HAL
open science

On the expressive power of non-deterministic and unambiguous Petri nets over infinite words

Olivier Finkel, Michal Skrzypczak

To cite this version:

Olivier Finkel, Michal Skrzypczak. On the expressive power of non-deterministic and unambiguous Petri nets over infinite words. Fundamenta Informaticae, 2021, 183 (3-4), pp.243-291. 10.3233/FI-2021-2088 . hal-03287498v2

HAL Id: hal-03287498

https://hal.science/hal-03287498v2

Submitted on 27 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Appeared in Fundamenta Informaticae 183(3-4) : 243–291 (2021).
Available at IOS Press through:
https://doi.org/10.3233/FI-2021-2088

On the Expressive Power of Non-deterministic and Unambiguous Petri Nets over Infinite Words

Olivier Finkel*

Institut de Mathématiques de Jussieu - Paris Rive Gauche

CNRS, Université de Paris, Sorbonne Université, Paris, France.

Olivier.Finkel@math.univ-paris-diderot.fr

Michał Skrzypczak†

Institute of Informatics

University of Warsaw

Banacha 2, 02-097 Warsaw, Poland

mskrzypczak@mimuw.edu.pl

Abstract. We prove that ω-languages of (non-deterministic) Petri nets and ω-languages of (non-deterministic) Turing machines have the same topological complexity: the Borel and Wadge hierarchies of the class of ω-languages of (non-deterministic) Petri nets are equal to the Borel and Wadge hierarchies of the class of ω-languages of (non-deterministic) Turing machines. We also show that it is highly undecidable to determine the topological complexity of a Petri net ω-language. Moreover, we infer from the proofs of the above results that the equivalence and the inclusion problems for ω-languages of Petri nets are Π^1_2-complete, hence also highly undecidable.

Additionally, we show that the situation is quite the opposite when considering unambiguous Petri nets, which have the semantic property that at most one accepting run exists on every input. We provide a procedure of determinising them into deterministic Muller counter machines with counter copying. As a consequence, we entail that the ω-languages recognisable by unambiguous Petri nets are $\mathbf{\Delta}^0_3$ sets.

*Address for correspondence: CNRS et Université de Paris, UFR de Mathématiques case 7012, 75205 Paris Cedex 13, France

†Author supported by Polish National Science Centre grant 2016/22/E/ST6/00041.

Keywords: Automata and formal languages, Petri nets, Infinite words, Logic in computer science, Cantor topology, Borel hierarchy, Wadge degrees, Highly undecidable properties, Unambiguous Petri nets

1. Introduction

In the sixties, Büchi was the first to study acceptance of infinite words by finite automata with the now called Büchi acceptance condition, in order to prove the decidability of the monadic second order theory of one successor over the integers. Since then there has been a lot of work on regular ω-languages, accepted by Büchi automata, or by some other variants of automata over infinite words, like Muller or Rabin automata, see [1, 2, 3]. The acceptance of infinite words by other finite machines, like pushdown automata, counter automata, Petri nets, Turing machines, ..., with various acceptance conditions, has also been studied, see [2, 4, 5, 6, 7].

The Cantor topology is a very natural topology on the set Σ^ω of infinite words over a finite alphabet Σ which is induced by the prefix metric. Then a way to study the complexity of languages of infinite words accepted by finite machines is to study their topological complexity and firstly to locate them with regard to the Borel and the projective hierarchies [1, 4, 8, 2].

Every ω-language accepted by a deterministic Büchi automaton is a $\mathbf{\Pi}^0_2$-set. On the other hand, it follows from McNaughton's Theorem that every regular ω-language is accepted by a deterministic Muller automaton, and thus is a Boolean combination of ω-languages accepted by deterministic Büchi automata. Therefore every regular ω-language is a $\mathbf{\Delta}^0_3$-set. Moreover, Landweber proved that the Borel complexity of any ω-language accepted by a Muller or Büchi automaton can be effectively computed (see [9, 3]). In a similar way, every ω-language accepted by a deterministic Muller Turing machine, and thus also by any Muller deterministic finite machine is a $\mathbf{\Delta}^0_3$-set, [4, 2].

The Wadge hierarchy is a great refinement of the Borel hierarchy, firstly defined by Wadge via reductions by continuous functions [10]. The trace of the Wadge hierarchy on the regular ω-languages is called the Wagner hierarchy. It has been completely described by Klaus Wagner in [11]. Its length is the ordinal ω^ω. Wagner gave an automaton-like characterisation of this hierarchy, based on the notions of chain and superchain, together with an algorithm to compute the Wadge (Wagner) degree of any given regular ω-language, see also [12, 13, 14, 15].

The Wadge hierarchy of deterministic context-free ω-languages was determined by Duparc in [16, 17]. Its length is the ordinal $\omega^{(\omega^2)}$. We do not know yet whether this hierarchy is decidable or not. But the Wadge hierarchy induced by deterministic partially blind 1-counter automata was described in an effective way in [18], and other partial decidability results were obtained in [19]. Then, it was proved in [20] that the Wadge hierarchy of 1-counter or context-free ω-languages and the Wadge hierarchy of effective analytic sets (which form the class of all the ω-languages accepted by non-deterministic Turing machines) are equal. Moreover, similar results hold about the Wadge hierarchy of infinitary rational relations accepted by 2-tape Büchi automata, [21]. Finally, the Wadge hierarchy of ω-languages of deterministic Turing machines was determined by Selivanov in [22].

We consider in this paper acceptance of infinite words by Petri nets. Petri nets are used for the description of distributed systems [23, 24, 25, 26, 27], and form a very important mathematical model in Concurrency Theory that has been developed for general concurrent computation. In the context of

Automata Theory, Petri nets may be defined as (partially) blind multicounter automata, as explained in [6, 4, 28]. First, one can distinguish between the places of a given Petri net by dividing them into the bounded ones (the number of tokens in such a place at any time is uniformly bounded) and the unbounded ones. Then each unbounded place may be seen as a partially blind counter, and the tokens in the bounded places determine the state of the partially blind multicounter automaton that is equivalent to the initial Petri net. The transitions of the Petri net may then be seen as the finite control of the partially blind multicounter automaton and the labels of these transitions are then the input symbols.

The infinite behaviour of Petri nets (i.e., Petri nets running over ω-words) was first studied by Valk [6] and by Carstensen in the case of deterministic Petri nets [29]. The topological complexity of ω-languages of *deterministic* Petri nets was completely determined. They are $\mathbf{\Delta}^0_3$-sets and their Wadge hierarchy has been determined by Duparc, Finkel, and Ressayre in [30]; its length is the ordinal ω^{ω^2}. On the other side, Finkel and Skrzypczak proved in [31] that there exist $\mathbf{\Sigma}^0_3$-complete, hence non $\mathbf{\Delta}^0_3$, ω-languages accepted by *non-deterministic* one-partially-blind-counter Büchi automata.

A next goal was to understand the expressive power of *non-determinism* in Petri nets, i.e. blind multicounter automata. The assumption of blindness is important here, as it is already known that ω-languages accepted by (non-blind) one-counter Büchi automata have the same topological complexity as ω-languages of Turing machines [20]. However, the non-blindness of the counter, i.e. the ability to use the zero-test of the counter, was essential in the proof of this result.

The first author proved in [32, 33] that ω-languages of *non-deterministic* Petri nets and effective analytic sets have the same topological complexity. More precisely the Borel and Wadge hierarchies of the class of ω-languages of Petri nets are equal to the Borel and Wadge hierarchies of the class of effective analytic sets. The proof is based on a simulation of a given real time 1-counter (with zero-test) Büchi automaton $\mathcal{A}$ accepting ω-words x over an alphabet Σ by a real time 4-blind-counter Büchi automaton $\mathcal{B}$ reading some special codes $h(x)$ of the words x. In particular, for each non-null recursive ordinal $\alpha < \omega_1^{\mathrm{CK}}$ there exist some $\mathbf{\Sigma}^0_\alpha$-complete and some $\mathbf{\Pi}^0_\alpha$-complete ω-languages of Petri nets, and the supremum of the set of Borel ranks of ω-languages of Petri nets is the ordinal γ^1_2, which is strictly greater than the first non-recursive ordinal ω_1^{CK}. Moreover it is proved in [32, 33] that it is highly undecidable to determine the topological complexity of a Petri net ω-language. Moreover, it is inferred from the proofs of the above results that also the equivalence and the inclusion problems for ω-languages of Petri nets are Π^1_2-complete, hence also highly undecidable.

A particular instance of the above construction of simulation is for the Wadge degrees of analytic sets $\mathbf{\Sigma}^1_1$ — it follows that there exists a real time 4-blind-counter Büchi automaton recognising a $\mathbf{\Sigma}^1_1$-complete ω-language, but also that it is consistent with the axiomatic system **ZFC** of set theory that there exists some non-Borel non-$\mathbf{\Sigma}^1_1$-complete ω-language of 4-blind-counter Büchi automata, [32, 33].

The first author also proved in [32] that the determinacy of Wadge games between two players in charge of ω-languages of Petri nets is equivalent to the effective analytic (Wadge) determinacy, and thus is not provable in the axiomatic system **ZFC**.

The second author independently proved in [34] that only one blind counter is enough to obtain a non-Borel ω-language — he exhibits an example of a real time 1-blind-counter Büchi automaton that also recognises a $\mathbf{\Sigma}^1_1$-complete ω-language.

In the present journal paper we gather results of both authors from [32, 33] and [34] with some complements which did not appear in these papers:

In particular, we prove that there exists a Petri net accepting an ω-language L such that L is a Borel $\mathbf{\Pi}^0_2$-set in one model of **ZFC** and non-Borel in another model of **ZFC** (Theorem 4.18). We also show that it is undecidable to determine the topological complexity of the ω-language accepted by a given 1-blind counter automaton and, as a consequence, we prove several other undecidability results for ω-languages accepted by 1-blind counter automata (Theorems 8.9, 8.10, 8.12, and 9.4).

We also study the important case of unambiguous Petri nets. In that case, we provide a determinisation procedure for unambiguous blind counter automata that constructs an equivalent deterministic Muller counter machine with zero tests and counter copying, hence also an equivalent deterministic Muller Turing machine. Its determinism already guarantees tight bounds on the topological complexity of unambiguous blind counter automata: their ω-languages belong to the Borel class $\mathbf{\Delta}^0_3$. Notice that the blindness of the counters is crucial here since one can obtain some $\mathbf{\Sigma}^0_3$-complete and $\mathbf{\Pi}^0_3$-complete ω-languages accepted by unambiguous 1-counter automata with zero-test [35]. The topological complexity of ω-languages of unambiguous Petri nets may also been compared with the complexity of ω-languages of unambiguous Turing machines: they form the class of effective Δ^1_1-sets which contains $\mathbf{\Sigma}^0_\alpha$-complete sets and $\mathbf{\Pi}^0_\alpha$-complete sets for each recursive ordinal $\alpha < \omega_1^{\mathrm{CK}}$ [36].

This paper is an extended journal version of both the paper [33] of the first author which appeared in the Proceedings of the 41st International Conference on Application and Theory of Petri Nets and Concurrency, Petri Nets 2020, which took place virtually in Paris on June 2020, and of the paper [34] of the second author which appeared in the Proceedings of the 12th International Conference on Reachability Problems, which took place in Marseille on September 2018.

The paper is organised as follows. In Section 2 we review the notions of (blind) counter automata and ω-languages. In Section 3 we recall notions of topology, and the Borel and Wadge hierarchies on a Cantor space. Section 4 is devoted to the main result of that work: the simulation construction that provides 4-blind counter Büchi automata for levels of the Wadge hierarchy occupied by 1-counter Büchi automata. Based on this construction, we show that the topological or arithmetical complexity of a Petri net ω-language is highly undecidable in Section 5. The equivalence and the inclusion problems for ω-languages of Petri nets are shown to be Π^1_2-complete in Section 6. Section 7 is devoted to a study of determinacy of Wadge games with winning conditions given by ω-languages recognised by blind counter automata. The example of a $\mathbf{\Sigma}^1_1$-complete ω-language recognised by a 1-blind counter Büchi automaton is given in Section 8. Section 9 is devoted to consequences of the high topological complexity of ω-languages recognisable by Petri nets with regard to inherent non-determinism and ambiguity. Section 10 provides a determinisation construction based on the assumption that a given automaton is unambiguous. Concluding remarks are given in Section 11. An additional Appendix A contains an extensive explanation of the technical construction from Section 10.

2. Basic notions

We assume the reader to be familiar with the theory of formal (ω-)languages [2, 3]. We recall the usual notations of formal language theory.

If Σ is a finite alphabet, a *non-empty finite word* over Σ is any sequence $u = a_1 \dots a_k$, where $a_i \in \Sigma$ for $i = 1, \dots, k$, and k is an integer ≥ 1. The *length* of u is k, denoted by $|u|$. The *empty word* is denoted by ϵ; its length is 0. $\Sigma^\star$ is the *set of finite words* (including the empty word) over Σ, and we denote $\Sigma^+ =_{\text{df}} \Sigma^\star \setminus \{\epsilon\}$. A (finitary) *language* V over an alphabet Σ is a subset of $\Sigma^\star$.

The *first infinite ordinal* is ω. An ω-*word* over Σ is an ω-sequence $a_1 \dots a_n \dots$, where for all integers $i \geq 1$, $a_i \in \Sigma$. When $\sigma = a_1 \dots a_n \dots$ is a finite word of length at least n or an ω-word, we write $\sigma(n) = a_n$, $\sigma[n] = \sigma(1)\sigma(2)\dots\sigma(n)$ for all $n \geq 1$ and $\sigma[0] = \epsilon$.

The usual concatenation product of two finite words u and v is denoted $u \cdot v$ (and sometimes just uv). This product is extended to the product of a finite word u and an ω-word v: the infinite word $u \cdot v$ is then the ω-word such that:

$(u \cdot v)(k) = u(k)$ if $k \leq |u|$, and $(u \cdot v)(k) = v(k - |u|)$ if $k > |u|$.

The *set of* ω-*words* over an alphabet Σ is denoted by Σ^ω. An ω-*language* V over an alphabet Σ is a subset of Σ^ω, and its complement (in Σ^ω) is $\Sigma^\omega \setminus V$, denoted V^-.

The *prefix relation* is denoted $\sqsubseteq$: a finite word u is a *prefix* of a finite word v (respectively, an infinite word v), denoted $u \sqsubseteq v$, if and only if there exists a finite word w (respectively, an infinite word w), such that $v = u \cdot w$.

2.1. Counter automata

Let k be an integer with $k \geq 1$. A k-counter machine has k *counters*, each of which containing a non-negative integer. The machine can test whether the content of a given counter is zero or not, but this is not possible if the counter is a blind (sometimes called partially blind, as in [28]) counter. This means that if a transition of the machine is enabled when the content of a blind counter is zero then the same transition is also enabled when the content of the same counter is a positive integer. The transitions depend on the letter read by the machine, the current state of the finite control, and the tests about the values of the counters. Notice that in the sequel we shall only consider real-time automata, i.e. ϵ-transitions are not allowed (but the general results of this paper will be easily extended to the case of non-real-time automata).

Formally, a non-deterministic *real time* k-counter machine is a 4-tuple $\mathcal{M} = \langle K, \Sigma, \Delta, q_0 \rangle$, where K is a finite set of *states*, Σ is a finite input alphabet, $q_0 \in K$ is an *initial state*, and $\Delta \subseteq K \times \Sigma \times \{0,1\}^k \times K \times \{0,1,-1\}^k$ is a *transition relation*.

If the machine $\mathcal{M}$ is in a state q and $c_i \in \mathbb{N}$ is the content of the ith counter $\mathcal{C}_i$ for $i = 1, \dots, k$; then the *configuration* (or *global state*) of $\mathcal{M}$ is the $(k+1)$-tuple $(q, c_1, \dots, c_k)$.

Consider $a \in \Sigma$, $q, q' \in K$, and $(c_1, \dots, c_k) \in \mathbb{N}^k$ such that $c_j = 0$ for $j \in E \subseteq \{1, \dots, k\}$ and $c_j > 0$ for $j \notin E$. If $(q, a, i_1, \dots, i_k, q', j_1, \dots, j_k) \in \Delta$ where $i_j = 0$ for $j \in E$ and $i_j = 1$ for $j \notin E$, then we write:

$$a : (q, c_1, \dots, c_k) \mapsto_{\mathcal{M}} (q', c_1 + j_1, \dots, c_k + j_k).$$

Thus the transition relation must obviously satisfy:
if $(q, a, i_1, \dots, i_k, q', j_1, \dots, j_k) \in \Delta$ and $i_m = 0$ for some $m \in \{1, \dots, k\}$ then $j_m = 0$ or $j_m = 1$ (but j_m may not be equal to -1).

Moreover, if the counters of $\mathcal{M}$ are blind, then, if $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_k) \in \Delta$ holds, and $i_m = 0$ for some $m \in \{1, \ldots, k\}$ then $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_k) \in \Delta$ also holds if $i_m = 1$ and the other integers are unchanged.

An ω-sequence of configurations $r = (q_i, c_1^i, \ldots c_k^i)_{i \geq 1}$ is called a run of $\mathcal{M}$ on an ω-word $\sigma = a_1 a_2 \ldots a_n \ldots$ over Σ iff:

(1) $(q_1, c_1^1, \ldots c_k^1) = (q_0, 0, \ldots, 0)$

(2) for each $i \geq 1$,

$$a_i : (q_i, c_1^i, \ldots c_k^i) \mapsto_{\mathcal{M}} (q_{i+1}, c_1^{i+1}, \ldots c_k^{i+1}).$$

For every such run r, $\mathrm{In}(r)$ is the set of all states visited infinitely many times during r.

Definition 2.1. A Büchi k-counter automaton is a 5-tuple $\mathcal{M} = \langle K, \Sigma, \Delta, q_0, F \rangle$, where $\mathcal{M}' = \langle K, \Sigma, \Delta, q_0 \rangle$ is a k-counter machine and $F \subseteq K$ is a set of *accepting states*. The ω-language accepted by $\mathcal{M}$ is: $\mathrm{L}(\mathcal{M}) =_{\mathrm{df}} \{\sigma \in \Sigma^\omega \mid$ there exists a run r of $\mathcal{M}$ on σ such that $\mathrm{In}(r) \cap F \neq \emptyset\}$.

Definition 2.2. A Muller k-counter automaton is a 5-tuple $\mathcal{M} = \langle K, \Sigma, \Delta, q_0, \mathcal{F} \rangle$, where $\mathcal{M}' = \langle K, \Sigma, \Delta, q_0 \rangle$ is a k-counter machine and $\mathcal{F} \subseteq 2^K$ is a set of accepting sets of states. The ω-language accepted by $\mathcal{M}$ is: $\mathrm{L}(\mathcal{M}) =_{\mathrm{df}} \{\sigma \in \Sigma^\omega \mid$ there exists a run r of $\mathcal{M}$ on σ such that $\mathrm{In}(r) \in \mathcal{F}\}$.

Given a configuration $(q, c_1, \ldots, c_k)$ one can extend the above definitions to $\mathrm{L}\big(\mathcal{M}, (q, c_1, \ldots, c_k)\big)$ which is the set of ω-words accepted by $\mathcal{M}$ starting from the configuration $(q, c_1, \ldots, c_k)$.

A counter machine $\mathcal{M} = \langle K, \Sigma, \Delta, q_0 \rangle$ is *deterministic* if its transition relation Δ is functional in the following sense: for each $q \in K$, $a \in \Sigma$, and $(i_1, \ldots, i_k) \in \{0, 1\}^k$ there is exactly one transition of the machine of the form $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_k)$ for some $q' \in K$ and $(j_1, \ldots, j_k) \in \{-1, 0, 1\}^k$.

Notice that the definition of a deterministic counter machine $\mathcal{M}$ ensures that for every input ω-word $\sigma = a_1 a_2 \ldots$ there exists a unique run r of $\mathcal{M}$ on σ. This observation motivates the following definition. We call a Büchi (resp. Muller) k-counter automaton $\mathcal{M}$ *unambiguous* if for every input ω-word σ there exists at most one accepting run r of $\mathcal{M}$ on σ. Notice that the uniqueness requirement is not limited to the sequence of states visited in r but also the exact counter values — two runs visiting the same states but with different counter values are considered distinct. Similarly, we say that $\mathcal{M}$ is *countably unambiguous* if for every input ω-word σ there exist at most countably many accepting runs of $\mathcal{M}$ on σ.

The semantic condition of unambiguity is known to be very intriguing [37], with certain tractability features [38, 39], and expressive power ranging between deterministic and non-deterministic models [40, 41]. Also, various *degrees of ambiguity*, have been studied, see e.g. [42, 36, 35, 43, 44] and the references therein.

There is a natural simulation order on the configurations of a counter automaton: a configuration $(q, c_1, \ldots, c_k)$ *simulates* $(q, c'_1, \ldots, c'_k)$ (denoted $(q, c_1, \ldots, c_k) \succeq (q, c'_1, \ldots, c'_k)$) if they have the same state q and the counter values c_i and c'_i satisfy coordinate-wise $c_j \geq c'_j$ for $j = 1, \ldots, k$.

Remark 2.3. If $(q, c_1, \ldots, c_k) \succeq (q, c'_1, \ldots, c'_k)$ are two configurations of a k-blind counter machine $\mathcal{M}$ then $\mathrm{L}\big(\mathcal{M}, (q, c_1, \ldots, c_k)\big) \supseteq \mathrm{L}\big(\mathcal{M}, (q, c'_1, \ldots, c'_k)\big)$.

Proof:
It is enough to notice that each accepting run of $\mathcal{M}$ from $(q, c'_1, \ldots, c'_k)$ can be lifted to an accepting run from $(q, c_1, \ldots, c_k)$ just by increasing the counter values. □

The above remark relies heavily on the assumption of the blindness of counters. Moreover, it is important that the acceptance condition of the machines is defined purely in terms of the visited states — the counter values do not intervene.

The above remark implies that, if there is exactly one counter, the maximal size of an anti-chain of the simulation order is bounded by the number of states.

It is well known that an ω-language is accepted by a non-deterministic (real time) Büchi k-counter automaton iff it is accepted by a non-deterministic (real time) Muller k-counter automaton [4]. Notice that it cannot be shown without using the non determinism of automata and this result is no longer true in the deterministic case.

The class of ω-languages accepted by real time k-counter Büchi automata (respectively, real time k-blind-counter Büchi automata) is denoted $\mathbf{r}\text{-}\mathbf{CL}(k)_\omega$ (respectively, $\mathbf{r}\text{-}\mathbf{BCL}(k)_\omega$). (Notice that in previous papers, as in [20], the class $\mathbf{r}\text{-}\mathbf{CL}(k)_\omega$ was denoted $\mathbf{r}\text{-}\mathbf{BCL}(k)_\omega$ so we have slightly changed the notation in order to distinguish the different classes).

The class $\mathbf{CL}(1)_\omega$ is a strict subclass of the class $\mathbf{CFL}_\omega$ of context free ω-languages accepted by pushdown Büchi automata.

If we omit the counter of a real-time Büchi 1-counter automaton, then we simply get the notion of Büchi automaton. The class of ω-languages accepted by Büchi automata is the class of regular ω-languages.

3. Hierarchies in Cantor space

3.1. Borel hierarchy and analytic sets

We assume the reader to be familiar with basic notions of topology which may be found in [45, 8, 2, 3]. There is a natural metric on the set Σ^ω of infinite words over a finite alphabet Σ containing at least two letters which is called the *prefix metric* and is defined as follows. For $u, v \in \Sigma^\omega$ and $u \neq v$ let $\delta(u, v) = 2^{-l_{\mathrm{pref}(u,v)}}$ where $l_{\mathrm{pref}(u,v)}$ is the first integer n such that the $(n{+}1)^{st}$ letter of u is different from the $(n{+}1)^{st}$ letter of v. This metric induces on Σ^ω the usual Cantor topology in which the *open subsets* of Σ^ω are of the form $W \cdot \Sigma^\omega$, for $W \subseteq \Sigma^\star$. A set $L \subseteq \Sigma^\omega$ is a *closed set* iff its complement $\Sigma^\omega - L$ is an open set.

Define now the *Borel hierarchy* of subsets of Σ^ω:

Definition 3.1. For a non-null countable ordinal α, the classes $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$ of the Borel hierarchy on the topological space Σ^ω are defined as follows:
$\mathbf{\Sigma}^0_1$ is the class of open subsets of Σ^ω, $\mathbf{\Pi}^0_1$ is the class of closed subsets of Σ^ω,
and for any countable ordinal $\alpha \geq 2$:
$\mathbf{\Sigma}^0_\alpha$ is the class of countable unions of subsets of Σ^ω in $\bigcup_{\gamma<\alpha} \mathbf{\Pi}^0_\gamma$.
$\mathbf{\Pi}^0_\alpha$ is the class of countable intersections of subsets of Σ^ω in $\bigcup_{\gamma<\alpha} \mathbf{\Sigma}^0_\gamma$.

The class of *Borel sets* is $\mathbf{\Delta}^1_1 =_{\rm df} \bigcup_{\xi<\omega_1} \mathbf{\Sigma}^0_\xi = \bigcup_{\xi<\omega_1} \mathbf{\Pi}^0_\xi$, where ω_1 is the first uncountable ordinal. There are also some subsets of Σ^ω which are not Borel. In particular the class of Borel subsets of Σ^ω is strictly included into the class $\mathbf{\Sigma}^1_1$ of *analytic sets* which are obtained by projection of Borel sets.

Definition 3.2. A subset A of Σ^ω is in the class $\mathbf{\Sigma}^1_1$ of *analytic* sets if the following condition is satisfied: there exists another finite set Y and a Borel subset B of $(\Sigma \times Y)^\omega$ such that $x \in A$ iff $\exists y \in Y^\omega$ such that $(x, y) \in B$, where (x, y) is the infinite word over the alphabet $\Sigma \times Y$ such that $(x, y)(i) = (x(i), y(i))$ for each integer $i \geq 1$.

We now define completeness with regard to reduction by continuous functions. For a countable ordinal $\alpha \geq 1$, a set $F \subseteq \Sigma^\omega$ is said to be a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$, $\mathbf{\Sigma}^1_1$)-*complete set* iff for any set $E \subseteq Y^\omega$ (with Y a finite alphabet): $E \in \mathbf{\Sigma}^0_\alpha$ (respectively, $E \in \mathbf{\Pi}^0_\alpha$, $E \in \mathbf{\Sigma}^1_1$) iff there exists a continuous function $f : Y^\omega \to \Sigma^\omega$ such that $E = f^{-1}(F)$.

Let us now recall the definition of the arithmetical hierarchy of ω-languages, see for example [2, 45]. Let Σ be a finite alphabet. An ω-language $L \subseteq \Sigma^\omega$ belongs to the class Σ_n iff there exists a recursive relation $R_L \subseteq (\mathbb{N})^{n-1} \times \Sigma^\star$ such that $L = \{\sigma \in \Sigma^\omega \mid \exists a_1 \ldots Q_n a_n \quad (a_1, \ldots, a_{n-1}, \sigma[a_n + 1]) \in R_L\}$, where Q_i is one of the quantifiers $\forall$ or $\exists$ (not necessarily in an alternating order). An ω-language $L \subseteq \Sigma^\omega$ belongs to the class Π_n if and only if its complement $\Sigma^\omega - L$ belongs to the class Σ_n. The inclusion relations that hold between the classes Σ_n and Π_n are the same as for the corresponding classes of the Borel hierarchy and the classes Σ_n and Π_n are strictly included in the respective classes $\mathbf{\Sigma}^0_n$ and $\mathbf{\Pi}^0_n$ of the Borel hierarchy.

As in the case of the Borel hierarchy, projections of arithmetical sets (of the second Π-class) lead beyond the arithmetical hierarchy, to the analytical hierarchy of ω-languages. The first class of the analytical hierarchy of ω-languages is the (lightface) class Σ^1_1 of effective analytic sets. An ω-language $L \subseteq \Sigma^\omega$ belongs to the class Σ^1_1 if and only if there exists a recursive relation $R_L \subseteq (\mathbb{N}) \times \{0, 1\}^\star \times \Sigma^\star$ such that: $L = \{\sigma \in \Sigma^\omega \mid \exists \tau (\tau \in \{0, 1\}^\omega \wedge \forall n \exists m ((n, \tau[m], \sigma[m]) \in R_L))\}$. Thus an ω-language $L \subseteq \Sigma^\omega$ is in the class Σ^1_1 iff it is the projection of an ω-language over the alphabet $\{0, 1\} \times \Sigma$ which is in the class Π_2.

Kechris, Marker, and Sami proved in [46] that the supremum of the set of Borel ranks of (lightface) Π^1_1 (so also of lightface Σ^1_1) sets is the ordinal γ^1_2. This ordinal is precisely defined in [46]. It holds that $\omega_1^{\rm CK} < \gamma^1_2$, where $\omega_1^{\rm CK}$ is the first non-recursive ordinal, called the Church-Kleene ordinal.

Notice that it seems still unknown whether *every* non null ordinal $\gamma < \gamma^1_2$ is the Borel rank of a (lightface) Π^1_1 (or Σ^1_1) set. On the other hand it is known that for every ordinal $\gamma < \omega_1^{\rm CK}$ there exist some $\mathbf{\Sigma}^0_\gamma$-complete and $\mathbf{\Pi}^0_\gamma$-complete sets in the class Δ^1_1.

Recall that a Büchi Turing machine is just a Turing machine working on infinite inputs with a Büchi-like acceptance condition, and that the class of ω-languages accepted by Büchi Turing machines is the class Σ^1_1 [5, 2].

3.2. Wadge hierarchy

We now introduce the Wadge hierarchy, which is a great refinement of the Borel hierarchy defined via reductions by continuous functions, [47, 10].

Definition 3.3. (Wadge [10])
Let X, Y be two finite alphabets. For $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, L is said to be *Wadge reducible* to L' ($L \leq_W L'$) iff there exists a continuous function $f : X^\omega \to Y^\omega$, such that $L = f^{-1}(L')$. ω-languages L and L' are Wadge equivalent iff $L \leq_W L'$ and $L' \leq_W L$. This will be denoted by $L \equiv_W L'$. Moreover, we shall say that $L <_W L'$ iff $L \leq_W L'$ but not $L' \leq_W L$.
A set $L \subseteq X^\omega$ is said to be *self dual* iff $L \equiv_W L^-$, and otherwise it is said to be *non self dual*.

The relation $\leq_W$ is reflexive and transitive, and $\equiv_W$ is an equivalence relation. The *equivalence classes* of $\equiv_W$ are called *Wadge degrees*. The Wadge hierarchy WH is the class of Borel subsets of a set X^ω, where X is a finite set, equipped with $\leq_W$ and with $\equiv_W$.
For $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, if $L \leq_W L'$ and $L = f^{-1}(L')$ where f is a continuous function from X^ω into Y^ω, then f is called a continuous reduction of L to L'. Intuitively it means that L is less complicated than L' because to check whether $x \in L$ it suffices to check whether $f(x) \in L'$ where f is a continuous function. Hence the Wadge degree of an ω-language is a measure of its topological complexity.
Notice that in the above definition, we consider that a subset $L \subseteq X^\omega$ is given together with the alphabet X.
We can now define the *Wadge class* of a set L:

Definition 3.4. Let L be a subset of X^ω. The Wadge class of L is :

$$[L] =_{\text{df}} \{L' \mid L' \subseteq Y^\omega \text{ for a finite alphabet } Y \text{ and } L' \leq_W L\}.$$

Recall that each *Borel class* $\mathbf{\Sigma}^0_\alpha$ and $\mathbf{\Pi}^0_\alpha$ is a *Wadge class*. A set $L \subseteq X^\omega$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively $\mathbf{\Pi}^0_\alpha$)-*complete set* iff for any set $L' \subseteq Y^\omega$, L' is in $\mathbf{\Sigma}^0_\alpha$ (respectively $\mathbf{\Pi}^0_\alpha$) iff $L' \leq_W L$.

There is a close relationship between Wadge reducibility and games which we now introduce.

Definition 3.5. Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$. The Wadge game $W(L, L')$ is a game with perfect information between two players, player 1 who is in charge of L and player 2 who is in charge of L'. Player 1 first writes a letter $a_1 \in X$, then player 2 writes a letter $b_1 \in Y$, then player 1 writes a letter $a_2 \in X$, and so on. The two players alternatively write letters a_n of X for player 1 and b_n of Y for player 2. After ω steps, the player 1 has written an ω-word $a \in X^\omega$ and the player 2 has written an ω-word $b \in Y^\omega$. The player 2 is allowed to skip, even infinitely often, provided he really writes an ω-word in ω steps. The player 2 wins the play iff $[a \in L \leftrightarrow b \in L']$, i.e. iff :

$$[(a \in L \text{ and } b \in L') \text{ or } (a \notin L \text{ and } b \notin L' \text{ and } b \text{ is infinite})].$$

Recall that a strategy for player 1 is a function $\sigma : (Y \cup \{s\})^\star \to X$. And a strategy for player 2 is a function $f : X^+ \to Y \cup \{s\}$. The strategy σ is a winning strategy for player 1 iff he always wins a play when he uses the strategy σ, i.e. when the nth letter he writes is given by $a_n = \sigma(b_1 \cdots b_{n-1})$, where b_i is the letter written by player 2 at step i and $b_i = s$ if player 2 skips at step i. A winning strategy for player 2 is defined in a similar manner.

Martin's Theorem states that every Gale-Stewart game $G(X)$ (see [48]), with X a Borel set, is determined and this implies the following :

Theorem 3.6. (Wadge)
Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$ be two Borel sets, where X and Y are finite alphabets. Then the Wadge game $W(L, L')$ is determined: one of the two players has a winning strategy. And $L \leq_W L'$ iff the player 2 has a winning strategy in the game $W(L, L')$.

Theorem 3.7. (Wadge)
Up to the complement and $\equiv_W$, the class of Borel subsets of X^ω, for a finite alphabet X having at least two letters, is a well ordered hierarchy. There is an ordinal $|WH|$, called the length of the hierarchy, and a map d^0_W from WH onto $|WH| - \{0\}$, such that for all $L, L' \subseteq X^\omega$:
$d^0_W L < d^0_W L' \leftrightarrow L <_W L'$ and
$d^0_W L = d^0_W L' \leftrightarrow [L \equiv_W L'$ or $L \equiv_W L'^-]$.

The Wadge hierarchy of Borel sets of **finite rank** has length ${}^1\varepsilon_0$ where ${}^1\varepsilon_0$ is the limit of the ordinals α_n defined by $\alpha_1 = \omega_1$ and $\alpha_{n+1} = \omega_1^{\alpha_n}$ for n a non negative integer, ω_1 being the first non countable ordinal. Then ${}^1\varepsilon_0$ is the first fixed point of the ordinal exponentiation of base ω_1. The length of the Wadge hierarchy of Borel sets in $\mathbf{\Delta}^0_\omega = \mathbf{\Sigma}^0_\omega \cap \mathbf{\Pi}^0_\omega$ is the ω_1^{th} fixed point of the ordinal exponentiation of base ω_1, which is a much larger ordinal. The length of the whole Wadge hierarchy of Borel sets is a huge ordinal, with regard to the ω_1^{th} fixed point of the ordinal exponentiation of base ω_1. It is described in [10, 47] by the use of the Veblen functions.

4. Wadge degrees of ω-languages of Petri nets

We are firstly going to prove the following result.

Theorem 4.1. The Wadge hierarchy of the class $\mathbf{r}$-$\mathbf{BCL}(4)_\omega$ is equal to the Wadge hierarchy of the class $\mathbf{r}$-$\mathbf{CL}(1)_\omega$.

In order to prove this result, we first define a coding of ω-words over a finite alphabet Σ by ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ where A, B and 0 are new letters not in Σ.

We shall code an ω-word $x \in \Sigma^\omega$ by the ω-word $h(x)$ defined by

$$h(x) = A0x(1)B0^2x(2)A \cdots B0^{2n}x(2n)A0^{2n+1}x(2n+1)B \cdots$$

This coding defines a mapping $h : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$.
The function h is continuous because for all ω-words $x, y \in \Sigma^\omega$ and each positive integer n, it holds that $\delta(x, y) < 2^{-n} \to \delta(h(x), h(y)) < 2^{-n}$.

We are going to state Lemma 4.2. Before that, we just describe some important facts given by this lemma and its proof. The lemma provides, from a real time 1-counter Büchi automaton $\mathcal{A}$ accepting ω-words over the alphabet Σ, a construction of a 4-blind-counter Büchi automaton $\mathcal{B}$ reading ω-words over the alphabet $\Gamma = \Sigma \cup \{A, B, 0\}$ which is able in some sense to simulate the automaton $\mathcal{A}$. Actually the automaton $\mathcal{B}$ simulates the reading of an ω-word x by $\mathcal{A}$ only when $\mathcal{B}$ reads the specific ω-word

$$h(x) = A0x(1)B0^2x(2)A \cdots B0^{2n}x(2n)A0^{2n+1}x(2n+1)B \cdots$$

The reading by the automaton $\mathcal{B}$ of the ω-word $h(x)$ will provide a decomposition of the ω-word $h(x)$ of the following form:

$$\begin{aligned} y = {} & Au_1v_1x(1)Bu_2v_2x(2)Au_3v_3x(3)B\cdots \\ & \cdots Bu_{2n}v_{2n}x(2n)Au_{2n+1}v_{2n+1}x(2n+1)B\cdots \end{aligned}$$

where, for all integers $i \geq 1$, $u_i, v_i \in 0^\star$, $x(i) \in \Sigma$, $|u_1| = 0$. Then an accepting run of $\mathcal{B}$ on $h(x)$ will correspond to an accepting run of $\mathcal{A}$ on x. Moreover the successive values of the single counter of $\mathcal{A}$ during this run will be the integers $|u_n|$, $n \geq 1$. Then the automaton $\mathcal{B}$ will be able to determine, using a finite control component during the reading of the finite word u_n, whether $|u_n| = 0$, and thus to simulate the zero-tests of the automaton $\mathcal{A}$.

The proof of the following lemma will explain this in detail.

Lemma 4.2. Let $\mathcal{A}$ be a real time 1-counter Büchi automaton accepting ω-words over the alphabet Σ. Then one can construct a real time 4-blind-counter Büchi automaton $\mathcal{B}$ reading ω-words over the alphabet $\Gamma = \Sigma \cup \{A, B, 0\}$, such that $\mathrm{L}(\mathcal{A}) = h^{-1}(\mathrm{L}(\mathcal{B}))$, i.e. $\forall x \in \Sigma^\omega.\ h(x) \in \mathrm{L}(\mathcal{B}) \longleftrightarrow x \in \mathrm{L}(\mathcal{A})$.

Proof:
Let $\mathcal{A} = (K, \Sigma, \Delta, q_0, F)$ be a real time 1-counter Büchi automaton accepting ω-words over the alphabet Σ. We are going to explain informally the behaviour of the 4-blind-counter Büchi automaton $\mathcal{B}$ when reading an ω-word of the form $h(x)$, even if we are going to see that $\mathcal{B}$ may also accept some infinite words which do not belong to the range of h. Recall that $h(x)$ is of the form

$$h(x) = A0x(1)B0^2x(2)A\cdots B0^{2n}x(2n)A0^{2n+1}x(2n+1)B\cdots$$

Notice that in particular every ω-word in $h(\Sigma^\omega)$ is of the form:

$$y = A0^{n_1}x(1)B0^{n_2}x(2)A\cdots B0^{n_{2n}}x(2n)A0^{n_{2n+1}}x(2n+1)B\cdots$$

where for all $i \geq 1$, $n_i > 0$ is a positive integer, and $x(i) \in \Sigma$.

Moreover it is easy to see that the set of ω-words $y \in \Gamma^\omega$ which can be written in the above form is a regular ω-language $\mathcal{R} \subseteq \Gamma^\omega$, and thus we can assume, using a classical product construction (see for instance [3]), that the automaton $\mathcal{B}$ will only accept some ω-words of this form.

Now the reading by the automaton $\mathcal{B}$ of an ω-word of the above form

$$y = A0^{n_1}x(1)B0^{n_2}x(2)A\cdots B0^{n_{2n}}x(2n)A0^{n_{2n+1}}x(2n+1)B\cdots$$

will give a decomposition of the ω-word y of the following form:

$$\begin{aligned} y = {} & Au_1v_1x(1)Bu_2v_2x(2)Au_3v_3x(3)B\cdots \\ & \cdots Bu_{2n}v_{2n}x(2n)Au_{2n+1}v_{2n+1}x(2n+1)B\cdots \end{aligned}$$

where, for all integers $i \geq 1$, $u_i, v_i \in 0^\star$, $x(i) \in \Sigma$, $|u_1| = 0$.

The automaton $\mathcal{B}$ will use its four *blind* counters, which we denote $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$, in the following way. Recall that the automaton $\mathcal{B}$ being non-deterministic, we do not describe the unique run of $\mathcal{B}$ on y, but the general case of a possible run.

At the beginning of the run, the value of each of the four counters is equal to zero. Then the counter $\mathcal{C}_1$ is increased of $|u_1|$ when reading u_1, i.e. the counter $\mathcal{C}_1$ is actually not increased since $|u_1| = 0$ and the finite control is here used to check this. Then the counter $\mathcal{C}_2$ is increased of 1 for each letter 0 of v_1 which is read until the automaton reads the letter $x(1)$ and then the letter B. Notice that at this time the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ are still equal to zero. Then the behaviour of the automaton $\mathcal{B}$ when reading the next segment $0^{n_2}x(2)A$ is as follows. The counter $\mathcal{C}_1$ is firstly decreased of 1 for each letter 0 read, when reading k_2 letters 0, where $k_2 \geq 0$ (notice that here $k_2 = 0$ because the value of the counter $\mathcal{C}_1$ being equal to zero, it cannot decrease under 0). Then the counter $\mathcal{C}_2$ is decreased of 1 for each letter 0 read, and next the automaton has to read one more letter 0, leaving unchanged the counters $\mathcal{C}_1$ and $\mathcal{C}_2$, before reading the letter $x(2)$. The end of the decreasing mode of $\mathcal{C}_1$ coincide with the beginning of the decreasing mode of $\mathcal{C}_2$, and this change may occur in a *non-deterministic way* (because the automaton $\mathcal{B}$ cannot check whether the value of $\mathcal{C}_1$ is equal to zero). Now we describe the behaviour of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ when reading the segment $0^{n_2}x(2)A$. Using its finite control, the automaton $\mathcal{B}$ has checked that $|u_1| = 0$, and then if there is a transition of the automaton $\mathcal{A}$ such that $x(1) : (q_0, |u_1|) \mapsto_{\mathcal{A}} (q_1, |u_1| + N_1)$ then the counter $\mathcal{C}_3$ is increased of 1 for each letter 0 read, during the reading of the $k_2 + N_1$ first letters 0 of 0^{n_2}, where k_2 is described above as the number of which the counter $\mathcal{C}_1$ has been decreased. This determines u_2 by $|u_2| = k_2 + N_1$ and then the counter $\mathcal{C}_4$ is increased by 1 for each letter 0 read until $\mathcal{B}$ reads $x(2)$, and this determines v_2. Notice that the automaton $\mathcal{B}$ keeps in its finite control the memory of the state q_1 of the automaton $\mathcal{A}$, and that, after having read the segment $0^{n_2} = u_2v_2$, the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ are respectively $|\mathcal{C}_3| = |u_2| = k_2 + N_1$ and $|\mathcal{C}_4| = |v_2| = n_2 - (|u_2|)$.

Now the run will continue. Notice that generally when reading a segment $B0^{n_{2n}}x(2n)A$ the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ will successively decrease when reading the first $(n_{2n} - 1)$ letters 0 and then will remain unchanged when reading the last letter 0, and the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ will successively increase, when reading the (n_{2n}) letters 0. Again the end of the decreasing mode of $\mathcal{C}_1$ coincide with the beginning of the decreasing mode of $\mathcal{C}_2$, and this change may occur in a *non-deterministic way*. But the automaton has kept in its finite control whether $|u_{2n-1}| = 0$ or not and also a state q_{2n-2} of the automaton $\mathcal{A}$. Now, if there is a transition of the automaton $\mathcal{A}$ such that $x(2n-1) : (q_{2n-2}, |u_{2n-1}|) \mapsto_{\mathcal{A}} (q_{2n-1}, |u_{2n-1}| + N_{2n-1})$ for some integer $N_{2n-1} \in \{-1; 0, 1\}$, and the counter $\mathcal{C}_1$ is decreased of 1 for each letter 0 read, when reading k_{2n} first letters 0 of $0^{n_{2n}}$, then the counter $\mathcal{C}_3$ is increased of 1 for each letter 0 read, during the reading of the $k_{2n} + N_{2n-1}$ first letters 0 of $0^{n_{2n}}$, and next the counter $\mathcal{C}_4$ is increased by 1 for each letter 0 read until $\mathcal{B}$ reads $x(2n)$, and this determines v_{2n}. Then after having read the segment $0^{n_{2n}} = u_{2n}v_{2n}$, the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ have respectively increased of $|u_{2n}| = k_{2n} + N_{2n-1}$ and $|v_{2n}| = n_{2n} - |u_{2n}|$. Notice that one cannot ensure that, after the reading of $0^{n_{2n}} = u_{2n}v_{2n}$, the exact values of these counters are $|\mathcal{C}_3| = |u_{2n}| = k_{2n} + N_{2n-1}$ and $|\mathcal{C}_4| = |v_{2n}| = n_{2n} - |u_{2n}|$. Actually this is due to the fact that one cannot ensure that the values of $\mathcal{C}_3$ and $\mathcal{C}_4$ are equal to zero at the beginning of the reading of the segment $B0^{n_{2n}}x(2n)A$ although we will see this is true and important in the particular case of a word of the form $y = h(x)$.

The run will continue in a similar manner during the reading of the next segment $A0^{n_{2n+1}}x(2n+1)B$, but here the role of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ on one side, and of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ on the other side, will be interchanged. More precisely the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ will successively decrease when reading the first $(n_{2n+1}-1)$ letters 0 and then will remain unchanged when reading the last letter 0, and the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ will successively increase, when reading the (n_{2n+1}) letters 0. The end of the decreasing mode of $\mathcal{C}_3$ coincide with the beginning of the decreasing mode of $\mathcal{C}_4$, and this change may occur in a *non-deterministic way*. But the automaton has kept in its finite control whether $|u_{2n}| = 0$ or not and also a state q_{2n-1} of the automaton $\mathcal{A}$. Now, if there is a transition of the automaton $\mathcal{A}$ such that $x(2n) : (q_{2n-1}, |u_{2n}|) \mapsto_{\mathcal{A}} (q_{2n}, |u_{2n}| + N_{2n})$ for some integer $N_{2n} \in \{-1; 0, 1\}$, and the counter $\mathcal{C}_3$ is decreased of 1 for each letter 0 read, when reading k_{2n+1} first letters 0 of $0^{n_{2n+1}}$, then the counter $\mathcal{C}_1$ is increased of 1 for each letter 0 read, during the reading of the $k_{2n+1} + N_{2n}$ first letters 0 of $0^{n_{2n+1}}$, and next the counter $\mathcal{C}_2$ is increased by 1 for each letter 0 read until $\mathcal{B}$ reads $x(2n+1)$, and this determines v_{2n+1}. Then after having read the segment $0^{n_{2n+1}} = u_{2n+1}v_{2n+1}$, the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ have respectively increased of $|u_{2n+1}| = k_{2n+1} + N_{2n}$ and $|v_{2n+1}| = n_{2n+1} - |u_{2n+1}|$. Notice that again one cannot ensure that, after the reading of $0^{n_{2n+1}} = u_{2n+1}v_{2n+1}$, the exact values of these counters are $|\mathcal{C}_1| = |u_{2n+1}| = k_{2n+1} + N_{2n}$ and $|\mathcal{C}_2| = |v_{2n+1}| = n_{2n+1} - |u_{2n+1}|$. This is due to the fact that one cannot ensure that the values of $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to zero at the beginning of the reading of the segment $A0^{n_{2n+1}}x(2n+1)B$ although we will see this is true and important in the particular case of a word of the form $y = h(x)$.

The run then continues in the same way if it is possible and in particular if there is no blocking due to the fact that one of the counters of the automaton $\mathcal{B}$ would have a negative value.

Now an ω-word $y \in \mathcal{R} \subseteq \Gamma^\omega$ of the above form will be accepted by the automaton $\mathcal{B}$ if there is such an infinite run for which a final state $q_f \in F$ of the automaton $\mathcal{A}$ has been stored infinitely often in the finite control of $\mathcal{B}$ in the way which has just been described above.

We now consider the particular case of an ω-word of the form $y = h(x)$, for some $x \in \Sigma^\omega$. Let then

$y = h(x) = A0x(1)B0^2x(2)A0^3x(3)B \cdots B0^{2n}x(2n)A0^{2n+1}x(2n+1)B \cdots$

We are going to show that, if y is accepted by the automaton $\mathcal{B}$, then $x \in \mathrm{L}(\mathcal{A})$. Let us consider a run of the automaton $\mathcal{B}$ on y as described above and which is an accepting run. We first show by induction on $n \geq 1$, that after having read an initial segment of the form

$$A0x(1)B0^2x(2)A \cdots A0^{2n-1}x(2n-1)B,$$

the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ are equal to zero, and the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfy $|\mathcal{C}_1| + |\mathcal{C}_2| = 2n - 1$. And similarly after having read an initial segment of the form

$$A0x(1)B0^2x(2)A \cdots B0^{2n}x(2n)A,$$

the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to zero, and the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ satisfy $|\mathcal{C}_3| + |\mathcal{C}_4| = 2n$.

For $n = 1$, we have seen that after having read the initial segment $A0x(1)B$, the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ will be respectively 0 and $|v_1|$ and here $|v_1| = 1$ and thus $|\mathcal{C}_1| + |\mathcal{C}_2| = 1$. On the other hand the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ have not yet increased so that the value of each of these counters is

equal to zero. During the reading of the segment 0^2 of $0^2x(2)A$ the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ successively decrease. But here $\mathcal{C}_1$ cannot decrease (with the above notations, it holds that $k_2 = 0$) so $\mathcal{C}_2$ must decrease of 1 because after the decreasing mode the automaton $\mathcal{B}$ must read a last letter 0 without decreasing the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ and then the letter $x(2) \in \Sigma$. Thus after having read $0^2x(2)A$ the values of $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to zero. Moreover the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ had their values equal to zero at the beginning of the reading of $0^2x(2)A$ and they successively increase during the reading of 0^2 and they remain unchanged during the reading of $x(2)A$ so that their values satisfy $|\mathcal{C}_3| + |\mathcal{C}_4| = 2$ after the reading of $0^2x(2)A$.

Assume now that for some integer $n > 1$ the claim is proved for all integers $k < n$ and let us prove it for the integer n. By induction hypothesis we know that at the beginning of the reading of the segment $A0^{2n-1}x(2n-1)B$ of y, the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to zero, and the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ satisfy $|\mathcal{C}_3| + |\mathcal{C}_4| = 2n-2$. When reading the $(2n-2)$ first letters 0 of $A0^{2n-1}x(2n-1)B$ the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ successively decrease and they must decrease completely because after there must remain only one letter 0 to be read by $\mathcal{B}$ before the letter $x(2n-1)$. Therefore after the reading of $A0^{2n-1}x(2n-1)B$ the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ are equal to zero. And since the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ are equal to zero before the reading of $0^{2n-1}x(2n-1)B$ and these counters successively increase during the reading of 0^{2n-1}, their values satisfy $|\mathcal{C}_1| + |\mathcal{C}_2| = 2n-1$ after the reading of $A0^{2n-1}x(2n-1)B$. We can reason in a very similar manner for the reading of the next segment $B0^{2n}x(2n)A$, the role of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ on one side, and of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ on the other side, being simply interchanged. This ends the proof of the claim by induction on n.

It is now easy to see by induction that for each integer $n \geq 2$, it holds that $k_n = |u_{n-1}|$. Then, since with the above notations we have $|u_{n+1}| = k_{n+1} + N_n = |u_n| + N_n$, and there is a transition of the automaton $\mathcal{A}$ such that $x(n) : (q_{n-1}, |u_n|) \mapsto_{\mathcal{A}} (q_n, |u_n| + N_n)$ for $N_n \in \{-1; 0, 1\}$, it holds that $x(n) : (q_{n-1}, |u_n|) \mapsto_{\mathcal{A}} (q_n, |u_{n+1}|)$. Therefore the sequence $(q_i, |u_i|)_{i\geq 0}$ is an accepting run of the automaton $\mathcal{A}$ on the ω-word x and $x \in \mathrm{L}(\mathcal{A})$. Notice that the state q_0 of the sequence $(q_i)_{i\geq 0}$ is also the initial state of $\mathcal{A}$.

Conversely, it is easy to see that if $x \in \mathrm{L}(\mathcal{A})$ then there exists an accepting run of the automaton $\mathcal{B}$ on the ω-word $h(x)$ and $h(x) \in \mathrm{L}(\mathcal{B})$. □

The above Lemma 4.2 shows that, given a real time 1-counter (with zero-test) Büchi automaton $\mathcal{A}$ accepting ω-words over the alphabet Σ, one can construct a real time 4-blind-counter Büchi automaton $\mathcal{B}$ which can simulate the 1-counter automaton $\mathcal{A}$ on the code $h(x)$ of the word x. On the other hand, we cannot describe precisely the ω-words which are accepted by $\mathcal{B}$ but are not in the set $h(\Sigma^\omega)$. However we can see that all these words have a special shape, as stated by the following lemma.

Lemma 4.3. Let $\mathcal{A}$ be a real time 1-counter Büchi automaton accepting ω-words over the alphabet Σ, and let $\mathcal{B}$ be the real time 4-blind-counter Büchi automaton reading words over the alphabet $\Gamma = \Sigma \cup \{A, B, 0\}$ which is constructed in the proof of Lemma 4.2. Let $y \in \mathrm{L}(\mathcal{B}) \setminus h(\Sigma^\omega)$ being of the following form

$$y = A0^{n_1}x(1)B0^{n_2}x(2)A0^{n_3}x(3)B \cdots B0^{n_{2n}}x(2n)A0^{n_{2n+1}}x(2n+1)B \cdots$$

and let i_0 be the smallest integer i such that $n_i \neq i$. Then it holds that either $i_0 = 1$ or $n_{i_0} < i_0$.

Proof:
Assume first that $y \in \mathrm{L}(\mathcal{B}) \setminus h(\Sigma^\omega)$ is of the following form

$$y = A0^{n_1}x(1)B0^{n_2}x(2)A \cdots B0^{n_{2n}}x(2n)A0^{n_{2n+1}}x(2n+1)B \cdots$$

and that the smallest integer i such that $n_i \neq i$ is an even integer $i_0 > 1$. Consider an infinite accepting run of $\mathcal{B}$ on y. It follows from the proof of the above Lemma 4.2 that after the reading of the initial segment

$$A0^{n_1}x(1)B0^{n_2}x(2)A \cdots A0^{i_0-1}x(i_0-1)B$$

the values of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ are equal to zero, and the values of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ satisfy $|\mathcal{C}_1| + |\mathcal{C}_2| = i_0 - 1$. Thus since the two counters must successively decrease during the next $n_{i_0} - 1$ letters 0, it holds that $n_{i_0} - 1 \leq i_0 - 1$ because otherwise either $\mathcal{C}_1$ or $\mathcal{C}_2$ would block. Therefore $n_{i_0} < i_0$ since $n_{i_0} \neq i_0$ by definition of i_0. The reasoning is very similar in the case of an odd integer i_0, the role of the counters $\mathcal{C}_1$ and $\mathcal{C}_2$ on one side, and of the counters $\mathcal{C}_3$ and $\mathcal{C}_4$ on the other side, being simply interchanged. □

Let $\mathcal{L} \subseteq \Gamma^\omega$ be the ω-language containing the ω-words over Γ which belong to one of the following ω-languages.

- $\mathcal{L}_1$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ which have not any initial segment in $A \cdot 0 \cdot \Sigma \cdot B$.
- $\mathcal{L}_2$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ which contain a segment of the form $B \cdot 0^n \cdot a \cdot A \cdot 0^m \cdot b$ or of the form $A \cdot 0^n \cdot a \cdot B \cdot 0^m \cdot b$ for some letters $a, b \in \Sigma$ and some positive integers $m \leq n$.

Lemma 4.4. The ω-language $\mathcal{L}$ is accepted by a (non-deterministic) real-time 1-blind counter Büchi automaton.

Proof:
First, it is easy to see that $\mathcal{L}_1$ is in fact a regular ω-language, and thus it is also accepted by a real-time 1-blind counter Büchi automaton (even without active counter). On the other hand it is also easy to construct a real time 1-blind counter Büchi automaton accepting the ω-language $\mathcal{L}_2$. The class of ω-languages accepted by *non-deterministic* real time 1-blind counter Büchi automata being closed under finite union in an effective way, one can construct a real time 1-blind counter Büchi automaton accepting $\mathcal{L}$. □

Lemma 4.5. Let $\mathcal{A}$ be a real time 1-counter Büchi automaton accepting ω-words over the alphabet Σ. Then one can construct a real time 4-blind counter Büchi automaton $\mathcal{P}_\mathcal{A}$ such that $\mathrm{L}(\mathcal{P}_\mathcal{A}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$.

Proof:
Let $\mathcal{A}$ be a real time 1-counter Büchi automaton accepting ω-words over Σ. We have seen in the proof of Lemma 4.2 that one can construct a real time 4-blind counter Büchi automaton $\mathcal{B}$ reading words

over the alphabet $\Gamma = \Sigma \cup \{A, B, 0\}$, such that $\mathrm{L}(\mathcal{A}) = h^{-1}(\mathrm{L}(\mathcal{B}))$, i.e. $\forall x \in \Sigma^\omega \quad h(x) \in \mathrm{L}(\mathcal{B}) \longleftrightarrow x \in \mathrm{L}(\mathcal{A})$. Moreover By Lemma 4.3 it holds that $\mathrm{L}(\mathcal{B}) \setminus h(\Sigma^\omega) \subseteq \mathcal{L}$ and thus $h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L} = \mathrm{L}(\mathcal{B}) \cup \mathcal{L}$. But By Lemma 4.4 the ω-language $\mathcal{L}$ is accepted by a (non-deterministic) real-time 1-blind counter Büchi automaton, hence also by a real-time 4-blind counter Büchi automaton. The class of ω-languages accepted by (non-deterministic) real-time 4-blind counter Büchi automata is closed under finite union in an effective way, and thus one can construct a real time 4-blind counter Büchi automaton $\mathcal{P}_\mathcal{A}$ such that $\mathrm{L}(\mathcal{P}_\mathcal{A}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$. □

We are now going to prove that if $\mathrm{L}(\mathcal{A}) \subseteq \Sigma^\omega$ is accepted by a real time 1-counter automaton $\mathcal{A}$ with a Büchi acceptance condition then $\mathrm{L}(\mathcal{P}_\mathcal{A}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$ will have the same Wadge degree as the ω-language $\mathrm{L}(\mathcal{A})$, except for some very simple cases.

We first notice that $h(\Sigma^\omega)$ is a closed subset of Γ^ω. Indeed it is the image of the compact set Σ^ω by the continuous function h, and thus it is a compact hence also closed subset of $\Gamma^\omega = (\Sigma \cup \{A, B, 0\})^\omega$. Thus its complement $h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$ is an open subset of Γ^ω. Moreover the set $\mathcal{L}$ is an open subset of Γ^ω, as it can be easily seen from its definition and one can easily define, from the definition of the ω-language $\mathcal{L}$, a finitary language $V \subseteq \Gamma^\star$ such that $\mathcal{L} = V \cdot \Gamma^\omega$. We shall also denote $\mathcal{L}' = h(\Sigma^\omega)^- \setminus \mathcal{L}$ so that Γ^ω is the disjoint union $\Gamma^\omega = h(\Sigma^\omega) \cup \mathcal{L} \cup \mathcal{L}'$. Notice that $\mathcal{L}'$ is the difference of the two open sets $h(\Sigma^\omega)^-$ and $\mathcal{L}$.

We now wish to return to the proof of the above Theorem 4.1 stating that the Wadge hierarchy of the class $\mathbf{r\text{-}BCL}(4)_\omega$ is equal to the Wadge hierarchy of the class $\mathbf{r\text{-}CL}(1)_\omega$.

To prove this result we firstly consider non self dual Borel sets. We recall the definition of Wadge degrees introduced by Duparc in [47] and which is a slight modification of the previous one.

Definition 4.6. (a) $d_w(\emptyset) = d_w(\emptyset^-) = 1$

(b) $d_w(L) = \sup\{d_w(L') + 1 \mid L' \text{ non self dual and } L' <_W L\}$
(for either L self dual or not, $L >_W \emptyset$).

Wadge and Duparc used the operation of sum of sets of infinite words which has as counterpart the ordinal addition over Wadge degrees.

Definition 4.7. (Wadge, see [10, 47])
Assume that $X \subseteq Y$ are two finite alphabets, $Y - X$ containing at least two elements, and that $\{X_+, X_-\}$ is a partition of $Y - X$ in two non empty sets. Let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, then $L' + L =_{\mathrm{df}} L \cup \{u \cdot a \cdot \beta \mid u \in X^\star,\ (a \in X_+ \textit{ and } \beta \in L') \textit{ or } (a \in X_- \textit{ and } \beta \in L'^-)\}$

This operation is closely related to the *ordinal sum* as it is stated in the following:

Theorem 4.8. (Wadge, see [10, 47])
Let $X \subseteq Y$, $Y - X$ containing at least two elements, $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$ be non self dual Borel sets. Then $(L + L')$ is a non self dual Borel set and $d_w(L' + L) = d_w(L') + d_w(L)$.

A player in charge of a set $L' + L$ in a Wadge game is like a player in charge of the set L but who can, at any step of the play, erase his previous play and choose to be this time in charge of L' or of L'^-. Notice that he can do this only one time during a play.

The following lemma was proved in [20]. Notice that below the empty set is considered as an ω-language over an alphabet Δ such that $\Delta - \Sigma$ contains at least two elements.

Lemma 4.9. Let $L \subseteq \Sigma^\omega$ be a non self dual Borel set such that $d_w(L) \geq \omega$. Then it holds that $L \equiv_W \emptyset + L$.

We can now prove the following lemma.

Lemma 4.10. Let $L \subseteq \Sigma^\omega$ be a non self dual Borel set accepted by a real time 1-counter Büchi automaton $\mathcal{A}$. Then there is an ω-language L' accepted by a real time 4-blind counter Büchi automaton such that $L \equiv_W L'$.

Proof:
Recall first that there are regular ω-languages of every finite Wadge degree, [2, 12]. These regular ω-languages are Boolean combinations of open sets, and they obviously belong to the class $\mathbf{r}\text{-}\mathbf{BCL}(4)_\omega$ since every regular ω-language belongs to this class.

So we have only to consider the case of non self dual Borel sets of Wadge degrees greater than or equal to ω.

Let then $L = \mathrm{L}(\mathcal{A}) \subseteq \Sigma^\omega$ be a non self dual Borel set, accepted by a real time 1-counter Büchi automaton $\mathcal{A}$, such that $d_w(L) \geq \omega$. By Lemma 4.5, $\mathrm{L}(\mathcal{P}_\mathcal{A}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$ is accepted by a a real time 4-blind counter Büchi automaton $\mathcal{P}_\mathcal{A}$, where the mapping $h : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$ is defined, for $x \in \Sigma^\omega$, by:

$$h(x) = A0x(1)B0^2x(2)A0^3x(3)B \cdots B0^{2n}x(2n)A0^{2n+1}x(2n+1)B \cdots$$

We set $L' = \mathrm{L}(\mathcal{P}_\mathcal{A})$ and we now prove that $L' \equiv_W L$.

Firstly, it is easy to see that the function h is a continuous reduction of L to L' and thus $L \leq_W L'$.

To prove that $L' \leq_W L$, it suffices to prove that $L' \leq_W \emptyset + (\emptyset + L)$ because Lemma 4.9 states that $\emptyset + L \equiv_W L$, and thus also $\emptyset + (\emptyset + L) \equiv_W L$. Consider the Wadge game $W(L', \emptyset + (\emptyset + L))$. Player 2 has a winning strategy in this game which we now describe.

As long as Player 1 remains in the closed set $h(\Sigma^\omega)$ (this means that the word written by Player 1 is a prefix of some infinite word in $h(\Sigma^\omega)$) Player 2 essentially copies the play of player 1 except that Player 2 skips when player 1 writes a letter not in Σ. He continues forever with this strategy if the word written by player 1 is always a prefix of some ω-word of $h(\Sigma^\omega)$. Then after ω steps Player 1 has written an ω-word $h(x)$ for some $x \in \Sigma^\omega$, and Player 2 has written x. So in that case $h(x) \in L'$ iff $x \in \mathrm{L}(\mathcal{A})$ iff $x \in \emptyset + (\emptyset + L)$.

But if at some step of the play, Player 1 "goes out of" the closed set $h(\Sigma^\omega)$ because the word he has now written is not a prefix of any ω-word of $h(\Sigma^\omega)$, then Player 1 "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$ and will stay in this set. Two cases may now appear.

First case. When Player 1 "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, he actually enters in the open set $\mathcal{L} = V \cdot \Gamma^\omega$ (this means that Player 1 has written an initial segment in V). Then the final word written by Player 1 will surely be inside L'. Player 2 can now write a letter of $\Delta - \Sigma$ in such a way that he is now like a player in charge of the whole set and he can now writes an ω-word u so that his final ω-word will be inside $\emptyset + L$, and also inside $\emptyset + (\emptyset + L)$. Thus Player 2 wins this play too.

Second case. When Player 1 "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, he does not enter in the open set $\mathcal{L} = V \cdot \Gamma^\omega$. Then Player 2, being first like a player in charge of the set $(\emptyset + L)$, can write a letter of $\Delta - \Sigma$ in such a way that he is now like a player in charge of the empty set and he can now continue, writing an ω-word u. If Player 1 never enters in the open set $\mathcal{L} = V \cdot \Gamma^\omega$ then the final word written by Player 1 will be in $\mathcal{L}'$ and thus surely outside L', and the final word written by Player 2 will be outside the empty set. So in that case Player 2 wins this play too. If at some step of the play Player 1 enters in the open set $\mathcal{L} = V \cdot \Gamma^\omega$ then his final ω-word will be surely in L'. In that case Player 1, in charge of the set $\emptyset + (\emptyset + L)$, can again write an extra letter and choose to be in charge of the whole set and he can now write an ω-word v so that his final ω-word will be inside $\emptyset + (\emptyset + L)$. Thus Player 2 wins this play too.

Finally we have proved that $L \leq_W L' \leq_W L$ thus it holds that $L' \equiv_W L$. This ends the proof. □

End of Proof of Theorem 4.1.

Let $L \subseteq \Sigma^\omega$ be a Borel set accepted by a real time 1-counter Büchi automaton $\mathcal{A}$. If the Wadge degree of L is finite, it is well known that it is Wadge equivalent to a regular ω-language, hence also to an ω-language in the class $\mathbf{r\text{-}BCL}(4)_\omega$. If L is non self dual and its Wadge degree is greater than or equal to ω, then we know from Lemma 4.10 that there is an ω-language L' accepted by a a real time 4-blind counter Büchi automaton such that $L \equiv_W L'$.

It remains to consider the case of self dual Borel sets. The alphabet Σ being finite, a self dual Borel set L is always Wadge equivalent to a Borel set in the form $\Sigma_1 \cdot L_1 \cup \Sigma_2 \cdot L_2$, where (Σ_1, Σ_2) form a partition of Σ, and $L_1, L_2 \subseteq \Sigma^\omega$ are non self dual Borel sets such that $L_1 \equiv_W L_2^-$. Moreover L_1 and L_2 can be taken in the form $L_{(u_1)} = u_1 \cdot \Sigma^\omega \cap L$ and $L_{(u_2)} = u_2 \cdot \Sigma^\omega \cap L$ for some $u_1, u_2 \in \Sigma^\star$, see [16]. So if $L \subseteq \Sigma^\omega$ is a self dual Borel set accepted by a real time 1-counter Büchi automaton then $L \equiv_W \Sigma_1 \cdot L_1 \cup \Sigma_2 \cdot L_2$, where (Σ_1, Σ_2) form a partition of Σ, and $L_1, L_2 \subseteq \Sigma^\omega$ are non self dual Borel sets accepted by real time 1-counter Büchi automata. We have already proved that there is an ω-language L_1' in the class $\mathbf{r\text{-}BCL}(4)_\omega$ such that $L_1' \equiv_W L_1$ and an ω-language L_2' in the class $\mathbf{r\text{-}BCL}(4)_\omega$ such that $L_2'^- \equiv_W L_2$. Thus $L \equiv_W \Sigma_1 \cdot L_1 \cup \Sigma_2 \cdot L_2 \equiv_W \Sigma_1 \cdot L_1' \cup \Sigma_2 \cdot L_2'$ and $\Sigma_1 \cdot L_1' \cup \Sigma_2 \cdot L_2'$ is an ω-language in the class $\mathbf{r\text{-}BCL}(4)_\omega$.

The reverse direction is immediate: if $L \subseteq \Sigma^\omega$ is a Borel set accepted by a 4-blind counter Büchi automaton $\mathcal{A}$, then it is also accepted by a Büchi Turing machine and thus by [20, Theorem 25] there exists a real time 1-counter Büchi automaton $\mathcal{B}$ such that $\mathrm{L}(\mathcal{A}) \equiv_W \mathrm{L}(\mathcal{B})$.

This concludes the proof of Theorem 4.1.

Recall that, for each non-null countable ordinal α, the $\mathbf{\Sigma}^0_\alpha$-complete sets (respectively, the $\mathbf{\Pi}^0_\alpha$-complete sets) form a single Wadge degree. Thus we can infer the following result from the above Theorem 4.1 and from the results of [20, 46].

Corollary 4.11. For each non-null recursive ordinal $\alpha < \omega_1^{\mathrm{CK}}$ there exist some $\mathbf{\Sigma}^0_\alpha$-complete and some $\mathbf{\Pi}^0_\alpha$-complete ω-languages in the class $\mathbf{r\text{-}BCL}(4)_\omega$. And the supremum of the set of Borel ranks of ω-languages in the class $\mathbf{r\text{-}BCL}(4)_\omega$ is the ordinal γ^1_2, which is precisely defined in [46].

We have only considered Borel sets in the above Theorem 4.1. However we know that there also exist some non-Borel ω-languages accepted by real time 1-counter Büchi automata, and even some $\mathbf{\Sigma}^1_1$-complete ones, [49].

By Lemma 4.7 of [50] the conclusion of the above Lemma 4.9 is also true if L is assumed to be an analytic but non-Borel set.

Lemma 4.12. ([50])
Let $L \subseteq \Sigma^\omega$ be an analytic but non-Borel set. Then $L \equiv_W \emptyset + L$.

Next the proof of the above Lemma 4.10 can be adapted to the case of an analytic but non-Borel set, and we can state the following result.

Theorem 4.13. Let $L \subseteq \Sigma^\omega$ be an analytic but non-Borel set accepted by a real time 1-counter Büchi automaton $\mathcal{A}$. Then there is an ω-language L' accepted by a real time 4-blind counter Büchi automaton such that $L \equiv_W L'$.

Proof:
It is very similar to the proof of the above Lemma 4.10, using Lemma 4.12 instead of the above Lemma 4.9. □

Remark 4.14. Using Lemma 4.12 instead of the above Lemma 4.9, the proofs of [20] can also be adapted to the case of a non-Borel set to show that for every effective analytic but non-Borel set $L \subseteq \Sigma^\omega$, where Σ is a finite alphabet, there exists an ω-language L' in $\mathbf{r\text{-}CL}(1)_\omega$ such that $L' \equiv_W L$, and thus also, by Theorem 4.13, an ω-language L'' accepted by a real time 4-blind counter Büchi automaton such that $L \equiv_W L''$.

This implies in particular the existence of a $\mathbf{\Sigma}^1_1$-complete, hence non Borel, ω-language accepted by a real-time 4-blind-counter Büchi automaton.

Corollary 4.15. There exists a $\mathbf{\Sigma}^1_1$-complete ω-language accepted by a 4-blind-counter automaton.

Notice that if we assume the axiom of $\mathbf{\Sigma}^1_1$-determinacy, then any set which is analytic but not Borel is $\mathbf{\Sigma}^1_1$-complete, see [48], and thus there is only one more Wadge degree (beyond Borel sets) containing $\mathbf{\Sigma}^1_1$-complete sets. On the other hand, if the axiom of (effective) Σ^1_1-determinacy does not hold, then there exist some effective analytic sets which are neither Borel nor $\mathbf{\Sigma}^1_1$-complete. Recall that **ZFC** is the commonly accepted axiomatic framework for Set Theory in which all usual mathematics can be developed.

Corollary 4.16. It is consistent with **ZFC** that there exist some ω-languages of Petri nets in the class $\mathbf{r\text{-}BCL}(4)_\omega$ which are neither Borel nor $\mathbf{\Sigma}^1_1$-complete.

Proof:
Recall that **ZFC** is the commonly accepted axiomatic framework for Set Theory in which all usual mathematics can be developed. The determinacy of Gale-Stewart games $G(A)$, where A is an (effective) analytic set, denoted **Det**(Σ^1_1), is not provable in **ZFC**; Martin and Harrington have proved that it is a large cardinal assumption equivalent to the existence of a particular real, called the real $0^\sharp$, see [51, page 637]. It is also known that the determinacy of (effective) analytic Gale-Stewart games is equivalent to the determinacy of (effective) analytic Wadge games, denoted **W-Det**(Σ^1_1), see [52].

It is known that, if **ZFC** is consistent, then there is a model of **ZFC** in which the determinacy of (effective) analytic Gale-Stewart games, and thus also the determinacy of (effective) analytic Wadge games, do not hold. It follows from [53, Theorem 4.3] that in such a model of **ZFC** there exists an effective analytic set which is neither Borel nor $\mathbf{\Sigma}^1_1$-complete. The result now follows from Theorem 4.13 and Remark 4.14. □

We can now get an amazing result on ω-languages of Petri nets from the following previous one which was obtained in [54]. Recall that in a model **V** of **ZFC**, we denote by **L** the class of constructible sets of **V** which induces a submodel of **ZFC**. The axiom **V=L** expresses that "every set is constructible" and it is consistent with **ZFC**. The ordinal $\omega_1^{\mathbf{L}}$ denotes the first uncountable ordinal in the model **L**. It is consistent that $\omega_1^{\mathbf{L}} = \omega_1$ since this is true in the model **L**. But it is also consistent with **ZFC** that $\omega_1^{\mathbf{L}} < \omega_1$. We refer the interested reader to [54], and to classical textbooks on set theory like [51] for more information about these notions.

The following result was obtained in [54]:

Theorem 4.17. One can construct a real-time 1-counter Büchi automaton $\mathcal{A}$ such that the topological complexity of the ω-language $\mathrm{L}(\mathcal{A})$ is not determined by the axiomatic system **ZFC**. Indeed it holds that:

1. (**ZFC + V=L**). The ω-language $\mathrm{L}(\mathcal{A})$ is analytic but not Borel.
2. (**ZFC** + $\omega_1^{\mathbf{L}} < \omega_1$). The ω-language $\mathrm{L}(\mathcal{A})$ is a $\mathbf{\Pi}^0_2$-set.

We can now easily get the following result:

Theorem 4.18. One can construct a 4-blind counter Büchi automaton $\mathcal{B}$ such that the topological complexity of the ω-language $\mathrm{L}(\mathcal{B})$ is not determined by the axiomatic system **ZFC**. Indeed it holds that:

1. (**ZFC + V=L**). The ω-language $\mathrm{L}(\mathcal{B})$ is analytic but not Borel.
2. (**ZFC** + $\omega_1^{\mathbf{L}} < \omega_1$). The ω-language $\mathrm{L}(\mathcal{B})$ is a $\mathbf{\Pi}^0_2$-set.

Proof:
It follows directly from Theorem 4.17 and from the construction of the real time 4-blind counter Büchi automaton $\mathcal{P}_{\mathcal{A}}$ such that $\mathrm{L}(\mathcal{P}_{\mathcal{A}}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$, which was described above. It then suffices to take $\mathcal{B} = \mathcal{P}_{\mathcal{A}}$, where $\mathcal{A}$ is the real-time 1-counter Büchi automaton constructed in the proof of Theorem 4.17. □

5. High undecidability of topological and arithmetical properties

We prove that it is highly undecidable to determine the topological complexity of a Petri net ω-language. As usual, since there is a finite description of a real time 1-counter Büchi automaton or of a 4-blind-counter Büchi automaton, we can define a Gödel numbering of all 1-counter Büchi automata or of all 4-blind-counter Büchi automata and then speak about the 1-counter Büchi automaton

(or 4-blind-counter Büchi automaton) of index z. Recall first the following result, proved in [55], where we denote $\mathcal{A}_z$ the real time 1-counter Büchi automaton of index z reading words over a fixed finite alphabet Σ having at least two letters. We refer the reader to a textbook like [56] for more background about the analytical hierarchy of subsets of the set $\mathbb{N}$ of natural numbers.

Theorem 5.1. Let α be a countable ordinal. Then

1. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is in the Borel class $\mathbf{\Sigma}^0_\alpha\}$ is Π^1_2-hard.
2. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is in the Borel class $\mathbf{\Pi}^0_\alpha\}$ is Π^1_2-hard.
3. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is a Borel set $\}$ is Π^1_2-hard.

Using the previous constructions we can now easily show the following result, where $\mathcal{P}_z$ is the real time 4-blind-counter Büchi automaton of index z.

Theorem 5.2. Let $\alpha \geq 2$ be a countable ordinal. Then

1. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is in the Borel class $\mathbf{\Sigma}^0_\alpha\}$ is Π^1_2-hard.
2. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is in the Borel class $\mathbf{\Pi}^0_\alpha\}$ is Π^1_2-hard.
3. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is a Borel set $\}$ is Π^1_2-hard.

Proof:
It follows from the fact that one can easily get an injective recursive function $g : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{P}_{\mathcal{A}_z} = h(\mathrm{L}(\mathcal{A}_z)) \cup \mathcal{L} = \mathrm{L}(\mathcal{P}_{g(z)})$ and from the following equivalences which hold for each countable ordinal $\alpha \geq 2$:

1. $\mathrm{L}(\mathcal{A}_z)$ is in the Borel class $\mathbf{\Sigma}^0_\alpha$ (resp., $\mathbf{\Pi}^0_\alpha$) $\Longleftrightarrow$ $\mathrm{L}(\mathcal{P}_{g(z)})$ is in the Borel class $\mathbf{\Sigma}^0_\alpha$ (resp., $\mathbf{\Pi}^0_\alpha$).
2. $\mathrm{L}(\mathcal{A}_z)$ is a Borel set $\Longleftrightarrow$ $\mathrm{L}(\mathcal{P}_{g(z)})$ is a Borel set. □

Recall that the arithmetical properties of ω-languages of real time 1-counter Büchi automata were also proved to be highly undecidable in [55].

Theorem 5.3. Let $n \geq 1$ be an integer. Then

1. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is in the arithmetical class $\Sigma_n\}$ is Π^1_2-complete.
2. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is in the arithmetical class $\Pi_n\}$ is Π^1_2-complete.
3. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{A}_z)$ is a Δ^1_1-set $\}$ is Π^1_2-complete.

We can now prove similar results for ω-languages of real time 4-blind-counter Büchi automata.

Theorem 5.4. Let $n \geq 2$ be an integer. Then

1. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is in the arithmetical class $\Sigma_n\}$ is Π^1_2-complete.
2. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is in the arithmetical class $\Pi_n\}$ is Π^1_2-complete.
3. $\{z \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z)$ is a Δ^1_1-set $\}$ is Π^1_2-complete.

Proof:
Firstly, the three sets of integers considered in this theorem can be seen to be in the class Π^1_2. This can be proved in a very similar way as in the proof of [55, Theorem 3.26]. Secondly, the completeness part of the results follows from Theorem 5.3 and from the fact that one can easily get an injective recursive function $g\colon \mathbb{N} \to \mathbb{N}$ such that $\mathcal{P}_{\mathcal{A}_z} = h(\mathrm{L}(\mathcal{A}_z)) \cup \mathcal{L} = \mathrm{L}(\mathcal{P}_{g(z)})$ and from the following equivalences which hold for each integer $n \geq 2$, due to the fact that $\mathcal{L}$ is an arithmetical Σ^0_1-set and h is a recursive function from Σ^ω onto the effective closed set $h(\Sigma^\omega)$.

1. $\mathrm{L}(\mathcal{A}_z)$ is in the arithmetical class Σ_n (resp., Π_n) $\Longleftrightarrow$ $\mathrm{L}(\mathcal{P}_{g(z)})$ is in the arithmetical class Σ_n (resp., Π_n)
2. $\mathrm{L}(\mathcal{A}_z)$ is a Δ^1_1-set $\Longleftrightarrow$ $\mathrm{L}(\mathcal{P}_{g(z)})$ is a Δ^1_1-set.

□

6. High undecidability of the equivalence and the inclusion problems

We now add a result obtained from our previous constructions and which is important for verification purposes.

Theorem 6.1. The equivalence and the inclusion problems for ω-languages of Petri nets, or even for ω-languages in the class $\mathbf{r\text{-}BCL}(4)_\omega$, are Π^1_2-complete.

1. $\{(z, z') \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z) = \mathrm{L}(\mathcal{P}_{z'})\}$ is Π^1_2-complete
2. $\{(z, z') \in \mathbb{N} \mid \mathrm{L}(\mathcal{P}_z) \subseteq \mathrm{L}(\mathcal{P}_{z'})\}$ is Π^1_2-complete

Proof:
Firstly, it is easy to see that each of these decision problems is in the class Π^1_2, since the equivalence and the inclusion problems for ω-languages of Turing machines are already in the class Π^1_2, see [57, 55]. The completeness part follows from the fact that the equivalence and the inclusion problems for ω-languages accepted by real time 1-counter Büchi automata are Π^1_2-complete [55], and from the fact that there exists an injective recursive function $g : \mathbb{N} \to \mathbb{N}$ such that $\mathcal{P}_{\mathcal{A}_z} = \mathcal{P}_{g(z)}$, and then from the following equivalences:

1. $\mathrm{L}(\mathcal{A}_z) = \mathrm{L}(\mathcal{A}_{z'}) \Longleftrightarrow \mathrm{L}(\mathcal{P}_{g(z)}) = \mathrm{L}(\mathcal{P}_{g(z')})$
2. $\mathrm{L}(\mathcal{A}_z) \subseteq \mathrm{L}(\mathcal{A}_{z'}) \Longleftrightarrow \mathrm{L}(\mathcal{P}_{g(z)}) \subseteq \mathrm{L}(\mathcal{P}_{g(z')})$

□

7. Determinacy of Wadge games

We proved in [50] that the determinacy of Wadge games between two players in charge of ω-languages accepted by real time 1-counter Büchi automata, denoted **W-Det**($\mathbf{r\text{-}CL}(1)_\omega$), is equivalent to the (effective) analytic Wadge determinacy.

We can now state the following result, proved within the axiomatic system **ZFC**.

Theorem 7.1. The determinacy of Wadge games between two players in charge of ω-languages in the class **r**-$\mathbf{BCL}(4)_\omega$ is equivalent to the effective analytic (Wadge) determinacy, and thus is not provable in the axiomatic system **ZFC**.

Proof:
It was proved in [50] that the following equivalence holds: **W-Det**(**r**-$\mathbf{CL}(1)_\omega$) $\Longleftrightarrow$ **W-Det**(Σ^1_1). The implication **W-Det**(Σ^1_1)$\Longrightarrow$ **W-Det**(**r**-$\mathbf{BCL}(4)_\omega$) is obvious since the class $\mathbf{BCL}(4)_\omega$ is included into the class Σ^1_1. To prove the reverse implication, we assume that **W-Det**(**r**-$\mathbf{BCL}(4)_\omega$) holds and we show that every Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$ between two players in charge of ω-languages of the class **r**-$\mathbf{CL}(1)_\omega$ is determined (we assume without loss of generality that the two real time 1-counter Büchi automata $\mathcal{A}$ and $\mathcal{B}$ read words over the same alphabet Σ).

It is sufficient to consider the cases where at least one of two ω-languages $\mathrm{L}(\mathcal{A})$ and $\mathrm{L}(\mathcal{B})$ is non-Borel, since the Borel Wadge determinacy is provable in **ZFC**. On the other hand, we have seen how we can construct some real time 4-blind-counter Büchi automata $\mathcal{P}_\mathcal{A}$ and $\mathcal{P}_\mathcal{B}$ such that $\mathrm{L}(\mathcal{P}_\mathcal{A}) = h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}$ and $\mathrm{L}(\mathcal{P}_\mathcal{B}) = h(\mathrm{L}(\mathcal{B})) \cup \mathcal{L}$.

We can firstly consider the case where $\mathrm{L}(\mathcal{A})$ is Borel of Wadge degree smaller than ω, and $\mathrm{L}(\mathcal{B})$ is non-Borel. In that case $\mathrm{L}(\mathcal{A})$ is in particular a $\mathbf{\Pi}^0_2$-set. Recall now that we can infer from Hurewicz's Theorem, see [48, page 160], that an analytic subset of Σ^ω is either $\mathbf{\Pi}^0_2$-hard or a $\mathbf{\Sigma}^0_2$-set. Thus $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Pi}^0_2$-hard and Player 2 has a winning strategy in the game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$.

Secondly we consider the case where $\mathrm{L}(\mathcal{A})$ and $\mathrm{L}(\mathcal{B})$ are either non-Borel or Borel of Wadge degree greater than ω. By hypothesis we know that the Wadge game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ is determined, and that one of the players has a winning strategy. Using the above constructions and reasonings we used in the proofs of Lemmas 4.5 and 4.10, we can easily show that the same player has a winning strategy in the Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$.

We now consider the two following cases:

First case. Player 2 has a w.s. in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$. If $\mathrm{L}(\mathcal{B})$ is Borel then $\mathrm{L}(\mathcal{P}_\mathcal{B})$ is easily seen to be Borel and then $\mathrm{L}(\mathcal{P}_\mathcal{A})$ is also Borel because $\mathrm{L}(\mathcal{P}_\mathcal{A}) \leq_W \mathrm{L}(\mathcal{P}_\mathcal{B})$. Thus $\mathrm{L}(\mathcal{A})$ is also Borel and the game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$ is determined. Assume now that $\mathrm{L}(\mathcal{B})$ is not Borel. Consider the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$. We claim that Player 2 has a w.s. in that game which is easily deduced from a w.s. of Player 2 in the Wadge game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B})) = W(h(\mathrm{L}(\mathcal{A})) \cup \mathcal{L}, h(\mathrm{L}(\mathcal{B})) \cup \mathcal{L})$. Consider a play in this latter game where Player 1 remains in the closed set $h(\Sigma^\omega)$: she writes a beginning of a word in the form

$$A0x(1)B0^2x(2)A \cdots B0^{2n}x(2n)A \cdots$$

Then player 2 writes a beginning of a word in the form

$$A0x'(1)B0^2x'(2)A \cdots B0^{2p}x'(2p)A \cdots$$

where $p \leq n$. Then the strategy of Player 2 in $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$ consists to write $x'(1)x'(2) \cdots x'(2p)$ when Player 1 writes $x(1)x(2) \cdots x(2n)$. (Notice that Player 2 is allowed to skip, provided he really writes an ω-word in ω steps). If the strategy for Player 2 in $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ was at some step to go out of the closed set $h(\Sigma^\omega)$ then this means that the word he has now written is not a prefix

of any ω-word of $h(\Sigma^\omega)$, and Player 2 "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$ and will stay in this set. Two subcases may now appear.

Subcase A. When Player 2 in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, he actually enters in the open set $\mathcal{L}$. Then the final word written by Player 2 will surely be inside $\mathrm{L}(\mathcal{P}_\mathcal{B})$. Player 2 in the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$ can now write a letter of $\Delta - \Sigma$ in such a way that he is now like a player in charge of the whole set and he can now write an ω-word u so that his final ω-word will be inside $\emptyset + (\emptyset + \mathrm{L}(\mathcal{B}))$. Thus Player 2 wins this play too.

Subcase B. When Player 2 in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, he does not enter in the open set $\mathcal{L}$. Then Player 2, in the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$, being first like a player in charge of the set $(\emptyset + \mathrm{L}(\mathcal{B}))$, can write a letter of $\Delta - \Sigma$ in such a way that he is now like a player in charge of the empty set and he can now continue, writing an ω-word u. If Player 2 in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ never enters in the open set $\mathcal{L}$ then the final word written by Player 2 will be in $\mathcal{L}'$ and thus surely outside $\mathrm{L}(\mathcal{P}_\mathcal{B})$, and the final word written by Player 2 will be outside the empty set. So in that case Player 2 wins this play too in the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$. If at some step of the play, in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$, Player 2 enters in the open set $\mathcal{L}$ then his final ω-word will be surely in $\mathrm{L}(\mathcal{P}_\mathcal{B})$. In that case Player 2, in charge of the set $\emptyset + (\emptyset + \mathrm{L}(\mathcal{B}))$ in the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$, can again write an extra letter and choose to be in charge of the whole set and he can now write an ω-word v so that his final ω-word will be inside $\emptyset + (\emptyset + \mathrm{L}(\mathcal{B}))$. Thus Player 2 wins this play too.

So we have proved that Player 2 has a w.s. in the Wadge game $W(\mathrm{L}(\mathcal{A}), \emptyset + (\emptyset + \mathrm{L}(\mathcal{B})))$ or equivalently that $\mathrm{L}(\mathcal{A}) \leq_W \emptyset + (\emptyset + \mathrm{L}(\mathcal{B}))$. But by Lemma 4.12 we know that $\mathrm{L}(\mathcal{B}) \equiv_W \emptyset + (\emptyset + \mathrm{L}(\mathcal{B}))$ and thus $\mathrm{L}(\mathcal{A}) \leq_W \mathrm{L}(\mathcal{B})$ which means that Player 2 has a w.s. in the Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$.

Second case. Player 1 has a w.s. in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$. Notice that this implies that $\mathrm{L}(\mathcal{P}_\mathcal{B}) \leq_W \mathrm{L}(\mathcal{P}_\mathcal{A})^-$. Thus if $\mathrm{L}(\mathcal{A})$ is Borel then $\mathrm{L}(\mathcal{P}_\mathcal{A})$ is Borel, $\mathrm{L}(\mathcal{P}_\mathcal{A})^-$ is also Borel, and $\mathrm{L}(\mathcal{P}_\mathcal{B})$ is Borel as the inverse image of a Borel set by a continuous function, and $\mathrm{L}(\mathcal{B})$ is also Borel, so the Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$ is determined. We now assume that $\mathrm{L}(\mathcal{A})$ is not Borel and we consider the Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$. Player 1 has a w.s. in this game which is easily constructed from a w.s. of the same player in the game $W(\mathrm{L}(\mathcal{P}_\mathcal{A}), \mathrm{L}(\mathcal{P}_\mathcal{B}))$ as follows. For this consider a play in this latter game where Player 2 does not go out of the closed set $h(\Sigma^\omega)$.

He writes a beginning of a word in the form

$$A0x(1)B0^2x(2)A \cdots B0^n x(n)A \cdots$$

Then Player 1 writes a beginning of a word in the form

$$A0x'(1)B0^2x'(2)A \cdots B0^p x'(p)A \cdots$$

where $n \leq p$ (notice that here without loss of generality the notation implies that n and p are even, since the segments $B0^n x(n)A$ and $B0^p x'(p)A$ begin with a letter B but this is not essential in the proof). Then the strategy for Player 1 in $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$ consists to write $x'(1)x'(2) \cdots x'(p)$ when Player 2 writes $x(1)x(2) \cdots x(n)$. After ω steps, the ω-word written by Player 1 is in $\mathrm{L}(\mathcal{A})$ iff the ω-word written by Player 2 is not in the set $\mathrm{L}(\mathcal{B})$, and thus Player 1 wins the play.

If the strategy for Player 1 in $W(\mathrm{L}(\mathcal{P}_{\mathcal{A}}), \mathrm{L}(\mathcal{P}_{\mathcal{B}}))$ was at some step to go out of the closed set $h(\Sigma^\omega)$ then this means that she "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$ and will stay in this set. Two subcases may now appear.

Subcase A. When Player 1 in the game $W(\mathrm{L}(\mathcal{P}_{\mathcal{A}}), \mathrm{L}(\mathcal{P}_{\mathcal{B}}))$ "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, she actually enters in the open set $\mathcal{L}$. Then the final word written by Player 1 will surely be inside $\mathrm{L}(\mathcal{P}_{\mathcal{A}})$. But she wins the play since she follows a winning strategy and this leads to a contradiction. Indeed if Player 2 decided to also enter in in the open set $\mathcal{L}$ then Player 2 would win the play. Thus this case is actually not possible.

Subcase B. When Player 1 in the game $W(\mathrm{L}(\mathcal{P}_{\mathcal{A}}), \mathrm{L}(\mathcal{P}_{\mathcal{B}}))$ "enters" in the open set $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$, she does not enter in the open set $\mathcal{L}$. But Player 2 would be able to do the same and enter in $h(\Sigma^\omega)^- = \mathcal{L} \cup \mathcal{L}'$ but not (for the moment) in the open set $\mathcal{L}$. And if at some step of the play, Player 1 would enter in the open set $\mathcal{L}$ then Player 2 could do the same, and thus Player 2 would win the play. Again this is not possible since Player 1 wins the play since she follows a winning strategy.

Finally both subcases A and B cannot occur and this shows that Player 1 has a w.s. in the Wadge game $W(\mathrm{L}(\mathcal{A}), \mathrm{L}(\mathcal{B}))$. □

8. Non-Borel ω-languages of one counter Petri nets

In this section we prove the following result.

Theorem 8.1. There exists a 1-blind counter automaton $\mathcal{A}_1$ such that the ω-language $\mathrm{L}(\mathcal{A}_1)$ of that automaton is $\mathbf{\Sigma}^1_1$-complete, hence non-Borel.

The crucial obstacle of that construction is the fact that the simulation order $\preceq$ for 1-blind counter automata has a finite width, namely the number of states of the machine. This means that the non-deterministic choices of such a machine are inherently ordered. This explains why we use a $\mathbf{\Sigma}^1_1$-hard ω-language that itself is a set of orders.

8.1. Topology of orders

Consider a set X and a relation $o \subseteq X \times X$ on X. We say that o is a *linear order* if it is reflexive, transitive, and anti-symmetric. We interpret a pair $(x, x') \in o$ as representing the fact that x is o-smaller-or-equal than x'. A linear order o is *ill-founded* if there exists an infinite sequence $x_0, x_1, \ldots$ of pairwise distinct elements of X such that for all n we have $(x_{n+1}, x_n) \in o$, i.e. the sequence is strictly decreasing. Such a sequence indicates an infinite o-descending chain. An order that is not ill-founded is called *well-founded*.

The binary tree is the set of all sequences of *directions* $\mathcal{T} =_{\mathrm{df}} \{\mathtt{L}, \mathtt{R}\}^*$ where the *directions* $\mathtt{L}$ and $\mathtt{R}$ are two fixed distinct symbols. For technical reasons we sometimes consider a third direction $\mathtt{M}$ (it does not occur in the binary tree).

A set $X \subseteq \mathcal{T}$ can be naturally identified with its characteristic function $X \in \{0,1\}^{\left(\{\mathtt{L},\mathtt{R}\}^*\right)}$. Thus, the family of all subsets of the binary tree, with the natural product topology, is homeomorphic with the Cantor set $\{0,1\}^\omega$.

The elements $v, x \in \mathcal{T}$ are called *nodes*. Nodes are naturally ordered by the following three orders:

- the prefix order $\sqsubseteq$, as defined on page 247,
- the lexicographic order: $v \leq_{\text{lex}} x$ if v is lexicographically smaller than x (we assume that $\mathtt{L} <_{\text{lex}} \mathtt{M} <_{\text{lex}} \mathtt{R}$),
- the infix order: $v \leq_{\text{inf}} x$ if $v\mathtt{M}^\omega$ (i.e. the ω-word obtained by appending infinitely many symbols $\mathtt{M}$ after v) is lexicographically less or equal than $x\mathtt{M}^\omega$. This order corresponds to the left-to-right order on the nodes of the tree, when it is drawn in a standard way.

Notice that, for every fixed n, when restricted to $\{\mathtt{L}, \mathtt{R}\}^n$, the lexicographic and infix orders coincide. However, $\mathtt{L} <_{\text{inf}} \epsilon <_{\text{inf}} \mathtt{R}$ but ϵ is the minimal element of $\leq_{\text{lex}}$. Both the lexicographic and infix orders are linear.

Since the infix order is countable, dense, and has no minimal nor maximal elements, we obtain the following fact.

Fact 8.2. $(\mathcal{T}, \leq_{\text{inf}})$ is isomorphic to the order of rational numbers $(\mathbb{Q}, \leq)$.

In the following part of the paper we will use the following set.

$$\text{IF}_{\text{inf}} =_{\text{df}} \{X \subseteq \mathcal{T} \mid X \text{ contains an infinite } \leq_{\text{inf}}\text{-descending chain}\}. \tag{1}$$

Lemma 8.3. The set IF_{inf} is $\mathbf{\Sigma}^1_1$-complete.

Proof:
IF_{inf} belongs to $\mathbf{\Sigma}^1_1$ just by the form of the definition.

We prove $\mathbf{\Sigma}^1_1$-hardness by a reduction from the set of ill-founded linear orders on ω (seen as elements of $\{0,1\}^{\omega\times\omega}$). Let us prove this fact more formally. Consider an element $o \in \{0,1\}^{\omega\times\omega}$ that is a linear order on ω. The latter set is $\mathbf{\Sigma}^1_1$-complete by a theorem by Lusin and Sierpiński [48, Theorem 27.12]. We will inductively define $X_o \subseteq \mathcal{T}$ in such a way to ensure that $o \mapsto X_o$ is a continuous mapping and o is ill-founded if and only if $X_o \in \text{IF}_{\text{inf}}$.

Let us proceed inductively, defining a sequence of nodes $(x_n)_{n\in\omega} \subseteq \mathcal{T}$. Our invariant says that $|x_k| = k$ and the map $k \mapsto x_k$ is an isomorphism of the orders $\big(\{0,\dots,n\}, o\big)$ and $\big(\{x_0,\dots,x_n\}, \leq_{\text{inf}}\big)$. We start with $x_0 = \epsilon$ (i.e. the root of $\mathcal{T}$). Assume that $x_0,\dots,x_n$ are defined and satisfy the invariants. By the definition of $\leq_{\text{inf}}$, there exists a node $x \in \{\mathtt{L}, \mathtt{R}\}^{n+1}$ such that for $k = 0, 1, \dots, n$ we have $x \leq_{\text{inf}} x_k$ if and only if $(n{+}1, k) \in o$. Let x_{n+1} be such a node.

The above induction defines an infinite sequence of nodes $x_0, x_1, \dots$ Let $X_o =_{\text{df}} \{x_n \mid n \in \omega\} \subseteq \mathcal{T}$. By the definition of X_o, the mapping $o \mapsto X_o$ is continuous — the fact whether a node $x \in \mathcal{T}$ belongs to X_o depends only on $o \cap \{0, 1, \dots, |x|\}^2$. Using our invariant, we know that the map $k \mapsto x_k$ is an isomorphism of the orders $\big(\omega, o\big)$ and $\big(X_o, \leq_{\text{inf}}\big)$. Thus, o is ill-founded if and only if $X_o \in \text{IF}_{\text{inf}}$. □

To construct our continuous reduction in the one-counter case, we need the following simple lemma that provides an alternative characterisation of the set IF_{inf}. Let us introduce the following definition.

Definition 8.4. A sequence $v_0, v_1, \ldots \in \mathcal{T}$ is called a *correct chain* if $v_0 = \epsilon$ and for every $n = 0, 1, \ldots$:

1. $|v_{n+1}| = |v_n| + 1$,
2. $v_{n+1} \leq_{\text{inf}} v_n\mathtt{R}$ (or equivalently $v_{n+1} \leq_{\text{lex}} v_n\mathtt{R}$).

A correct chain is *witnessing* for a set $X \subseteq \mathcal{T}$ if for infinitely many n we have $v_n \in X$ and $v_{n+1} \leq_{\text{inf}} v_n\mathtt{L}$.

Intuitively, the definition forces the sequence to be *not so-much increasing* in the infix order $\leq_{\text{inf}}$: the successive element v_{n+1} needs to be *to the left* in the tree from $v_n\mathtt{R}$. Such a sequence is *witnessing* for a set X if infinitely many times it belongs to X and at these moments it actually drops in $\leq_{\text{inf}}$.

Lemma 8.5. A set $X \subseteq \mathcal{T}$ belongs to IF_{inf} if and only if there exists a correct chain witnessing for X.

Proof:
First take a correct chain witnessing for X. Let $x_0, x_1, \ldots$ be the subsequence that shows that $(v_n)_{n\in\omega}$ is witnessing for X. In that case, by the definition, for all n we have $x_n \in X$ and $x_{n+1} <_{\text{inf}} x_n$ (because $x_{n+1}\mathtt{M}^\omega \leq_{\text{lex}} x_n\mathtt{LR}^\omega <_{\text{lex}} x_n\mathtt{M}^\omega$). Thus, X has an infinite $\leq_{\text{inf}}$-descending chain and belongs to IF_{inf}.

Now assume that $X \in \text{IF}_{\text{inf}}$ and $x_0 >_{\text{inf}} x_1 >_{\text{inf}} x_2 >_{\text{inf}} \ldots$ is a sequence witnessing that. Without loss of generality we can assume that $|x_{n+1}| > |x_n|$ because for each fixed depth k there are only finitely many nodes of $\mathcal{T}$ in $\{\mathtt{L}, \mathtt{R}\}^{\leq k}$. We can now add intermediate nodes in-between the sequence $(x_n)_{n\in\omega}$ to construct a correct chain witnessing for X; the following pseudo-code realises this goal:

```
n := 0;
i := 0;
while (true) {
  if (n > |x_i|) {
    i := i + 1;
  }

  v_n := x_i[n];
  n := n + 1;
}
```

Clearly, Property 1 in the definition of a correct chain is guaranteed. Let $i \in \omega$ and $n = |x_i|$. By the fact that $x_{i+1} <_{\text{inf}} x_i$ we know that $x_{i+1}[n+1] \leq_{\text{inf}} x_i\mathtt{L}$. Therefore, for every $n \in \omega$ we have $v_{n+1} \leq_{\text{inf}} v_n\mathtt{R}$ and if $n = |x_i|$ for some i then $v_{n+1} \leq_{\text{inf}} v_n\mathtt{L}$. It implies that the sequence $(v_n)_{n\in\omega}$ satisfies Property 2 in the definition of a correct chain. It is clearly witnessing for X because it contains $(x_n)_{n\in\omega}$ as a subsequence. □

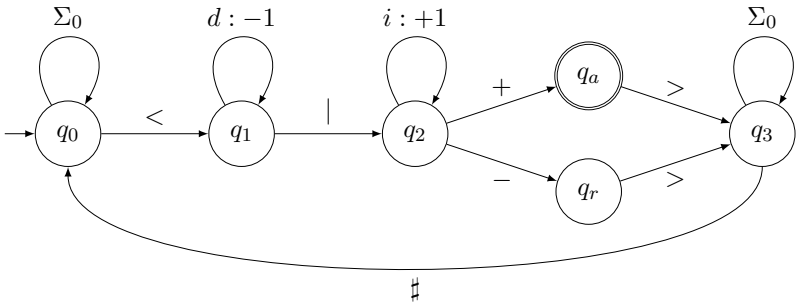

Figure 1. The automaton $\mathcal{A}_1$ with 1-blind counter that recognises a $\mathbf{\Sigma}^1_1$-complete ω-language.

8.2. Hardness for one counter

We are now ready to provide a definition of a 1-blind counter Büchi automaton $\mathcal{A}_1$ recognising $\mathbf{\Sigma}^1_1$-complete ω-language. The automaton $\mathcal{A}_1$ is depicted in Figure 1, with the convention that an edge $q \xrightarrow{a:j} q'$ represents a transition from the state q to q', over the letter a; that modifies the unique counter value by j, i.e. $a\colon (q, c_1) \mapsto_{\mathcal{A}_1} (q', c_1 + j)$. Moreover, the state q_0 is initial and q_a is the only accepting state.

Let $\Sigma_0 =_{\mathrm{df}} \{<, d, |, i, +, -, >\}$ and consider the alphabet $\Sigma =_{\mathrm{df}} \Sigma_0 \cup \{\sharp\}$.

An ω-word accepted by $\mathcal{A}_1$ consists of infinitely many *phases* separated by $\sharp$. Each phase is a finite word over the alphabet Σ_0. In our reduction we will restrict to phases being sequences of *blocks*, each block being a finite word of the form given by the following definition (for $n, m \in \omega$ and $s \in \{+, -\}$):

$$B^s(-n, +m) =_{\mathrm{df}} < d^n \mid i^m \; s > \; \in A_0^*. \tag{2}$$

Such a block is *accepting* if $s = +$, otherwise $s = -$ and the block is *rejecting*. If $\mathcal{A}_1$ starts reading a block and moves from q_0 to q_1 over $<$ then we say that it *chooses* this block. Otherwise $\mathcal{A}_1$ stays in q_0 and it does not *choose* the given block. By the construction of the automaton $\mathcal{A}_1$, in every run it needs to choose exactly one block from each phase. Additionally, the run is accepting if and only if infinitely many of the chosen blocks are accepting.

Proposition 8.6. There exists a continuous reduction from $\mathrm{IF}_{\mathrm{inf}}$ to the ω-language recognised by $\mathcal{A}_1$.

We will take a set $X \subseteq \mathcal{T}$ and construct an ω-word $\alpha(X)$. This ω-word will be a concatenation of infinitely many phases $u_0 \sharp u_1 \sharp \cdots$. The n-th phase u_n will depend on $X \cap \{\mathtt{L}, \mathtt{R}\}^n$. The configurations (q_0, c) reached at the beginning of an n-th phase will be in correspondence with the nodes $v \in \{\mathtt{L}, \mathtt{R}\}^n$. The bigger the value c, the higher in the infix order (or the lexicographic order, as they overlap here) the respective node v is.

To precisely define our ω-word $\alpha(X)$ we need to define some auxiliary functions. First, we define inductively the function $b\colon \mathcal{T} \to \omega$, assigning to nodes $v \in \mathcal{T}$ their binary value $b(v)$:

- $b(\epsilon) = 0$,
- $b(v\mathtt{L}) = 2 \cdot b(v)$,

- $b(v\mathtt{R}) = 2 \cdot b(v) + 1$.

Note that for every $n \in \omega$ we have

$$b\big(\{\mathtt{L},\mathtt{R}\}^n\big) = \{0, 1, \ldots, 2^n{-}1\},$$

and the function is bijective between these sets.

Now we can define fast-growing functions: $m\colon \{-1\} \cup \omega \to \omega$ and $e\colon \mathcal{T} \to \omega$:

$$\begin{aligned} m(-1) &= 1, \\ m(n) &= m(n-1) \cdot 2^n && \text{for } n \in \omega, \\ e(v) &= m(|v|-1) \cdot b(v) && \text{for } v \in \mathcal{T}. \end{aligned}$$

These functions allow to use a big range of the possible values of a single counter of the automaton to represent particular nodes of the tree. We will use the following two invariants of this definition, for $n \in \omega$ and $v, v' \in \{\mathtt{L},\mathtt{R}\}^n$:

$$v <_{\text{inf}} v' \iff e(v) \leq e(v'), \tag{3}$$

$$e(v) + m(|v|-1) \;\leq\; m(|v|). \tag{4}$$

We take any $n = 0, 1, \ldots$ and define the n-th phase u_n. Let u_n be the concatenation of the following blocks, for all $v \in \{\mathtt{L},\mathtt{R}\}^n$ and $d \in \{\mathtt{L},\mathtt{R}\}$:

$$B^s\big(-e(v), +e(vd)\big),$$

where $s = +$ if $v \in X$ and $d = \mathtt{L}$; otherwise $s = -$. Thus, the n-th phase is a concatenation of 2^{n+1} blocks, one for each node vd in $\{\mathtt{L},\mathtt{R}\}^{n+1}$.

To conclude the proof of Proposition 8.6 it is enough to prove the following two lemmas.

Lemma 8.7. If there exists a correct chain witnessing for X then $\alpha(X) \in \mathrm{L}(\mathcal{A}_1)$.

Proof:
Consider a correct chain $(v_n)_{n\in\omega}$ witnessing for X. Assume that $I \subseteq \omega$ is an infinite set such that for $n \in I$ we have $v_n \in X$ and $v_{n+1} \leq_{\text{inf}} v_n\mathtt{L}$. Let us construct inductively a run r of $\mathcal{A}_1$ on $\alpha(X)$. The invariant is that for each $n \in \omega$ the configuration of r before reading the n-th phase of $\alpha(X)$ is of the form (q_0, c_n) with $c_n \geq e(v_n)$. To define r it is enough to decide which block to choose from an n-th phase of $\alpha(X)$:

- if $n \in I$ then choose the block $B^+\big(-e(v_n), +e(v_n\mathtt{L})\big)$,
- otherwise choose the block $B^-\big(-e(v_n), +e(v_n\mathtt{R})\big)$.

Notice that by the invariant, it is allowed to choose the respective blocks as $c_n \geq e(v_n)$. Because of (3) and the fact that $(v_n)_{n\in\omega}$ is a correct chain, the invariant is preserved. As the set I is infinite, the constructed run chooses an accepting block infinitely many times and thus is accepting. □

Lemma 8.8. If $\alpha(X) \in \mathrm{L}(\mathcal{A}_1)$ then there exists a correct chain witnessing for X.

Proof:
Assume that r is an accepting run of $\mathcal{A}_1$ over $\alpha(X)$. For $n = 0, 1, \ldots$ let (q_0, c_n) be the configuration in r before reading the n-th phase of $\alpha(X)$ and assume that r chooses a block of the form $B^{s_n}\big(-e(v_n), +e(v_n d_n)\big)$ in the n-th phase of $\alpha(X)$. Our aim is to show that $(v_n)_{n\in\omega}$ is a correct chain witnessing for X. First notice that by the construction of $\alpha(X)$ we have $|v_n| = n$.

Clearly, as the counter needs to be non-negative, we have $e(v_n) \leq c_n$. Notice that by (4) we obtain inductively for $n = 0, 1, \ldots$ that $c_n < m(n)$. Therefore, we have

$$\begin{aligned} m(n) \cdot b(v_{n+1}) = e(v_{n+1}) \leq c_{n+1} = \\ = c_n - e(v_n) + e(v_n d_n) < m(n) + e(v_n d_n) = \\ = m(n) + m(n) \cdot b(v_n d_n). \end{aligned}$$

By dividing by $m(n)$ we obtain $b(v_{n+1}) < 1 + b(v_n d_n)$, thus $b(v_{n+1}) \leq b(v_n d_n)$ and therefore $v_{n+1} \leq_{\inf} v_n d_n \leq_{\inf} v_n \mathtt{R}$. Moreover, if $s_n = +$ (i.e. the n-th chosen block is accepting) then $v_n \in X$ and $d_n = \mathtt{L}$. Therefore, as r chooses infinitely many accepting blocks, $(v_n)_{n\in\omega}$ is witnessing for X.

□

This concludes the proof of Proposition 8.6. Thus the ω-language of $\mathcal{A}_1$ is indeed $\mathbf{\Sigma}^1_1$-hard and therefore $\mathbf{\Sigma}^1_1$-complete.

8.3. Undecidable properties of 1-blind counter ω-languages

From the above result we can now easily infer the following undecidability result.

Theorem 8.9. It is undecidable to determine whether the ω-language of a given 1-blind counter automaton is Borel (respectively, in the Borel class $\mathbf{\Sigma}^0_\alpha$, in the Borel class $\mathbf{\Pi}^0_\alpha$, for a given ordinal α).

Proof:
Let $L_1 = \mathrm{L}(\mathcal{A}_1) \subseteq \Sigma^\omega$ be the $\mathbf{\Sigma}^1_1$-complete ω-language accepted by the 1-blind counter automaton $\mathcal{A}_1$ given above.

We consider the shuffle operation for two ω-words x and y in Σ^ω given by

$$\mathrm{Sh}(x, y) = x(1)y(1)x(2)y(2).. \in \Sigma^\omega$$

This is extended to the shuffle of ω-languages by $\mathrm{Sh}(L, L') = \{\mathrm{Sh}(x, y) \mid x \in L \text{ and } y \in L'\}$.

Let now L be an ω-language of a given 1-blind counter automaton $\mathcal{A}$ over the alphabet Σ. We set $S = \mathrm{Sh}(L_1, \Sigma^\omega) \cup \mathrm{Sh}(\Sigma^\omega, L)$ It is easy to see that one can construct, from $\mathcal{A}$ and $\mathcal{A}_1$, another 1-blind counter automaton $\mathcal{B}$ accepting S.

There are now two cases:

First Case. $L = \Sigma^\omega$.

In that case $S = \Sigma^\omega$ is in every Borel class (and actually in every Wadge class, except the class of the empty set).

Second Case. L is not equal to Σ^ω.

In that case there exists an ω-word $x \in \Sigma^\omega$ which is not in L. Let now T be the intersection of S and of $\mathrm{Sh}(\Sigma^\omega, \{x\})$. Then $T = \mathrm{Sh}(L_1, \{x\})$ is also $\mathbf{\Sigma}^1_1$-complete, and since L_1 is continuously reducible to S by $F : y \to \mathrm{Sh}(y, x)$ it follows that S is $\mathbf{\Sigma}^1_1$-complete.

Now the conclusion follows from a recent result that the universality problem for one blind counter Büchi automata is undecidable, see [58]. □

Notice that one can also get other undecidability results, using the above one about topological properties. First we can state the following theorem showing that the arithmetical properties of 1-blind counter ω-languages are also undecidable.

Theorem 8.10. It is undecidable to determine whether the ω-language of a given 1-blind counter automaton is an effective Δ^1_1-set (respectively, an arithmetical Σ^0_n-set, an arithmetical Π^0_n-set, for a given integer $n \geq 1$).

Proof:
We can use the above proof of Theorem 8.9. Indeed, in the first case the ω-language $S = \Sigma^\omega$ is in every arithmetical class. Moreover, in the second case the ω-language S is not a Borel set, and thus it is not an effective Δ^1_1-set and does not belong to any arithmetical class. □

Remark 8.11. It is open to determine the exact complexity of these undecidable problems. In particular we do not know whether they are highly undecidable, as in the general case of Petri nets or of 4-blind counter automata.

We can also use topological properties to prove other undecidability properties which are not directly linked to topology.

Theorem 8.12. It is undecidable to determine whether the ω-language of a given 1-blind counter automaton $\mathcal{A}$:

1. is a regular ω-language.
2. is accepted by a deterministic Petri net.
3. is accepted by a deterministic Turing machine.
4. has a complement $\mathrm{L}(\mathcal{A})^-$ which is accepted by a Petri net.
5. has a complement $\mathrm{L}(\mathcal{A})^-$ which is accepted by a Turing machine.

Proof:
We can again use the above proof of Theorem 8.9. Indeed, in the first case the ω-language $S = \Sigma^\omega$ satisfies the five items of the theorem. In the second case the ω-language S is non-Borel hence it is not a Boolean combination of $\mathbf{\Pi}^0_2$-sets and it does not satisfy any of the three first items. Moreover, in this case the complement of S is $\mathbf{\Pi}^1_1$-complete hence it cannot be accepted by any Turing machine and in particular by any Petri net (with Büchi acceptance condition). □

9. Inherent non-determinism

In this section we formally state and prove the following corollary.

Corollary 9.1. No model of deterministic, unambiguous, nor even countably-unambiguous automata with countably many configurations and a Borel acceptance condition can capture the class of ω-languages recognisable by real-time 1-blind counter Büchi automata.

It is expressed in the same spirit as the corresponding Theorem 5.5 in [59]: we consider an abstract model of automata $\mathcal{A}$ with a countable set of configurations C, an initial configuration $c_{\mathrm{I}} \in C$, a transition relation $\delta \subseteq C \times \Sigma \times C$, and an acceptance condition $W \subseteq C^\omega$. The notions of a run $\mathrm{run}(\alpha, \rho)$; an accepting run $\mathrm{acc}(\rho)$; and the language $\mathrm{L}(\mathcal{A})$ are defined in the standard way. Thus, under the assumption that the acceptance condition W is Borel, the set

$$P =_{\mathrm{df}} \big\{(\alpha, \rho) \in \Sigma^\omega \times C^\omega \mid \mathrm{run}(\alpha, \rho) \wedge \mathrm{acc}(\rho)\big\},$$

as in Definition 2.1 is also Borel. The assumptions that the machine is deterministic, unambiguous, or countably-unambiguous imply that the cardinality of the sections $P_\alpha =_{\mathrm{df}} \{\rho \mid (\alpha, \rho) \in P\}$ for $\alpha \in \Sigma^\omega$ is at most countable. Therefore, the following *small section theorem* by Lusin and Novikov applies.

Theorem 9.2. (see [48, Theorem 18.10])
Let X, Y be standard Borel spaces and let $P \subseteq X \times Y$ be Borel. If every section P_x is countable, then P has a Borel uniformisation and therefore $\pi_X(P)$ is Borel.

Therefore, we know that $\mathrm{L}(\mathcal{A}) = \pi_{\Sigma^\omega}(P)$ is Borel. Thus, no such machine can recognise $\mathrm{L}(\mathcal{A}_1)$ for the automaton $\mathcal{A}_1$ from Section 8, or any non-Borel ω-language of Petri nets obtained in Section 4 as these languages are non-Borel.

Notice that the above Theorem of Lusin and Novikov had already been used in the study of ambiguity of context-free ω-languages in [42] or of ω-languages of Turing machines in [36] and even for tree languages of tree automata [43]. In particular, it is proved in [36] that if $L \subseteq X^\omega$ is accepted by a Büchi Turing machine $\mathcal{T}$ and L is an analytic but non-Borel set, then the set of ω-words, which have $2^{\aleph_0}$ accepting runs by $\mathcal{T}$, has cardinality $2^{\aleph_0}$. This extends a similar result of [42] in the case of context-free ω-languages and infinitary rational relations. In that case we say that the ω-language L has the maximum degree of ambiguity (with regard to acceptance by Büchi Turing machines). From this result we can also infer the following one about ω-languages of Petri nets.

Theorem 9.3. Let $L \subseteq \Sigma^\omega$ be an ω-language accepted by a Büchi k-counter automaton $\mathcal{A}$ such that L is an analytic but non-Borel set. The set of ω-words, which have $2^{\aleph_0}$ accepting runs by $\mathcal{A}$, has cardinality $2^{\aleph_0}$.

Moreover, it is proved in [36, Theorem 4.12] that it is consistent with **ZFC** that there exists an ω-language accepted by a real-time 1-counter Büchi automaton which belongs to the Borel class $\mathbf{\Pi}^0_2$ and which has the maximum degree of ambiguity with regard to acceptance by Turing machines. It is

then easy to infer from this result and from the previous constructions of Section 4 that a similar result holds for an ω-language in the Borel class $\mathbf{\Pi}^0_2$ accepted by a 4-blind counter Büchi automaton.

We end this section with the following undecidability result.

Theorem 9.4. It is undecidable to determine whether the ω-language of a given 1-blind counter automaton $\mathcal{A}$:

1. is accepted by an unambiguous Petri net.
2. has the maximum degree of ambiguity with regard to acceptance by Petri nets.
3. has the maximum degree of ambiguity with regard to acceptance by Turing machines.

Proof:
We can again use the above proof of Theorem 8.9. Indeed, in the first case the ω-language $S = \Sigma^\omega$ is accepted by an unambiguous (and even deterministic) Petri net (without any counter). In the second case the ω-language S is non-Borel and thus it has the maximum degree of ambiguity with regard to acceptance by Petri nets or even by Turing machines. □

10. Determinisation of unambiguous Petri nets

The previous sections showed that non-deterministic blind counter automata are stronger in expressive power than any reasonable model of computation with a restricted form of non-determinism. This opens the question what is the actual expressive power of unambiguous blind counter automata. In this section we provide a construction allowing to simulate them using a variant of *deterministic* counter automata with *copying*.

A counter automaton $\mathcal{M} = \langle K, \Sigma, \Delta, q_0 \rangle$ allows *copying* if its transitions can additionally require to copy the value of one counter $\mathcal{C}_j$ to another counter $\mathcal{C}_{j'}$, symbolically $\mathcal{C}_{j'} := \mathcal{C}_j$. The copying instructions can be represented by another component of the transition relation taken from the set $2^{(\{1,\ldots,k\}^2)}$ indicating which counters should be copied to which (we can assume that all the copying is executed simultaneously). A run of such a machine is defined analogously as in Section 2 with the additional requirement that the copying instructions are executed in the natural way.

Theorem 10.1. If $\mathcal{A}$ is an unambiguous blind counter Büchi automaton then $\mathrm{L}(\mathcal{A})$ can be recognised by a deterministic Muller counter machine $\mathcal{M}$ that allows copying, hence by a deterministic Muller Turing machine. Moreover, the translation from $\mathcal{A}$ to $\mathcal{M}$ is effective.

The machine $\mathcal{M}$ above is not blind and during the construction we extensively use its ability to perform zero tests. For a discussion of the possible variants of the machine models involved in that theorem, see the end of this section.

It is known that every ω-language accepted by a deterministic Muller Turing machine is a Boolean combination of arithmetical Π^0_2-sets, hence an arithmetical Δ^0_3-set. In particular, an ω-language accepted by a deterministic Muller Turing machine is a Boolean combination of $\mathbf{\Pi}^0_2$-sets, hence a Borel $\mathbf{\Delta}^0_3$-set [2]. Thus we can also state the following corollary of Theorem 10.1.

Corollary 10.2. Assume that $\mathcal{A}$ is an unambiguous blind counter Büchi automaton. Then the ω-language $\mathrm{L}(\mathcal{A})$ is a Boolean combination of $\mathbf{\Pi}^0_2$-sets, hence a Borel $\mathbf{\Delta}^0_3$-set. It is actually an effective $\mathbf{\Delta}^0_3$-set, i.e. an arithmetical (lightface) Δ^0_3-set.

The overall structure of the construction is a variant of the powerset construction with an additional trimming. First we show how to split the space of configurations of a given unambiguous machine into finitely many regions (called clubs). Then we argue that the assumption of unambiguity implies, that after reading a prefix w of the input word σ, the machine cannot reach two distinct configurations from the same club — it would lead to two distinct accepting runs on a certain word $\sigma' \sqsupseteq w$. This means that it is enough to keep track of at most one run of the machine in each of the finitely many clubs. Based on that, we build a deterministic counter machine that stores all those finitely many runs in its memory.

10.1. Lasso patterns

We begin by recalling known structural properties of blind counter automata, namely *lasso patterns* that are used to *pump* runs of the automaton. The patterns lie at the core of the decidability algorithms for these automata. We will use these properties later to construct accepting runs of the machine under certain assumptions, which leads to the effectiveness of the provided translation.

Fix a Büchi blind counter automaton $\mathcal{A} = \langle K, \Sigma, \Delta, q_0, F\rangle$ with a set of states K, k-counters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, and a transition relation Δ. To avoid double indexing, we will denote a transition $(q, a, i_1, \ldots, i_k, q', j_1, \ldots, j_i)$ of $\mathcal{A}$ by (q, a, I, q', J) with $I = (i_1, \ldots, i_k) \in \{0,1\}^k$ and $J = (j_1, \ldots, j_k) \in \{-1, 0, 1\}^k$. Similarly, a configuration $(q, c_1, \ldots, c_k)$ of such a counter automaton can be written (q, τ) with $\tau = (c_1, \ldots, c_k) \in \mathbb{N}^k$.

A *lasso pattern* is a sequence of transitions $\big(\delta_i{=}(q_i, a_i, I_i, q'_i, J_i)\big)_{i=0,\ldots,\ell} \subseteq \Delta$ and a number $0 \le \ell' \le \ell$, such that the following conditions hold:

1. q_0 is the initial state of $\mathcal{A}$, and $I_0 = (0, 0, \ldots, 0)$,
2. for each $i = 0, 1, \ldots, \ell{-}1$ we have $q'_i = q_{i+1}$,
3. $q'_\ell = q_{\ell'}$,
4. for each $i = 0, 1, \ldots, \ell$ the sum $\sum_{j=0}^{i} J_j$ belongs to $\mathbb{N}^k$ (i.e. is coordinate-wise non-negative),
5. the sum $\sum_{j=\ell'}^{\ell} J_j$ also belongs to $\mathbb{N}^k$,
6. for each $i = 0, 1, \ldots, \ell$ and $c = 1, \ldots, k$ if $(I_i)_c = 1$ then $\sum_{j=0}^{i-1} (J_j)_c > 0$.

These conditions are meant to ensure that one can construct a run of $\mathcal{A}$ that uses consecutively the transitions from the lasso pattern. Notice that if $(I_i)_c = 0$ in the last item of the definition then the assumption of blindness guarantees that there is another transition in Δ with $(I_i)_c = 1$, so we don't need to restrict the sum $\sum_{j=0}^{i-1} (J_j)_c$ in that case.

We call ℓ the *length* of the lasso pattern and ℓ' is its *looping point*. A lasso pattern as above is *accepting* if for some $i = \ell', \ldots, \ell$ the state q_i is accepting. The definition of a lasso pattern is based on Problem 3.2 in [60]. The exact properties used in the definition of a lasso pattern are chosen in such a way to ensure the following remark.

Remark 10.3. If a blind counter Büchi automaton $\mathcal{A}$ has an accepting lasso pattern $(\delta_i)_{i=0,\dots,\ell} \subseteq \Delta$ with a looping point $0 \leq \ell' \leq \ell$ then $\mathrm{L}(\mathcal{A}) \neq \emptyset$.

Proof:
Let $u = a_0 a_1 \cdots a_{\ell'-1}$ and $w = a_{\ell'} a_{\ell'+1} \cdots a_\ell$. Then the ω-word $\alpha = uwww\cdots$ belongs to $\mathrm{L}(\mathcal{A})$ because one can construct an accepting run of $\mathcal{A}$ over this ω-word using the transitions of the assumed lasso pattern. □

The following theorem from [60, Section 3] implies decidability of the emptiness problem for blind counter Büchi automata by providing the converse implication.

Theorem 10.4. If $\mathcal{A}$ is a blind counter Büchi automaton such that $\mathrm{L}(\mathcal{A}) \neq \emptyset$ then $\mathcal{A}$ has an accepting lasso pattern of length bounded by a function computable[1] based on $\mathcal{A}$. As a consequence, it is decidable if $\mathrm{L}\big(\mathcal{A}\big)$ is empty.

10.2. Clubs of configurations

This section is devoted to an introduction of a technical concept used in the determinisation procedure: *clubs of configurations*. These are regions of the configuration space of a counter automaton that represent somehow similar behaviour of the machine. We will see later on that certain clubs that are *optimal* can be treated in a homogeneous way (Lemma 10.9); and moreover each club can be split into a finite family of optimal ones (Proposition 10.11).

Fix a counter automaton $\mathcal{A}$ with a set of states K and k-counters $\mathcal{C}_1, \dots, \mathcal{C}_k$. Let $N \in \mathbb{N}$ and $\gamma = \big(\gamma_1, \dots, \gamma_k\big)$ be a vector where each γ_i for $i = 1, \dots, k$ is either a natural number or the expression $(\geq N)$. Recall that we will denote the configurations of $\mathcal{A}$ by (q, τ) with $\tau = (c_1, \dots, c_k) \in \mathbb{N}^k$ being the vector of counter values. A *club* is a set of configurations of $\mathcal{A}$ of the form

$$[q, \gamma] =_{\mathrm{df}} \big\{(q, \tau) \mid \forall 1 \leq i \leq k.\ (\gamma_i = \tau_i \in \mathbb{N}) \vee (\gamma_i = (\geq N) \wedge \tau_i \geq N)\big\}. \tag{5}$$

The *dimension* of a club is the number of expressions $(\geq N)$ that appear in γ. Similarly, the value $N \in \mathbb{N}$ is the *threshold* of the club. The *minimal configuration* of a club $[q, \gamma]$ is the configuration (q, τ) where for $i = 1, \dots, k$ the coordinate τ_i equals γ_i when $\gamma_i \in \mathbb{N}$ and equals N otherwise. Notice that the minimal configuration of a club is the $\preceq$-least element of the club (see the definition of the simulation order on page 248 and Remark 2.3).

Let $[q, \gamma]$ be a club with threshold N and $M \geq N$ be a natural number. By $[q, \gamma\restriction_M]$ (called the *restriction* of $[q, \gamma]$ to the threshold M) we denote the club obtained from $[q, \gamma]$ by replacing each occurrence of $(\geq N)$ by $(\geq M)$. Notice that, as sets of configurations we have

$$[q, \gamma\restriction_M] \subseteq [q, \gamma]. \tag{6}$$

[1]The function is doubly-exponential, see the comment before Theorem 3.1 in [60].

10.3. Languages of clubs

Each club $[q, \gamma]$, seen as a set of configurations of $\mathcal{A}$, induces its ω-language $\mathrm{L}\big(\mathcal{A}, [q, \gamma]\big)$ being just the set theoretic union of all the ω-languages $\mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ for all configurations $(q, \tau) \in [q, \gamma]$. Although the notation might suggest that, we do not consider the possibility to treat clubs as configurations of a counter automaton and perform transitions on clubs — instead we execute the automaton from each of the single configurations $(q, \tau) \in [q, \gamma]$ and then take the union of these ω-languages.

We will now show how to decide if the ω-language of a club is empty, see Corollary 10.6 at the end of this subsection. For that, fix a k-blind counter Büchi automaton $\mathcal{A}$ and a club $[q, \gamma]$ of dimension d and threshold N.

Let (q, τ_0) be the minimal configuration of $[q, \gamma]$. Without loss of generality we can assume that $\gamma = \big(\gamma_1, \ldots, \gamma_{k'}, (\geq N), \ldots, (\geq N)\big)$ with the values $\gamma_1, \ldots, \gamma_{k'}$ being natural numbers and $k = k' + d$. Notice that by the choice of τ_0 we know that $\tau_0 = \big(\gamma_1, \ldots, \gamma_{k'}, N, \ldots, N\big)$.

Let $a_0, a_1, a_2 \in \Sigma$ be any (not necessarily distinct) fixed letters of the alphabet Σ. Let $\tau_0^1, \ldots, \tau_0^Z$ be a sequence of vectors in $\{0, 1\}^k$ such that $\sum_{j=1}^{Z} \tau_0^j = \tau_0$, with $Z \in \mathbb{N}$ being the maximal of the coordinates of τ_0. Let $\tau_1 = (0, \ldots, 0, 1, \ldots, 1)$ be a vector with k' zeros followed by d ones.

Consider the blind counter Büchi automaton denoted $\big([q, \gamma] \cdot \mathcal{A}\big)$ depicted on Figure 2. This automaton first reads a sequence of letters a_0 increasing all the counters to the exact values given by τ_0 (using the vectors τ_0^j for that); then it can arbitrarily many times read a_1 and increase the counters numbered $k'+1, k'+2, \ldots, k$, i.e. those corresponding to the value $(\geq N)$ in γ; and then it reads a_2 and moves to a copy of the automaton $\mathcal{A}$ into the state q.

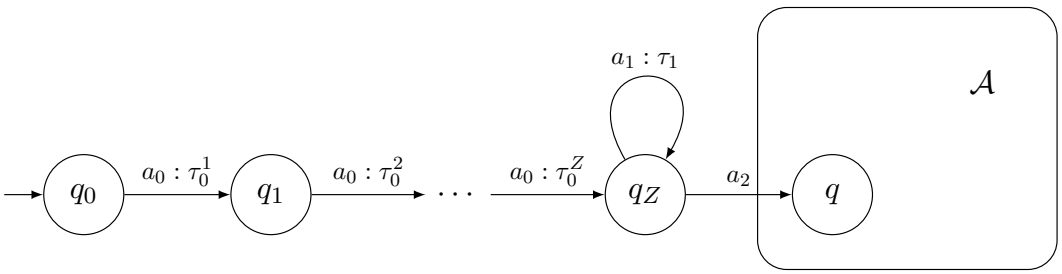

Figure 2. The automaton denoted $\big([q, \gamma] \cdot \mathcal{A}\big)$ that checks non-emptiness of the set $\mathrm{L}\big(\mathcal{A}, [q, \gamma]\big)$.

Proposition 10.5. Fix a blind counter Büchi automaton $\mathcal{A}$ and a club $[q, \gamma]$ with threshold N. Then one can effectively compute a number $M \geq N$ such that the following conditions are equivalent:

1. $\mathrm{L}\big(\mathcal{A}, [q, \gamma]\big) \neq \emptyset$,
2. $\mathrm{L}\big([q, \gamma] \cdot \mathcal{A}\big) \neq \emptyset$,
3. $\mathrm{L}\big(\mathcal{A}, (q, \tau_0')\big) \neq \emptyset$ for (q, τ_0') the minimal configuration of $[q, \gamma\restriction_M]$,
4. $\mathrm{L}\big(\mathcal{A}, [q, \gamma\restriction_M]\big) \neq \emptyset$.

Proof:
Let us denote by $\mathcal{A}'$ the automaton $\big([q,\gamma]\cdot\mathcal{A}\big)$. Let M_ℓ be the bound on the length of a lasso pattern for $\mathcal{A}'$ given by Theorem 10.4 and take $M =_{\mathrm{df}} N + M_\ell$.

We start with the implication $1 \Rightarrow 2$. Assume that $\alpha \in \mathrm{L}\big(\mathcal{A},(q,\tau)\big)$ for some $(q,\tau) \in [q,\gamma]$. Let M' be the maximal coordinate of τ. Consider $\alpha' = a_0^Z a_1^{M'} a_2 \alpha$, where Z is taken as in the construction of $\mathcal{A}'$. Let (q,τ') be the configuration of $\mathcal{A}'$ reached after reading the prefix $a_0^Z a_1^{M'} a_2$ of α'. By the choice of M', we know that $\tau' \succeq \tau$. Therefore, by Remark 2.3 we have $\alpha \in \mathrm{L}\big(\mathcal{A},(q,\tau')\big)$. Thus, there exists an accepting run of $\mathcal{A}'$ over α' and $\mathrm{L}\big(\mathcal{A}'\big) \neq \emptyset$.

Now consider the implication $2 \Rightarrow 3$. Theorem 10.4 together with the choice of M_ℓ guarantee that if $\mathrm{L}\big(\mathcal{A}'\big) \neq \emptyset$ then there exists an accepting lasso pattern of $\mathcal{A}'$. Let $\big(\delta_i{=}(q_i,a_i,I_i,q_i',J_i)\big)_{i=0,\dots,\ell} \subseteq \Delta$ and $0 \le \ell' \le \ell$ be such a lasso pattern.

As the states $q_0,\dots,q_Z$ of $\mathcal{A}'$ are not accepting, it means that the state q must appear among $(q_i)_{i\le\ell}$. Let j be the minimal index such that $q_j = q$. Since $q_0,\dots,q_Z$ are not reachable from the copy of $\mathcal{A}$ within $\mathcal{A}'$, we know that $Z < j \le \ell'$. Consider the run r of $\mathcal{A}'$ over an ω-word α that is obtained by following the transitions of the considered lasso pattern. We know that $r(j) = (q,\tau)$ for a certain configuration (q,τ) and the considered state q. The rest of this run is accepting, witnessing that $\mathrm{L}\big(\mathcal{A},(q,\tau)\big) \neq \emptyset$. However, by the construction of $\mathcal{A}'$ we know that $\tau = \tau_0 + \tau_1 \cdot (j{-}Z{-}1)$.

Recall that $M = N + M_\ell$, which implies that τ_0' from Item 3 of the statement has the form $\tau_0{+}\tau_1{\cdot}M_\ell$. As $j \le M_\ell$, we know that $(q,\tau) \preceq (q,\tau_0')$ and Remark 2.3 implies that $\mathrm{L}\big(\mathcal{A},(q,\tau_0')\big) \neq \emptyset$.

The implication $3 \Rightarrow 4$ is obvious, as $(q,\tau_0') \in [q,\gamma{\restriction}_M]$. Similarly, $4 \Rightarrow 1$ is also clear because $[q,\gamma{\restriction}_M] \subseteq [q,\gamma]$, see (6). □

Corollary 10.6. It is decidable for a club $[q,\gamma]$ of $\mathcal{A}$ if the set of ω-words $\mathrm{L}\big(\mathcal{A},[q,\gamma]\big)$ is empty.

10.4. Optimal clubs

A club $[q,\gamma]$ is called *trivial* if $\mathrm{L}\big(\mathcal{A},[q,\gamma]\big) = \emptyset$; otherwise $[q,\gamma]$ is called *non trivial*. Proposition 10.5 implies that each non-trivial club $[q,\gamma]$ can be restricted to another non-trivial club $[q,\gamma{\restriction}_M] \subseteq [q,\gamma]$ such that already the ω-language $\mathrm{L}\big(\mathcal{A},(q,\tau_0')\big)$ is non empty for τ_0' the minimal configuration of $[q,\gamma{\restriction}_M]$. In the latter part of the construction we will be interested in such *optimal* clubs.

Let $[q,\gamma]$ be a club with the minimal configuration (q,τ). Then $[q,\gamma]$ is called *optimal* if the following implication holds:

$$\mathrm{L}\big(\mathcal{A},[q,\gamma]\big) \neq \emptyset \implies \mathrm{L}\big(\mathcal{A},(q,\tau)\big) \neq \emptyset.$$

Notice that each club of dimension 0, as a set of configurations, is a singleton and therefore it is optimal by the definition. Obviously, an optimal non-trivial club with the minimal configuration (q,τ) satisfies $\mathrm{L}\big(\mathcal{A},(q,\tau)\big) \neq \emptyset$.

Remark 10.7. By Theorem 10.4 and Corollary 10.6 it is decidable whether a given club is trivial and whether it is optimal.

Lemma 10.8. If $[q,\gamma] \supseteq [q,\gamma']$ are two clubs and $[q,\gamma]$ is optimal then also $[q,\gamma']$ is optimal.

Proof:
By the assumption that $[q,\gamma] \supseteq [q,\gamma']$, the minimal configuration (q,τ) of $[q,\gamma]$ is $\preceq$-smaller than the minimal configuration (q,τ') of $[q,\gamma']$. Thus, Remark 2.3 implies that if the ω-language $\mathrm{L}\big(\mathcal{A},(q,\tau)\big)$ is non empty then also $\mathrm{L}\big(\mathcal{A},(q,\tau')\big)$ is non empty. □

The following lemma shows that each club can be made optimal by increasing its threshold.

Lemma 10.9. If $[q,\gamma]$ is a club with threshold N then there exists $M \geq N$ such that the club $[q,\gamma\restriction_M]$ is optimal. Moreover, the value of M can be effectively computed based on $\mathcal{A}$ and a representation of $[q,\gamma]$.

Proof:
It is enough to take the value M from Proposition 10.5. The implication $4 \Rightarrow 3$ of the proposition implies that $[q,\gamma\restriction_M]$ is always optimal. □

If one doesn't care about the computability of the value M in Lemma 10.9, then one can use directly Remark 2.3, as expressed by the following remark.

Remark 10.10. Every machine model satisfying Remark 2.3 has the following property: if $[q,\gamma]$ is a club with threshold N then there exists $M \geq N$ such that the club $[q,\gamma\restriction_M]$ is optimal.

Proof:
Let $\mathcal{A}$ be a machine of the considered model that satisfies Remark 2.3. If $\mathrm{L}\big(\mathcal{A},[q,\gamma]\big)$ is empty then the club is already optimal for $M = N$. Otherwise, let $(q,\tau) \in [q,\gamma]$ be a configuration such that $\mathrm{L}\big(\mathcal{A},(q,\tau)\big) \neq \emptyset$. Let M be the maximal coordinate of τ and let (q,τ_0) be the minimal configuration of $[q,\gamma\restriction_M]$ (it is obtained from γ by replacing each coordinate $(\geq N)$ by M). Then $(q,\tau) \preceq (q,\tau_0)$ and therefore $\mathrm{L}\big(\mathcal{A},(q,\tau_0)\big) \neq \emptyset$, which means that the club $[q,\gamma\restriction_M]$ is optimal. □

The following proposition is the technical core of the construction. We believe that it is of independent interest.

Proposition 10.11. Let $[q,\gamma]$ be a club. Then $[q,\gamma]$, as a set of configurations, can be written as a finite pairwise disjoint union of optimal clubs. Moreover, such a decomposition can be computed effectively.

Proof:
The proof is inductive on the dimension d of $[q,\gamma]$. Since each club of dimension 0 is optimal, the thesis holds for $d = 0$. Assume the thesis for all the clubs of dimensions at most d and consider a club $[q,\gamma]$ of dimension $d > 0$ with threshold N. Without loss of generality we can assume that $\gamma = \big(\gamma_1,\ldots,\gamma_{k'},(\geq N),\ldots,(\geq N)\big)$ — the total number k of coordinates of γ is $k'+d$ and all the values $\gamma_1,\ldots,\gamma_{k'}$ are natural numbers.

Apply Lemma 10.9 to $[q,\gamma]$ to obtain $M \geq N$ such that the restriction $[q,\gamma\restriction_M]$ is optimal. Notice that $\gamma\restriction_M = \big(\gamma_1,\ldots,\gamma_{k'},(\geq M),\ldots,(\geq M)\big)$.

Let F be the set of clubs of the form $[q, \gamma']$ where γ' equals γ on coordinates $1,\ldots, k'$ and for each coordinate $i = k'+1,\ldots, k$ either $\gamma'_i = (\geq M)$ or γ'_i is a natural number satisfying $N \leq \gamma'_i < M$. Since there is exactly 2^d choices for a set of coordinates with $(\geq M)$ in γ, and for each such choice there is only finitely many clubs with $(\geq M)$ on exactly those coordinates in F, the set F is finite.

Claim 10.12. The clubs in F are pairwise disjoint and $[q, \gamma] = \bigcup F$.

The disjointness follows directly from the construction. For the union, it is enough to notice that each $[q, \gamma'] \in F$ satisfies $[q, \gamma'] \subseteq [q, \gamma]$ and each $(q, \tau) \in [q, \gamma]$ can be found in one of the clubs of F.

Notice that $[q, \gamma\restriction_M] \in F$ — it corresponds to the choice of all d coordinates being $(\geq M)$. Let $F' = F \setminus \{[q, \gamma\restriction_M]\}$. Observe that if $[q, \gamma'] \in F'$ then the dimension of $[q, \gamma']$ is at most $d-1$. Thus, we can apply the inductive assumption to each club in F' and take the union of all these clubs. Let F'' be the set of the clubs obtained this way. The clubs in F'' are pairwise disjoint, optimal, and disjoint from $[q, \gamma\restriction_M]$. Thus,

$$[q, \gamma] = \bigcup F'' \cup [q, \gamma\restriction_M],$$

is a decomposition of $[q, \gamma]$ into finitely many pairwise disjoint optimal clubs.

For the effectiveness of the above construction, it is enough to observe that the bound M given by Lemma 10.9 is effective and the rest of the construction is just a recursive application of the same procedure. □

10.5. Unambiguous automata

Fix an unambiguous blind counter Büchi automaton $\mathcal{A}$ with a set of states K and k-blind counters $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Our goal is to show that such an automaton cannot simultaneously (reading a finite word) reach two distinct configurations belonging to the same optimal non-trivial club.

Lemma 10.13. Let $[q, \gamma]$ be an optimal club and $w \in \Sigma^*$ be a finite word. Assume that for $i = 1, 2$ there exists a configuration $(q, \tau_i) \in [q, \gamma]$ that can be reached while reading w from the initial configuration. If $\tau_1 \neq \tau_2$ then the club $[q, \gamma]$ is trivial.

Similarly, if a configuration (q, τ) can be reached by two distinct runs while reading a finite word w from the initial configuration then $\mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ is empty.

Proof:
Consider the first statement of the lemma. Assume to the contrary that $[q, \gamma]$ is non trivial. Let (q, τ) be the minimal configuration of $[q, \gamma]$. By the assumption of optimality, the ω-language $\mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ is non-empty, let $\alpha \in \mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ be an ω-word witnessing that. Since both (q, τ_1) and (q, τ_2) belong to $[q, \gamma]$, we know that $(q, \tau) \preceq (q, \tau'_1)$ and $(q, \tau) \preceq (q, \tau_2)$. Thus, Remark 2.3 implies that

$$\alpha \in \mathrm{L}\big(\mathcal{A}, (q, \tau_1)\big) \cap \mathrm{L}\big(\mathcal{A}, (q, \tau_2)\big).$$

This gives a contradiction because it means that $\mathcal{A}$ has two distinct accepting runs over $w \cdot \alpha$.

For the second statement of the lemma, the proof is analogous: if $\alpha \in \mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ then $\mathcal{A}$ has two distinct accepting runs over $w \cdot \alpha$. □

Notice that we can easily extend the above lemma to p-unambiguous blind counter Büchi automaton, for an integer $p \geq 1$. Let us call a blind counter Büchi automaton p-unambiguous if every ω-word over the input alphabet has at most p accepting runs. Then we can state the following result which holds for a p-unambiguous blind counter Büchi automaton. The proof is similar to the above one.

Lemma 10.14. Let $[q, \gamma]$ be an optimal club and $w \in \Sigma^*$ be a finite word. Assume that for $i = 1, 2, \ldots, p, p+1$ there exists a configuration $(q, \tau_i) \in [q, \gamma]$ that can be reached while reading w from the initial configuration. If for all $i, j \in [1, p+1]$ $i \neq j \to \tau_i \neq \tau_j$ then the club $[q, \gamma]$ is trivial.

Similarly, if a configuration (q, τ) can be reached by $p+1$ distinct runs while reading a finite word w from the initial configuration then $\mathrm{L}\big(\mathcal{A}, (q, \tau)\big)$ is empty.

10.6. The machine $\mathcal{M}$

The construction of the machine $\mathcal{M}$ is based on the above Lemma 10.13. Thanks to it, we know that in the naive powerset construction, it is enough to remember at most one configuration of $\mathcal{A}$ for each optimal club. Moreover, by Proposition 10.11, one can split the whole configuration space of $\mathcal{A}$ into finitely many pairwise disjoint optimal clubs.

Let F be a (finite) family of these clubs. Let the machine $\mathcal{M}$ store at most one configuration (q, τ) for each club $[q, \gamma] \in F$ (of course we require that $(q, \tau) \in [q, \gamma]$). Notice that a finite family of counters is enough to store all these configurations at once.

Upon reading a successive letter, $\mathcal{M}$ can update all the stored configurations according to all the possible transitions of $\mathcal{A}$. Whenever two distinct configurations obtained that way belong to the same club of F (we call such a situation a *collision*), $\mathcal{M}$ can *discard* both these configurations, because the club is guaranteed to be trivial.

More precisely, there are in fact two possible scenarios for a configuration (q', τ') to be discarded because of a collision. The first case is that there might be a distinct configuration $(q', \tau'') \neq (q', \tau')$ reachable by one of the simulated transitions, such that both (q', τ') and (q', τ'') belong to a single club $[q', \gamma'] \in F$. The second case is that the same configuration (q', τ') can also be reached by a different transition from one of the previously stored configurations. However, in both cases Lemma 10.13 applies, guaranteeing that $\mathrm{L}\big(\mathcal{A}, (q', \tau')\big) = \emptyset$.

Due to the policy of discarding, the invariant of storing at most one configuration per club from F is preserved.

Finally, $\mathcal{M}$ uses a Muller condition to check if any of the runs of $\mathcal{A}$ that are simulated in parallel, turned out to be accepting.

See Appendix A for a precise construction of $\mathcal{M}$ and a discussion of exact computational features used in its construction.

This concludes the proof of Theorem 10.1.

Remark 10.15. Notice that Theorem 10.1 can be extended to the case of a p-unambiguous blind counter Büchi automaton, using Lemma 10.14 instead of Lemma 10.13. We do not enter into the details which are left to the reader. The ideas are identical but the machine $\mathcal{M}$ will be simply more complicated since it has to store at most p configurations (q, τ) for each club $[q, \gamma] \in F$.

Theorem 10.16. If $\mathcal{A}$ is a p-unambiguous blind counter Büchi automaton, for an integer $p \geq 1$, then $\mathrm{L}(\mathcal{A})$ can be recognised by a deterministic Muller counter machine $\mathcal{M}$ that allows copying, hence by a deterministic Muller Turing machine. Moreover, the translation from $\mathcal{A}$ to $\mathcal{M}$ is effective.

Corollary 10.17. Assume that $\mathcal{A}$ is a p-unambiguous blind counter Büchi automaton, for an integer $p \geq 1$. Then the ω-language $\mathrm{L}(\mathcal{A})$ is a Boolean combination of $\mathbf{\Pi}^0_2$-sets, hence a Borel $\mathbf{\Delta}^0_3$-set. It is actually an effective $\mathbf{\Delta}^0_3$-set, i.e. an arithmetical (lightface) Δ^0_3-set.

11. Concluding remarks

We have proved that the Wadge hierarchy of Petri nets ω-languages, and even of ω-languages in the class $\mathbf{r}\text{-}\mathbf{BCL}(4)_\omega$, is equal to the Wadge hierarchy of effective analytic sets, and that it is highly undecidable to determine the topological complexity of a Petri net ω-language. Based on the constructions used in the proofs of the above results, we have also shown that the equivalence and the inclusion problems for ω-languages of Petri nets are Π^1_2-complete, hence highly undecidable. In some sense, from the two points of view of the topological complexity and of highly undecidable problems, our results show that, in contrast with the finite behaviour, the infinite behaviour of non-deterministic Petri nets is closer to the infinite behaviour of Turing machines than to that of finite automata.

As further developments showing the inherent complexity of the model, we have proved that the determinacy of Wadge games between two players in charge of ω-languages of Petri nets is equivalent to the (effective) analytic determinacy, which is known to be a large cardinal assumption, and thus is not provable in the axiomatic system **ZFC**. We have also provided a Petri net whose ω-language is either a Borel $\mathbf{\Pi}^0_2$-set or a non-Borel set, depending on the model of **ZFC** under consideration.

Additionally, we have shown that in fact only one counter is enough to obtain a single $\mathbf{\Sigma}^1_1$-complete ω-language, i.e. an ω-language of maximal topological complexity among those recognisable by Petri nets. All these results imply that non-deterministic Petri nets are expressively stronger than any reasonable model of deterministic or unambiguous machines.

We have also studied the expressive power of unambiguous Petri nets. As it turns out, they admit a determinisation procedure. As a consequence, the topological complexity of their ω-languages is low in the Borel hierarchy (they all are $\mathbf{\Delta}^0_3$ sets).

It remains open for further study to determine the Borel and Wadge hierarchies of ω-languages accepted by automata with less than four blind counters. In particular, it then remains open to determine whether there exist some ω-languages accepted by 1-blind-counter automata which are Borel of rank greater than 3.

Acknowledgements The authors would like to cordially thank Georg Zetzsche for his suggestions about decidability issues for blind counter automata.

The authors also thank anonymous referees of this paper and of the papers incorporated in this journal version — their suggestions allowed to improve the presentation.

References

[1] Thomas W. Automata on infinite objects. In: van Leeuwen J (ed.), Handbook of Theoretical Computer Science, volume B, Formal models and semantics, Elsevier, 1990 pp. 135–191. doi:10.1016/B978-0-444-88074-1.50009-3.

[2] Staiger L. ω-languages. In: Handbook of formal languages, Vol. 3, pp. 339–387. Springer, Berlin, 1997. doi:10.1007/978-3-642-59126-6.

[3] Perrin D, Pin JÉ. Infinite words, automata, semigroups, logic and games, volume 141 of *Pure and Applied Mathematics*. Elsevier, 2004. ISBN:978-0-12-532111-2, 0079-8169.

[4] Engelfriet J, Hoogeboom HJ. X-automata on ω-words. *Theoretical Computer Science*, 1993. **110**(1):1–51. doi:10.1016/0304-3975(93)90349-X.

[5] Cohen R, Gold A. ω-computations on Turing machines. *Theoretical Computer Science*, 1978. **6**:1–23. doi:0.1016/0304-3975(78)90002-6.

[6] Valk R. Infinite behaviour of Petri nets. *Theoretical computer science*, 1983. **25**(3):311–341. doi:10.1016/0304-3975(83)90115-9.

[7] Staiger L. Recursive Automata on Infinite Words. In: Enjalbert P, Finkel A, Wagner KW (eds.), Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science, STACS 93, Würzburg, Germany, February 25-27, 1993, volume 665 of *Lecture Notes in Computer Science*. Springer, 1993 pp. 629–639. doi:10.1007/3-540-56503-5_62.

[8] Lescow H, Thomas W. Logical specifications of infinite computations. In: de Bakker JW, de Roever WP, Rozenberg G (eds.), A Decade of Concurrency, volume 803 of *Lecture Notes in Computer Science*. Springer, 1994 pp. 583–621. doi:10.1007/3-540-58043-3_29.

[9] Landweber L. Decision problems for ω-automata. *Mathematical Systems Theory*, 1969. **3**(4):376–384. doi:10.1007/BF01691063.

[10] Wadge W. Reducibility and determinateness in the Baire space. Ph.D. thesis, University of California, Berkeley, 1983.

[11] Wagner K. On ω-regular sets. *Information and Control*, 1979. **43**(2):123–177. doi:10.1016/S0019-9958(79)90653-3.

[12] Selivanov V. Fine Hierarchy of Regular ω-Languages. *Theoretical Computer Science*, 1998. **191**:37–59. doi:10.1007/3-540-59293-8_201.

[13] Selivanov V. Wadge Reducibility and Infinite Computations. *Special Issue on Intensional Programming and Semantics in honour of Bill Wadge on the occasion of his 60th cycle, Mathematics in Computer Science*, 2008. **2**(1):5–36. doi:10.1007/s11786-008-0042-x.

[14] Selivanov V. Fine hierarchies and m-reducibilities in theoretical computer science. *Theoretical Computer Science*, 2008. **405**(1-2):116–163. doi:10.1016/j.tcs.2008.06.031.

[15] Simonnet P. Automates et théorie descriptive. Ph.D. thesis, Université Paris VII, 1992.

[16] Duparc J. A hierarchy of deterministic context free ω-languages. *Theoretical Computer Science*, 2003. **290**(3):1253–1300. doi:10.1016/S0304-3975(02)00567-4.

[17] Duparc J, Finkel O, Ressayre JP. Computer science and the fine structure of Borel sets. *Theoretical Computer Science*, 2001. **257**(1-2):85–105. doi:10.1016/S0304-3975(00)00111-0.

[18] Finkel O. An effective extension of the Wagner hierarchy to blind counter automata. In: Proceedings of Computer Science Logic, 15th International Workshop, CSL 2001, volume 2142 of *Lecture Notes in Computer Science*. Springer, 2001 pp. 369–383. doi:10.1007/3-540-44802-0_26.

[19] Finkel O. Topological properties of omega context free languages. *Theoretical Computer Science*, 2001. **262**(1-2):669–697. doi:10.1016/S0304-3975(00)00405-9.

[20] Finkel O. Borel Ranks and Wadge Degrees of Context Free Omega Languages. *Mathematical Structures in Computer Science*, 2006. **16**(5):813–840. URL `https://hal.archives-ouvertes.fr/hal-00139169`.

[21] Finkel O. Wadge Degrees of Infinitary Rational Relations. *Special Issue on Intensional Programming and Semantics in honour of Bill Wadge on the occasion of his 60th cycle, Mathematics in Computer Science*, 2008. **2**(1):85–102. doi:10.1007/s11786-008-0045-7.

[22] Selivanov V. Wadge Degrees of ω-Languages of Deterministic Turing Machines. *RAIRO-Theoretical Informatics and Applications*, 2003. **37**(1):67–83. doi:10.1051/ita:2003008.

[23] Esparza J. Decidability and complexity of Petri net problems, an introduction. *Lectures on Petri Nets I: Basic Models*, 1998. pp. 374–428. doi:10.1007/3-540-65306-6_20.

[24] Rozenberg G. Lectures on concurrency and Petri nets: advances in Petri nets, volume 3098. Springer Verlag, 2004. doi:10.1007/b98282.

[25] Haddad S. Decidability and Complexity of Petri Net Problems. In: Diaz M (ed.), Petri Nets: Fundamental Models, Verification and Applications, pp. 87–122. Wiley-ISTE, 2009. doi:10.1002/9780470611647.ch4.

[26] Hopcroft J, Pansiot JJ. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 1979. **8**(2):135–159. doi:10.1016/0304-3975(79)90041-0.

[27] Leroux J, Schmitz S. Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019. IEEE, 2019 pp. 1–13. doi:10.1109/LICS.2019.8785796.

[28] Greibach S. Remarks on Blind and Partially Blind One Way Multicounter Machines. *Theoretical Computer Science*, 1978. **7**:311–324. doi:10.1016/0304-3975(78)90020-8.

[29] Carstensen H. Infinite behaviour of deterministic Petri nets. In: Proceedings of Mathematical Foundations of Computer Science 1988, volume 324 of *Lecture Notes in Computer Science*. Springer, 1988 pp. 210–219. doi:10.1007/BFb0017144.

[30] Duparc J, Finkel O, Ressayre JP. The Wadge hierarchy of Petri nets ω-languages. In: Special Volume in Honor of Victor Selivanov at the occasion of his sixtieth birthday, Logic, computation, hierarchies, volume 4 of *Ontos Math. Log.*, pp. 109–138. De Gruyter, Berlin, 2014. URL `https://hal.archives-ouvertes.fr/hal-00743510`.

[31] Finkel O, Skrzypczak M. On the topological complexity of ω-languages of non-deterministic Petri nets. *Information Processing Letters*, 2014. **114**(5):229–233. arXiv:1401.6835 [cs.LO].

[32] Finkel O. Wadge Degrees of ω-Languages of Petri Nets, 2017. Preprint available from arXiv:1712.07945.

[33] Finkel O. On the High Complexity of Petri Nets ømega-Languages. In: Janicki R, Sidorova N, Chatain T (eds.), Application and Theory of Petri Nets and Concurrency - 41st International Conference, PETRI NETS 2020, Paris, France, June 24-25, 2020, Proceedings, volume 12152 of *Lecture Notes in Computer Science*. Springer, 2020 pp. 69–88. doi:10.1007/978-3-030-51831-8_4.

[34] Skrzypczak M. Büchi VASS Recognise $\mathbf{\Sigma}^1_1$-complete ω-languages. In: Proceedings of Reachability Problems - 12th International Conference, RP 2018, Marseille, France, September 24-26, volume 11123 of *Lecture Notes in Computer Science*. Springer, 2018 pp. 133–145. arXiv:1708.09658 [cs.FL].

[35] Finkel O. Ambiguity in omega context free languages. *Theoretical Computer Science*, 2003. **301**(1-3):217–270. doi:10.1016/S0304-3975(02)00584-4.

[36] Finkel O. Ambiguity of ω-Languages of Turing machines. *Logical Methods in Computer Science*, 2014. **10**(3:12):1–18. URL `https://hal.archives-ouvertes.fr/hal-00735050`.

[37] Colcombet T. Forms of Determinism for Automata (Invited Talk). In: Dürr C, Wilke T (eds.), 29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France, volume 14 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012 pp. 1–23. doi:10.4230/LIPIcs.STACS.2012.1.

[38] Stearns RE, III HBH. On the Equivalence and Containment Problems for Unambiguous Regular Expressions, Regular Grammars and Finite Automata. *SIAM Journal on Computing*, 1985. **14**(3):598–611. doi:10.1137/0214044.

[39] Mottet A, Quaas K. The Containment Problem for Unambiguous Register Automata. In: Niedermeier R, Paul C (eds.), 36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany, volume 126 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019 pp. 53:1–53:15. doi:10.4230/LIPIcs.STACS.2019.53.

[40] Carton O, Michel M. Unambiguous Büchi Automata. *Theoretical Computer Science*, 2003. **297**(1–3):37–81. doi:10.1016/S0304-3975(02)00618-7.

[41] Carayol A, Löding C, Niwiński D, Walukiewicz I. Choice functions and well-orderings over the infinite binary tree. *Central European Journal of Mathematics*, 2010. **8**:662–682. doi:10.2478/s11533-010-0046-z.

[42] Finkel O, Simonnet P. Topology and ambiguity in omega context free languages. *Bulletin of the Belgian Mathematical Society*, 2003. **10**(5):707–722. ArXiv:0801.0533.

[43] Finkel O, Simonnet P. On Recognizable Tree Languages Beyond the Borel Hierarchy. *Fundamenta Informaticae*, 2009. **95**(2-3):287–303. arXiv:0909.0393 [math.LO].

[44] Rabinovich A, Tiferet D. Degrees of Ambiguity of Büchi Tree Automata. In: Chattopadhyay A, Gastin P (eds.), 39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India, volume 150 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019 pp. 50:1–50:14. doi:10.4230/LIPIcs.FSTTCS.2019.50.

[45] Moschovakis YN. Descriptive set theory. North-Holland Publishing Co., Amsterdam, 1980. doi:10.2307/2273241.

[46] Kechris AS, Marker D, Sami RL. Π^1_1 Borel sets. *Journal of Symbolic Logic*, 1989. **54**(3):915–920. doi:10.2307/2274751.

[47] Duparc J. Wadge hierarchy and Veblen hierarchy: Part 1: Borel sets of finite rank. *Journal of Symbolic Logic*, 2001. **66**(1):56–86. doi:10.2307/2694911.

[48] Kechris AS. Classical descriptive set theory. Springer-Verlag, New York, 1995. ISBN:0-387-94374-9.

[49] Finkel O. Borel hierarchy and omega context free languages. *Theoretical Computer Science*, 2003. **290**(3):1385–1405. doi:10.1016/S0304-3975(02)00042-7.

[50] Finkel O. The Determinacy of Context-Free Games. *The Journal of Symbolic Logic*, 2013. **78**(4):1115–1134. arXiv:1312.3412 [cs.LO].

[51] Jech T. Set theory, third edition. Springer, 2002. doi:10.1007/3-540-44761-X.

[52] Louveau A, Saint-Raymond J. The strength of Borel Wadge determinacy. In: Cabal Seminar 81–85, volume 1333 of *Lecture Notes in Mathematics*, pp. 1–30. Springer, 1988. ISBN:978-3-540-45896-8.

[53] Harrington L. Analytic determinacy and $0^\sharp$. *Journal of Symbolic Logic*, 1978. **43**(4):685–693. doi:10.2307/2273508.

[54] Finkel O. The complexity of infinite computations in models of set theory. *Logical Methods in Computer Science*, 2009. **5**(4:4):1–19. doi:10.2168/LMCS-5(4:4)2009.

[55] Finkel O. Highly Undecidable Problems for Infinite Computations. *Theoretical Informatics and Applications*, 2009. **43**(2):339–364. URL `http://eudml.org/doc/250587`.

[56] Odifreddi P. Classical Recursion Theory, Vol I, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1989. ISBN:9780444894830, 9780080886596.

[57] Castro J, Cucker F. Nondeterministic ω-Computations and the Analytical Hierarchy. *Journal Math. Logik und Grundlagen d. Math*, 1989. **35**:333–342. doi:10.1002/malq.19890350406.

[58] Böhm S, Göller S, Halfon S, Hofman P. On Büchi One-Counter Automata. In: Vollmer H, Vallée B (eds.), 34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany, volume 66 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017 pp. 14:1–14:13. doi:10.4230/LIPIcs.STACS.2017.14.

[59] Hummel S, Skrzypczak M. The Topological Complexity of MSO+U and Related Automata Models. *Fundamenta Informaticae*, 2012. **119**(1):87–111. doi:10.3233/FI-2012-728.

[60] Habermehl P. On the complexity of the linear-time μ-calculus for Petri Nets. In: Azéma P, Balbo G (eds.), Application and Theory of Petri Nets 1997. Springer Berlin Heidelberg, 1997 pp. 102–116. doi:10.1007/3-540-63139-9_32.

[61] Selivanov V. Fine Hierarchy of Regular ω-Languages. In: Proceedings of the International Joint Conference on the Theory and Practice of Software Development TAPSOFT-95, in Aarhus, Denmark, volume 915 of *Lecture Notes in Computer Science*. Springer, 1995 pp. 277–287. doi:10.1007/3-540-59293-8_201.

[62] Selivanov V. Wadge Degrees of ω-Languages of Deterministic Turing Machines. In: Proceedings of the International Conference STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, volume 2607 of *Lecture Notes in Computer Science*. Springer, 2003 pp. 97–108. doi:10.1007/3-540-36494-3_10.

[63] Selivanov V. Fine Hierarchy of Regular Aperiodic omega-Languages. *International Journal of Foundations of Computer Science*, 2008. **19**(3):649–675. doi:10.1142/S0129054108005875.

[64] Finkel O. Wadge Hierarchy of omega context free languages. *Theoretical Computer Science*, 2001. **269**(1–2):283–315. doi:10.1016/S0304-3975(01)00008-1.

[65] Minsky ML. Recursive Unsolvability of Post's Problem of "Tag" and other Topics in Theory of Turing Machines. *Annals of Mathematics*, 1961. **74**(3):437–455. doi:10.2307/1970290.

[66] Mayr R. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 2003. **297**(1):337 – 354. doi:10.1016/S0304-3975(02)00646-1.

A. Construction of the machine $\mathcal{M}$ from Section 10

In this appendix we provide a precise construction of a real time counter machine $\mathcal{M}$ with zero tests and copying, as stated in Theorem 10.1. We assume that an unambiguous blind counter Büchi automaton $\mathcal{A}$ is fixed.

A.1. Simulating transitions

First we show how it can represent single configurations of $\mathcal{A}$ and simulate transitions.

Let $[q, \gamma_0] = \big[q, \big((\geq 0), (\geq 0), \ldots, (\geq 0)\big)\big]$ be a maximal club of dimension k (recall that k is the number of blind counters of the input automaton $\mathcal{A}$). Notice that $\bigcup_{q \in K} [q, \gamma_0]$ is the set of all configurations of $\mathcal{A}$. Apply Proposition 10.11 to each of the clubs $[q, \gamma_0]$ obtaining a finite set of clubs F_q. Let $F =_{\mathrm{df}} \bigcup_{q \in K} F_q$. Then F is a finite set of pairwise disjoint optimal clubs and $\bigcup F$ is the set of all configurations of $\mathcal{A}$. By further splitting the clubs in F and applying Lemma 10.8, we can ensure that all the clubs in F share the same threshold N — it is enough to take as N the maximal threshold of the clubs in F and then perform a similar splitting as in the proof of Proposition 10.11. Clearly we can ensure that $N > 0$. Notice that the set F can be effectively computed based on $\mathcal{A}$ — it is enough to use Proposition 10.11 for that.

For each club $[q, \gamma] \in F$ the machine $\mathcal{M}$ has a separate set of k counters, denoted $\mathcal{C}_1^{[q,\gamma]}, \ldots, \mathcal{C}_k^{[q,\gamma]}$. This means that $\mathcal{M}$ has $k \cdot |F|$ counters in total. A configuration $(q, \tau) \in [q, \gamma] \in F$ is represented in these counters by subtracting N from each of the counter values, i.e. for $j = 1, \ldots, k$ the counter $\mathcal{C}_j^{[q,\gamma]}$ stores the value $\max(\tau_j - N, 0)$. Notice that if the jth coordinate of γ equals $(\geq N)$ then $\tau_j - N \geq 0$ and our way of storing the value is exact. On the other hand, if γ_j is a natural number then $\tau_j = \gamma_j$, because $(q, \tau) \in [q, \gamma]$. This means that in this case the value τ_j (even if smaller than N) is determined by γ and therefore known.

Now we will show how the machine $\mathcal{M}$ can simulate a transition $a : (q, \tau) \mapsto_{\mathcal{A}} (q', \tau')$ of $\mathcal{A}$. Assume that $(q, \tau) \in [q, \gamma] \in F$ and the configuration (q, τ) is stored in the counters $\mathcal{C}_1^{[q,\gamma]}, \ldots, \mathcal{C}_k^{[q,\gamma]}$ of $\mathcal{M}$. First notice that based on γ the machine $\mathcal{M}$ can decide if the transition is possible at all, i.e. if the non-negativity conditions of the transition are met by τ (the assumption that $N > 0$ is used here). Moreover, for each coordinate j such that $\gamma_j = (\geq N)$, the machine $\mathcal{M}$ can use a zero test on $\mathcal{C}_j^{[q,\gamma]}$ to determine if $\tau_j = N$ or $\tau_j > N$. The remaining coordinates of τ are fixed by the knowledge of γ. This allows $\mathcal{M}$ to determine the unique club $[q', \gamma'] \in F$ such that $(q', \tau') \in [q', \gamma']$. Thus, $\mathcal{M}$ can copy the values from the counters $\mathcal{C}_1^{[q,\gamma]}, \ldots, \mathcal{C}_k^{[q,\gamma]}$ to the counters $\mathcal{C}_1^{[q',\gamma']}, \ldots, \mathcal{C}_k^{[q',\gamma']}$ and additionally perform the counter modifications as in the original transition: if $\tau'_j \neq \tau_j$ then the machine updates $\mathcal{C}_j^{[q',\gamma']}$ in such a way to ensure that $\mathcal{C}_j^{[q',\gamma']} = \max(\tau'_j - N, 0)$ — this update may also require to perform a zero test on $\mathcal{C}_j^{[q,\gamma]}$ to know if $\tau_j = N$ or not.

There are two technical subtleties of the above construction. First $[q', \gamma']$ might be equal to $[q, \gamma]$ and then no copying is needed. Second, it might be the case that $\gamma_j = \tau_j$ is a natural number greater than N in which case the value of $\mathcal{C}_j^{[q,\gamma]}$ is a positive number equal to $\tau_j - N$. In that case that number needs to be stored in $\mathcal{C}_j^{[q,\gamma]}$ (even though it is fixed by γ) and then copied to $\mathcal{C}_j^{[q',\gamma']}$ because it might be the case that $\gamma'_j = (\geq N)$.

We exploit here the fact that the thresholds of all the clubs in F are the same.

A.2. Simulating non-determinism

The above simulation allows us to represent one configuration of $\mathcal{A}$ for each club $[q, \gamma] \in F$. Moreover, we know how to perform transitions on these configurations. We can also simulate non-determinism of $\mathcal{A}$: if $a : (q, \tau) \mapsto_{\mathcal{A}} (q', \tau')$ and $a : (q, \tau) \mapsto_{\mathcal{A}} (q'', \tau'')$ such that (q', τ') and (q'', τ'') belong to distinct clubs of F then $\mathcal{M}$ can simulate both transitions simultaneously. This means that $\mathcal{M}$ can simulate in parallel all possible runs of $\mathcal{A}$ over the given input word α. The only situation when that fails is a *collision*, i.e. the situation when two[2] configurations (q', τ') and (q', τ'') can be reached via distinct finite runs over a joint prefix of α, with both (q', τ') and (q', τ'') belonging to the same club $[q', \gamma'] \in F$.

The crucial observation that makes the construction of $\mathcal{M}$ possible is Lemma 10.13 — whenever a collision occurs, it means that either $(q', \tau') \neq (q', \tau'')$ and the club $[q', \gamma']$ is trivial (i.e. $\mathrm{L}\big(\mathcal{A}, [q', \gamma']\big) = \emptyset$); or $(q', \tau') = (q', \tau'')$ and $\mathrm{L}\big(\mathcal{A}, (q', \tau')\big) = \emptyset$. Thus, the following remark holds.

Remark A.1. If a configuration (q', τ') is a part of a collision then $\mathrm{L}\big(\mathcal{A}, (q', \tau')\big) = \emptyset$.

Let us explain the construction more formally. Assume that the control state of $\mathcal{M}$ remembers for which clubs $[q, \gamma] \in F$ the counters $\mathcal{C}_1^{[q,\gamma]}, \ldots, \mathcal{C}_k^{[q,\gamma]}$ actually represent a configuration of $\mathcal{A}$ (otherwise their values are irrelevant). Thus, the set of states of $\mathcal{M}$ is the set of bitmaps 2^F that mark some clubs $[q, \gamma] \in F$ as *inhabited* and the other as *not inhabited*. The initial configuration of $\mathcal{M}$ stores 0 in all the counters and the bitmap marks only one club as inhabited — the one containing $(q_0, 0, \ldots, 0)$, i.e. the initial configuration of $\mathcal{A}$.

Reading a letter $a \in \Sigma$, the machine $\mathcal{M}$ simulates all the possible transitions $a : (q, \tau) \mapsto_{\mathcal{A}} (q', \tau')$ for all the configurations (q, τ) represented in the counters $\mathcal{C}_1^{[q,\gamma]}, \ldots, \mathcal{C}_k^{[q,\gamma]}$ for each inhabited club $[q, \gamma] \in F$. Consider the case when a collision occurs, and two configurations $(q', \tau'), (q', \tau'') \in [q', \gamma']$ belonging to the same club $[q', \gamma'] \in F$ need to be stored in the counters $\mathcal{C}_1^{[q',\gamma']}, \ldots, \mathcal{C}_k^{[q',\gamma']}$. In that case $\mathcal{M}$ *discards* both configurations (q', τ') and (q', τ'').

The control state of $\mathcal{M}$ is updated accordingly to know which clubs are inhabited: a club $[q', \gamma'] \in F$ is inhabited after such a transition iff at least one of the simulated transitions led to a configuration $(q', \tau') \in [q', \gamma']$ that was not discarded. This concludes the definition of the transition function of $\mathcal{M}$.

A.3. Acceptance condition and equivalence

We now need to define the acceptance condition of $\mathcal{M}$. Assume that an ω-word $\sigma = a_1 a_2 \ldots$ has been read. Consider a run r of $\mathcal{A}$ over σ. The consecutive transitions used in r are simulated by $\mathcal{M}$ when reading σ. There are two possibilities: either one of the configurations in r is discarded in $\mathcal{M}$ because of a collision; or all the configurations in r are simulated by $\mathcal{M}$ (i.e. none of them is discarded). In the latter case we say that r is *simulated*.

[2]We do not assume that the configurations (q', τ') and (q', τ'') are distinct, it might be the case that $\tau' = \tau''$ but there are two distinct runs reaching the configuration (q', τ').

Let the acceptance condition of $\mathcal{M}$ be chosen in such a way that a run of $\mathcal{M}$ is accepting if and only if it there exists at least one accepting run r of $\mathcal{A}$ that is simulated. Notice that Remark A.1 together with the policy of discarding configurations upon collisions imply the following fact.

Fact A.2. If r is an accepting run of $\mathcal{A}$ over an ω-word σ then it is simulated, i.e. none of its configurations is discarded by $\mathcal{M}$.

Lemma A.3. One can encode the above acceptance condition of $\mathcal{M}$ as a Muller condition, at the cost of extending the state space of $\mathcal{M}$.

Proof:
Consider a transition taken by $\mathcal{M}$ from one of its states upon reading a letter a. Our goal is to encode that transition as an element θ of $2^{F\times F}$. We will call θ the *graph* of that transition of $\mathcal{M}$. Let the graph θ contain a pair $\langle [q,\gamma],[q',\gamma']\rangle \in F\times F$ if and only if the club $[q,\gamma]$ was inhabited at the beginning of the transition by a configuration $(q,\tau)\in[q,\gamma]$, $\mathcal{M}$ simulated a transition $a:(q,\tau)\mapsto_{\mathcal{A}}(q',\tau')$, the configuration (q',τ') was not discarded, and $(q',\tau')\in[q',\gamma']$.

Each infinite execution of $\mathcal{M}$ while reading an ω-word σ defines a sequence of graphs $\theta_1,\theta_2,\ldots$ encoding the successive transitions of $\mathcal{M}$. The ω-word $\theta_1\theta_2\ldots\in\left(2^{F\times F}\right)^\omega$ is called the *graph* of σ. A *path* in such a graph is a sequence of clubs $\left([q_i,\gamma_i]\right)_{i=1,\ldots}$ such that for each $i\geq 1$ we have $\langle [q_i,\gamma_i],[q_{i+1},\gamma_{i+1}]\rangle\in\theta_i$. Such a path is *accepting* if infinitely many of the states q_i are accepting.

Notice that (for a fixed ω-word σ) there is a bijection between runs r of $\mathcal{A}$ over σ that are simulated and paths in the graph of σ. Moreover, accepting runs correspond to accepting paths. Therefore, to check if there is an accepting simulated run it is enough to check if the graph $\theta_1\theta_2\ldots\in\left(2^{F\times F}\right)^\omega$ contains an accepting path.

Treating $2^{F\times F}$ as a finite alphabet, the ω-language of ω-words $\theta_1\theta_2\ldots\in\left(2^{F\times F}\right)^\omega$ that contain an accepting path is regular and therefore it can be recognised by a deterministic Muller automaton $\mathcal{D}$. By extending each state of $\mathcal{M}$ by an additional state of $\mathcal{D}$, one can encode the acceptance condition of $\mathcal{D}$ directly on $\mathcal{M}$, turning it into a Muller machine. □

Notice that the above lemma relies heavily on the fact that the acceptance condition of $\mathcal{A}$ is encoded only on the set of states K of $\mathcal{A}$, without considering the counter values. This allows us to detect accepting runs of $\mathcal{A}$ by looking only on the sequence of clubs that were visited.

This concludes the construction of the machine $\mathcal{M}$. The following corollary of Fact A.2 implies that $\mathcal{M}$ satisfies the requirements from Theorem 10.1.

Corollary A.4. The machine $\mathcal{M}$ accepts an ω-word $\sigma\in\Sigma^\omega$ if and only if $\sigma\in\mathrm{L}(\mathcal{A})$.

A.4. Discussion

The above construction is quite flexible both in terms of the input and output models (i.e. the exact abilities of the automata $\mathcal{A}$ and $\mathcal{M}$ respectively). For instance, if one doesn't care about the effectiveness of the determinisation construction, one can extend it to allow the input automaton to *reset* the value of some of its counters to 0 (this can be simulated by $\mathcal{M}$ using copying). Clearly Remark 2.3 is

still valid for these machines, therefore Remark 10.10 applies one can repeat the rest of the argument proving the existence of the machine $\mathcal{M}$. Notice that one cannot hope for an effective variant of Remark 10.10 (i.e. Lemma 10.9) because the non-emptiness problem for Büchi blind counter automata with resets is undecidable (see Theorem 10 and Lemma 17 in [66]).

On the other hand, it seems that the ability of the output machine $\mathcal{M}$ to perform zero tests is inherent to that construction. At the same time, $\mathcal{A}$ needs to be blind because otherwise it would violate Remark 2.3. This implies that there might be no single type of counter machines so that both $\mathcal{A}$ and $\mathcal{M}$ could be assumed to be of that type (which would provide a determinisation construction within that type of automata). Also, there seems to be no reasonable way to avoid the need of copying the counters in $\mathcal{M}$, as the graph of the possible traces in $\mathcal{M}$ can be complex and branching.

The above construction is arranged in such a way to ensure that $\mathcal{M}$ is a real time multicounter automaton with zero tests and counter copying. This model of machines is much more concrete than general Turing machines, but in the end its expressive power is essentially the same — already two-counter Minsky machines with zero tests are able to simulate arbitrary Turing machines [65].