle K_{q,F} \rangle \top).$$

Note how $\langle \mathsf{b} \rangle \top \wedge x$ and $\langle \mathsf{a} \rangle \top \wedge x$ replace x_1 and x_2 from the vectorial formula. Consider the corresponding semantic game on a word w. Consider configurations of the game with the main disjunction as the formula component. Every infinite play of the game must visit such configurations infinitely often. In such a configuration, if the next letter in the model is either a or b then ∃ve must choose the right or left disjunct respectively. In particular, once the game reaches a configuration where $\langle \mathsf{a} \rangle \top$ holds, it must also hold every time the variable x in unraveled in the future. As a result, ∃ve wins from a configuration where $\langle \mathsf{a} \rangle \top$ holds against ∀dam's counter $n < \omega$ iff there is $u_1 ... u_{n+1} v_F$ starting in the current position such that $u_1, ..., u_{n+1} \in K_{q,q}$, each u_i starts with a, and $v_F \in K_{q,F}$. Moreover, ∃ve wins from a position where $\langle \mathsf{b} \rangle \top$ holds, against ∀dam's $n + 1 < \omega$, iff there is $v_I \in \Gamma^* K_{q_I,q}$ starting in the current position such that the next position after v_I satisfies $\langle \mathsf{a} \rangle \top$ and ∃ve wins from there against n. Putting this together, we get that ∃ve wins from a position satisfying $\langle \mathsf{b} \rangle \top$ against n iff there is $v_I u_1 ... u_n v_F = w_n$ as in condition (2) above. Since the game starts with ∀dam choosing an arbitrary $n < \omega$, it follows that indeed φ defines $\mathcal{U}_q(L)$.

6.2 Greater expressive power of the vectorial calculus

We now show an example of a property that is definable in the vectorial countdown calculus but not in the scalar one.

Fixing $\mathsf{Act} = \{\mathsf{a}, \mathsf{b}\}$, consider a model $\mathcal{M} = (M, \xrightarrow{\mathsf{a}}, \xrightarrow{\mathsf{b}})$ with points $M = \{\mathsf{m}_i, \mathsf{n}_i \mid i < \omega\}$, and with exactly the edges: $\mathsf{m}_i \xrightarrow{\mathsf{a}} \mathsf{m}_j$, $\mathsf{n}_i \xrightarrow{\mathsf{a}} \mathsf{m}_j$ and $\mathsf{n}_i \xrightarrow{\mathsf{b}} \mathsf{m}_j$ for all $i > j$; and $\mathsf{m}_i \xrightarrow{\mathsf{b}} \mathsf{m}_j$ for all i and j. Note that the relation $\xrightarrow{\mathsf{a}}$ is a subset of $\xrightarrow{\mathsf{b}}$. The model is shown in Fig. 1.

Consider the vectorial sentence $\nu^\omega_1(x_1, x_2).(\langle \mathsf{b} \rangle x_2, \langle \mathsf{a} \rangle x_2)$. This describes the property *there are arbitrarily long paths with labels in* ba^*, and so it is true in all points m_i and false in all points n_i. The following result immediately implies that this property cannot be defined in the scalar countdown calculus:

▶ **Theorem 11.** *For every scalar sentence φ, there exists $i < \omega$ s.t.* $\mathsf{n}_i \in [\![\varphi]\!] \iff \mathsf{m}_i \in [\![\varphi]\!]$.

Proof (sketch). The heart of the proof is Proposition 10 which says that scalar formulas correspond to injectively ranked automata. In such an automaton, whenever the counter corresponding to rank r is modified, the automaton must be in *the same state*, which allows the players to copy their strategies between different positions of the semantic game. ◀

7 Strictness of the countdown nesting hierarchy

A natural question is whether greater *coutdown nesting*, i.e. the maximal nesting of μ^α and ν^α operators with $\alpha \neq \infty$, results in more expressive power. We give a positive answer: under mild assumptions, the hierarchy is strict. From now on, focus on the monomodal case (i.e. $|\mathsf{Act}| = 1$) and we assume that the only ordinal used by formulas is ω.[1]

▶ **Theorem 12.** *For every $k < \omega$, formulas with countdown nesting $k + 1$ have strictly more expressive power than those with nesting at most k.*

In order to prove strictness, it suffices to prove it on a restricted class of models. We will show that the hierarchy is strict already on the class of transitive, linear, well-founded models – i.e. (up to isomorphism) ordinals.

More specifically, an ordinal $\kappa \in \mathsf{Ord}$ can be seen as a model with $\alpha \to \beta$ iff $\alpha > \beta$. Since κ is an induced submodel of κ' whenever $\kappa \leq \kappa'$, we can consider a single ordinal model with κ big enough. For our purposes, the first uncountable ordinal ω_1 is be sufficient.

We call a subset $S \subseteq \omega_1$ *stable above* α if either $[\alpha, \omega_1) \subseteq S$ or $[\alpha, \omega_1) \cap S = \emptyset$. A *stabilization point* of a valuation $\mathsf{val} : \mathsf{Var} \to \mathcal{P}(\omega_1)$ is the least $\alpha \leq \omega_1$ such that interpretations of all the variables are stable above α.

Observe that the set $[\omega^k, \omega_1) \subseteq [0, \omega_1)$ can be defined by the following sentence with countdown nesting k:

$$[\omega^k, \omega_1) = [\![\nu^\omega x_1 ... \nu^\omega x_k . \Diamond(\textstyle\bigwedge_{i \leq k} x_i)]\!]. \tag{6}$$

Indeed, the semantic game can be decomposed into two alternating phases: (i) ∀dam chooses a tuple of finite ordinals $(\alpha_1, ..., \alpha_k) \in \omega^k$ and (ii) ∃ve responds with a successor in the model. Since at each step ∀dam has to pick a lexicographically smaller tuple (and he starts by picking any tuple) it is easy to see that he wins iff the initial point is at least ω^k. We will show that for all $k > 0$, countdown nesting k is *necessary* to define this language. The proof relies on the following lemma.

▶ **Lemma 13.** *For every $k < \omega$ and a formula φ with countdown nesting k, there exists an ordinal $\alpha_\varphi < \omega^{k+1}$ such that φ stabilizes α_φ above the valuation, i.e. for every valuation* val *stabilizing at β, $[\![\varphi]\!]^{\mathsf{val}}$ is stable above $\beta + \alpha_\varphi$.*

Proof (sketch). By induction on the complexity of the formula φ. The base case is immediate, as for every $x \in \mathsf{Var}$ it suffices to take $\alpha_x = 0$. For propositional connectives and modal operators we take $\alpha_{\psi_1 \vee \psi_2} = \alpha_{\psi_1 \wedge \psi_2} = \mathsf{max}(\alpha_{\psi_1}, \alpha_{\psi_2})$ and $\alpha_{\Diamond\psi} = \alpha_{\Box\psi} = \alpha_\psi + 1$. The most interesting case are countdown and fixpoint operators. There the lemma follows from the fact that for every formula φ there is a finite constant $t_\varphi < \omega$ such that for every valuation val stable above κ, in the part $[\kappa, \omega_1)$ of the model above κ, φ changes its truth value at most t_φ times. ◀

[1] This assumption could be replaced with a weaker requirement: there exists a maximal ordinal α that we are allowed to use, and α is additively indecomposable.

From this the theorem follows immediately, as the sentence φ has no free variables and thus it stabilizes at $\alpha_\varphi < \omega^{k+1}$ regardless of the valuation.

8 Decidability issues

We briefly discuss decidability issues in the countdown μ-calculus. Note that in a finite model every monotone map reaches its fixpoints in finitely many steps. Hence, if we replace every η^α in φ with η^∞ and denote the resulting formula by $\widehat{\varphi}$, then *in every finite model* $[\![\varphi]\!] = [\![\widehat{\varphi}]\!]$. It immediately follows that:

▶ **Proposition 14.** *The model checking problem for the* μ^α-ML*, i.e. the problem: "Given* $\varphi \in \mu^\alpha$-ML *and a point* m *in a (finite) model* $\mathcal{M}$*, does* $\mathsf{m} \models \varphi$*?" is decidable.*

Note that as a corollary we get that deciding the winner of a given (finite) countdown game $\mathcal{G}$ is also decidable, as set of positions where ∃ve wins can be easily defined in μ^α-ML.

A more interesting problem is *satisfiability*: "Given $\varphi \in \mu^\alpha$-ML, is there a model $\mathcal{M}$ and a point m s.t. $\mathsf{m} \models \varphi$?".

▶ **Proposition 15.** *A formula* $\varphi \in \mu^\alpha$-ML *has* positive countdown *if it does not use* ν^α *with* $\alpha \neq \infty$*. The satisfiability problem is decidable for such formulas.*

Proof. Observe that for φ with positive countdown, in every model we have $[\![\varphi]\!] \subseteq [\![\widehat{\varphi}]\!]$. Hence, if φ is satisfiable, then so is $\widehat{\varphi}$ – but since μ-ML has a finite model property, this means that $\widehat{\varphi}$ has a *finite* model, where $\widehat{\varphi}$ and φ are equivalent. Thus, φ is satisfiable iff $\widehat{\varphi}$ is, and the problem reduces to μ-ML satisfiability. ◀

Dualizing the above we get that the *validity* problem is decidable for formulas with *negative countdown*, i.e. with $\alpha = \infty$ for every μ^α.

References

1. André Arnold and Damian Niwinski. *Rudiments of calculus*. Elsevier, 2001.
2. Mikołaj Bojańczyk. Weak mso with the unbounding quantifier. *Theory of Computing Systems*, 48(3):554–576, 2011. doi:10.1007/s00224-010-9279-2.
3. Mikołaj Bojańczyk, Edon Kelmendi, Rafał Stefański, and Georg Zetzsche. Extensions of ω-regular languages. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 266–272, 2020. doi:10.1145/3373718.3394779.
4. Mikolaj Bojanczyk, Pawel Parys, and Szymon Torunczyk. The MSO+U Theory of (N,<) Is Undecidable. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:8, 2016. doi:10.4230/LIPIcs.STACS.2016.21.
5. Mikołaj Bojanczyk and Szymon Torunczyk. Weak MSO+U over infinite trees. In *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, volume 14 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 648–660, 2012. doi:10.4230/LIPIcs.STACS.2012.648.
6. Mikołaj Bojańczyk. A bounding quantifier. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic*, pages 41–55, September 2004. doi:10.1007/978-3-540-30124-0_7.
7. Mikołaj Bojańczyk and Thomas Colcombet. Bounds in w-regularity. In *Proceedings - Symposium on Logic in Computer Science*, pages 285–296, January 2006. doi:10.1109/LICS.2006.17.
8. Mikołaj Bojańczyk, Laure Daviaud, Bruno Guillon, Vincent Penelle, and A. V. Sreejith. Undecidability of mso+"ultimately periodic". *ArXiv*, abs/1807.08506, 2018.

9 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasi-polynomial time. *SIAM Journal on Computing*, 0(0), 2017. `doi:10.1137/17M1145288`.

10 Thomas Colcombet. Regular Cost Functions, Part I: Logic and Algebra over Words. *Logical Methods in Computer Science*, Volume 9, Issue 3, 2013. `doi:10.2168/LMCS-9(3:3)2013`.

11 Lauri Hella, Antti Kuusisto, and Raine Rönnholm. Bounded game-theoretic semantics for modal mu-calculus. In *GandALF*, 2020.

12 Szczepan Hummel and Michał Skrzypczak. The topological complexity of mso+u and related automata models. *Fundamenta Informaticae*, 119:87–111, 2012.

13 David Janin and Igor Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In Ugo Montanari and Vladimiro Sassone, editors, *CONCUR '96: Concurrency Theory*, pages 263–277, 1996.

14 Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983. `doi:10.1016/0304-3975(82)90125-6`.

15 Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016. `doi:10.1007/978-3-662-52947-8`.

16 Yde Venema. Lectures on the modal μ-calculus, 2020.

17 Igor Walukiewicz. Pushdown processes: Games and model-checking. *Information and Computation*, 164(2):234–263, 2001. `doi:10.1006/inco.2000.2894`.