Math. Struct. in Comp. Science (2006), *vol.* 16, *pp.* 813–840. © 2006 Cambridge University Press
doi:10.1017/S0960129506005597 Printed in the United Kingdom

Borel ranks and Wadge degrees of context free ω-languages

OLIVIER FINKEL

Equipe de Logique Mathématique, U.F.R. de Mathématiques, Université Paris 7, 2 Place Jussieu 75251 Paris cedex 05, France
Email: `finkel@logique.jussieu.fr`

Received 13 September 2005; revised 31 January 2006

We show that the Borel hierarchy of the class of context free ω-languages, or even of the class of ω-languages accepted by Büchi 1-counter automata, is the same as the Borel hierarchy of the class of ω-languages accepted by Turing machines with a Büchi acceptance condition. In particular, for each recursive non-null ordinal α, there exist some $\mathbf{\Sigma}^0_\alpha$-complete and some $\mathbf{\Pi}^0_\alpha$-complete ω-languages accepted by Büchi 1-counter automata. And the supremum of the set of Borel ranks of context free ω-languages is an ordinal γ_2^1 that is strictly greater than the first non-recursive ordinal ω_1^{CK}. We then extend this result, proving that the Wadge hierarchy of context free ω-languages, or even of ω-languages accepted by Büchi 1-counter automata, is the same as the Wadge hierarchy of ω-languages accepted by Turing machines with a Büchi or a Muller acceptance condition.

1. Introduction

Languages of infinite words accepted by finite automata were first studied by Büchi to prove the decidability of the monadic second-order theory of one successor over the integers. The theory of the so-called regular ω-languages is now well established and has found many applications for specification and verification of non-terminating systems; see Thomas (1990), Staiger (1997) and Perrin and Pin (2004) for many results and references. More powerful machines, such as pushdown automata and Turing machines, have also been considered for the reading of infinite words, see the survey Staiger (1997) and the fundamental study Engelfriet and Hoogeboom (1993) on **X**-automata, that is, finite automata equipped with a storage type **X**. A way to study the complexity of ω-languages is to study their topological complexity, and, in particular, to locate them with regard to the Borel and projective hierarchies. On the one hand, all ω-languages accepted by *deterministic* **X**-automata with a Muller acceptance condition are Boolean combinations of $\mathbf{\Pi}^0_2$-sets, and hence $\mathbf{\Delta}^0_3$-sets (Staiger 1997; Engelfriet and Hoogeboom 1993). This implies, from Mc Naughton's Theorem, that all regular ω-languages, which are accepted by deterministic Muller automata, are also $\mathbf{\Delta}^0_3$-sets. On the other hand, for *non-deterministic* finite machines, it is natural to ask the question, posed by Lescow and Thomas in Lescow and Thomas (1994): What is the topological complexity of ω-languages accepted by automata equipped with a given storage type **X**? It is well known that every ω-language accepted by a Turing machine (and hence also by an **X**-automaton) with a Muller acceptance condition is an analytic set. In previous papers, we proved that there

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

are context free ω-languages, accepted by Büchi or Muller pushdown automata, of every finite Borel rank, of infinite Borel rank, or even analytic but non-Borel sets, (Duparc *et al.* 2001; Finkel 2001a; Finkel 2003a; Finkel 2003b). In this paper we show that the Borel hierarchy of ω-languages accepted by **X**-automata, for every storage type **X** such that 1-counter automata can be simulated by **X**-automata, is the same as the Borel hierarchy of ω-languages accepted by Turing machines with a Büchi acceptance condition. In particular, for each recursive non-null ordinal α, there exist some $\mathbf{\Sigma}^0_\alpha$-complete and some $\mathbf{\Pi}^0_\alpha$-complete ω-languages accepted by Büchi 1-counter automata, and hence also in the class $\mathbf{CFL}_\omega$ of context free ω-languages.

Here we should indicate a mistake in the conference paper Finkel (2005). We wrote in that paper that it is well known that if $L \subseteq \Sigma^\omega$ is a (lightface) Σ^1_1 set, that is, accepted by a Turing machine with a Büchi acceptance condition, and is a Borel set of rank α, then α is smaller than the Church Kleene ordinal ω_1^{CK}, which is the first non-recursive ordinal. This fact, which is true if we replace the (lightface) class Σ^1_1 by the (lightface) class Δ^1_1, is actually not true. Kechris, Marker and Sami proved in Kechris *et al.* (1989) that the supremum of the set of Borel ranks of (lightface) Π^1_1, and thus also of (lightface) Σ^1_1, sets is the ordinal γ^1_2. This ordinal is precisely defined in Kechris *et al.* (1989) and it is strictly greater than the ordinal ω_1^{CK}. The proofs we give in this paper show that the ordinal γ^1_2 is also the supremum of the set of Borel ranks of ω-languages accepted by Büchi 1-counter automata, or of context free ω-languages.

By considering the Wadge hierarchy, which is a great refinement of the Borel hierarchy (Wadge 1983; Duparc 2001), we show the following strengthening of the preceding result. The Wadge hierarchy of the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ of ω-languages accepted by real time 1-counter Büchi automata, and hence also of the class $\mathbf{CFL}_\omega$, is the Wadge hierarchy of the class of ω-languages accepted by Turing machines with a Büchi acceptance condition.

We think that the surprising result obtained in this paper is of interest for both logicians working on hierarchies arising in recursion theory or descriptive set theory, and also for computer scientists working on questions connected with non-terminating systems, such as the construction of effective strategies in infinite games (Walukiewicz 2000; Thomas 2002; Cachat 2002; Serre 2004b).

The paper is organised as follows. In Section 2 we define multicounter automata, which will be useful tools in the rest of the paper. The Borel hierarchy is described in Section 3. In Section 4 we study the Borel hierarchy of ω-languages accepted by real-time 8-counter automata, and in Section 5 The Borel hierarchy of the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$. Results for the Wadge hierarchy of the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ are given in Section 6.

2. Multicounter automata

We assume the reader is familiar with the theory of formal (ω)-languages (Thomas 1990; Staiger 1997). We shall use the usual notation of formal language theory.

When Σ is a finite alphabet, a *non-empty finite word* over Σ is any sequence $x = a_1 \ldots a_k$, where $a_i \in \Sigma$ for $i = 1, \ldots, k$, and k is an integer $\geqslant 1$. The *length* of x is k, denoted by $|x|$.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

The *empty word* has no letter and is denoted by λ; its length is 0. For $x = a_1 \dots a_k$, we write $x(i) = a_i$ and $x[i] = x(1) \dots x(i)$ for $i \leqslant k$ and $x[0] = \lambda$. $\Sigma^\star$ is the *set of finite words* (including the empty word) over Σ.

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 $i \geqslant 1$, $a_i \in \Sigma$. When σ is an ω-word over Σ, we write $\sigma = \sigma(1)\sigma(2)\dots\sigma(n)\dots$, where for all i, $\sigma(i) \in \Sigma$, and $\sigma[n] = \sigma(1)\sigma(2)\dots\sigma(n)$ for all $n \geqslant 1$ and $\sigma[0] = \lambda$.

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 $v = u.w$. The *set of* ω-*words* over the alphabet Σ is denoted by Σ^ω. An ω-*language* over an alphabet Σ is a subset of Σ^ω. The complement (in Σ^ω) of an ω-language $V \subseteq \Sigma^\omega$ is $\Sigma^\omega - V$, denoted V^-.

Definition 2.1. Let k be an integer $\geqslant 1$. A k-counter machine (k-CM) is a 4-tuple $\mathcal{M}=(K, \Sigma, \Delta, q_0)$, where K is a finite set of states, Σ is a finite input alphabet, $q_0 \in K$ is the initial state, and $\Delta \subseteq K \times (\Sigma \cup \{\lambda\}) \times \{0,1\}^k \times K \times \{0,1,-1\}^k$ is the transition relation. The k-counter machine $\mathcal{M}$ is said to be *real time* if and only if $\Delta \subseteq K \times \Sigma \times \{0,1\}^k \times K \times \{0,1,-1\}^k$, that is, if and only if there are no λ-transitions.

If the machine $\mathcal{M}$ is in state q and $c_i \in \mathbf{N}$ is the content of the i^{th} counter $\mathcal{C}_i$, then the configuration (or global state) of $\mathcal{M}$ is the $(k+1)$-tuple $(q, c_1, \dots, c_k)$.

For $a \in \Sigma \cup \{\lambda\}$, $q, q' \in K$ and $(c_1, \dots, c_k) \in \mathbf{N}^k$ 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 $j \notin E$, we write

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

$\mapsto^\star_{\mathcal{M}}$ is the transitive and reflexive closure of $\mapsto_{\mathcal{M}}$. (The subscript $\mathcal{M}$ will be omitted whenever the meaning remains clear).

Thus we see that the transition relation must 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).

Let $\sigma = a_1 a_2 \dots a_n$ be a finite word over Σ. A sequence of configurations

$$r = (q_i, c^i_1, \dots c^i_k)_{1 \leqslant i \leqslant p},$$

for $p \geqslant n+1$, is called a run of $\mathcal{M}$ on σ, starting in configuration $(p, c_1, \dots, c_k)$, if and only if:

(1) $(q_1, c^1_1, \dots c^1_k) = (p, c_1, \dots, c_k)$.

(2) For each $i \geqslant 1$, there exists $b_i \in \Sigma \cup \{\lambda\}$ such that

$$b_i : (q_i, c^i_1, \dots c^i_k) \mapsto_{\mathcal{M}} (q_{i+1}, c^{i+1}_1, \dots c^{i+1}_k).$$

(3) $a_1.a_2.a_3 \dots a_n = b_1.b_2.b_3 \dots b_p$.

Let $\sigma = a_1 a_2 \dots a_n \dots$ be an ω-word over Σ. An ω-sequence of configurations

$$r = (q_i, c^i_1, \dots c^i_k)_{i \geqslant 1}$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

is called a run of $\mathcal{M}$ on σ, starting in configuration $(p, c_1, \ldots, c_k)$, if and only if:

(1) $(q_1, c_1^1, \ldots c_k^1) = (p, c_1, \ldots, c_k)$.

(2) For each $i \geqslant 1$, there exists $b_i \in \Sigma \cup \{\lambda\}$ such that

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

such that either

$$a_1 a_2 \ldots a_n \ldots = b_1 b_2 \ldots b_n \ldots$$

or $b_1 b_2 \ldots b_n \ldots$ is a finite prefix of $a_1 a_2 \ldots a_n \ldots$

The run r is said to be complete when $a_1 a_2 \ldots a_n \ldots = b_1 b_2 \ldots b_n \ldots$

For every such run, In(r) is the set of all states entered infinitely often during run r.

A complete run r of M on σ, starting in configuration $(q_0, 0, \ldots, 0)$, will simply be called 'a run of M on σ'.

Definition 2.2. A Büchi k-counter automaton is a 5-tuple $\mathcal{M}=(K, \Sigma, \Delta, q_0, F)$, where $\mathcal{M}'=(K, \Sigma, \Delta, q_0)$ is a k-counter machine and $F \subseteq K$ is the set of accepting states. The ω-language accepted by $\mathcal{M}$ is

$$L(\mathcal{M})= \{\sigma \in \Sigma^\omega \mid \text{ there exists a run r of } \mathcal{M} \text{ on } \sigma \text{ such that In}(r) \cap F \neq \emptyset\}.$$

Definition 2.3. A Muller k-counter automaton is a 5-tuple $\mathcal{M}=(K, \Sigma, \Delta, q_0, \mathcal{F})$, where $\mathcal{M}'=(K, \Sigma, \Delta, q_0)$ is a k-counter machine and $\mathcal{F} \subseteq 2^K$ is the set of accepting sets of states. The ω-language accepted by $\mathcal{M}$ is

$$L(\mathcal{M})=\{\sigma \in \Sigma^\omega \mid \text{ there exists a run r of } \mathcal{M} \text{ on } \sigma \text{ such that } \exists F \in \mathcal{F} \;\; \text{In}(r) = F\}.$$

— The class of Büchi k-counter automata will be denoted **BC**(k).

— The class of real-time Büchi k-counter automata will be denoted **r-BC**(k).

— The class of ω-languages accepted by Büchi k-counter automata will be denoted **BCL**$(k)_\omega$.

— The class of ω-languages accepted by real-time Büchi k-counter automata will be denoted **r-BCL**$(k)_\omega$.

It is well known that an ω-language is accepted by a (real-time) Büchi k-counter automaton if and only if it is accepted by a (real-time) Muller k-counter automaton (Engelfriet and Hoogeboom 1993). Note that this cannot be shown without using the non-determinism of automata, and this result is no longer true in the deterministic case.

Note also that the 1-counter automata introduced above are equivalent to pushdown automata with a stack alphabet in the form $\{Z_0, A\}$ where Z_0 is the bottom symbol, which always remains at the bottom of the stack and only appears there, and A is another stack symbol. The pushdown stack may be seen as a counter whose content is the integer N if the stack content is the word $Z_0.A^N$.

In the model introduced here, the counter value cannot be increased by more than 1 during a single transition. However, this does not change the class of ω-languages accepted by such automata. So the class **BCL**$(1)_\omega$ is equal to the class **1-ICL**$_\omega$, introduced in Finkel (2001b), and it is a strict subclass of the class **CFL**$_\omega$ of context free ω-languages accepted by Büchi pushdown automata.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

3. Borel hierarchy

We assume the reader is familiar with basic notions of topology, which may also be found in Moschovakis (1980), Lescow and Thomas (1994), Kechris (1995), Staiger (1997) and Perrin and Pin (2004). There is a natural metric on the set Σ^ω of infinite words over a finite alphabet Σ, 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 for which *open subsets* of Σ^ω are in the form $W.\Sigma^\omega$, where $W \subseteq \Sigma^\star$. A set $L \subseteq \Sigma^\omega$ is a *closed set* if and only if its complement $\Sigma^\omega - L$ is an open set. We now define 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 $\alpha \geqslant 2$:

— $\mathbf{\Sigma}^0_\alpha$ 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 $\bigcup_{\gamma<\alpha} \mathbf{\Sigma}^0_\gamma$.

For a countable ordinal α, a subset of Σ^ω is a Borel set of *rank* α if and only if it is in $\mathbf{\Sigma}^0_\alpha \cup \mathbf{\Pi}^0_\alpha$ but not in $\bigcup_{\gamma<\alpha}(\mathbf{\Sigma}^0_\gamma \cup \mathbf{\Pi}^0_\gamma)$.

There are also some subsets of Σ^ω that 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* that are obtained by projection of Borel sets, see, for example, Staiger (1997), Lescow and Thomas (1994), Perrin and Pin (2004) and Kechris (1995) for more details.

We now define completeness with regard to reduction by continuous functions. For a countable ordinal $\alpha \geqslant 1$, 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* if and only if for any set $E \subseteq Y^\omega$ (with Y a finite alphabet): $E \in \mathbf{\Sigma}^0_\alpha$ (respectively, $E \in \mathbf{\Pi}^0_\alpha$, $E \in \mathbf{\Sigma}^1_1$) if and only if there exists a continuous function $f : Y^\omega \to \Sigma^\omega$ such that $E = f^{-1}(F)$. $\mathbf{\Sigma}^0_n$- and $\mathbf{\Pi}^0_n$-complete sets, with n an integer $\geqslant 1$, are thoroughly characterised in Staiger (1986).

The (lightface) class Σ^1_1 of *effective analytic sets* is the class of sets that are obtained by projection of arithmetical sets. It is well known that a set $L \subseteq \Sigma^\omega$, where Σ is a finite alphabet, is in the class Σ^1_1 if and only if it is accepted by a Turing machine with a Büchi or Muller acceptance condition (Staiger 1997).

As indicated in the introduction, the conference paper Finkel (2005) contained a mistake. We wrote there that it is well known that if $L \subseteq \Sigma^\omega$ is a (lightface) Σ^1_1 set, and is a Borel set of rank α, then α is smaller than ω_1^{CK}. This fact, which is true if we replace the (lightface) class Σ^1_1 by the (lightface) class Δ^1_1, is actually not true. Kechris, Marker and Sami proved in Kechris *et al.* (1989) that the supremum of the set of Borel ranks of (lightface) Π^1_1, and thus also of (lightface) Σ^1_1, sets is the ordinal γ^1_2.

This ordinal is precisely defined in Kechris *et al.* (1989). Kechris, Marker and Sami proved that the ordinal γ^1_2 is strictly greater than the ordinal δ^1_2, which is the first non-Δ^1_2 ordinal. Thus, in particular, $\omega_1^{\mathrm{CK}} < \gamma^1_2$. The exact value of the ordinal γ^1_2 may depend on

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

axioms of set theory (Kechris *et al.* 1989). It is consistent with the axiomatic system **ZFC** that γ_2^1 is equal to the ordinal δ_3^1, which is the first non-Δ_3^1 ordinal (because $\gamma_2^1 = \delta_3^1$ in **ZFC + (V=L)**). On the other hand, the axiom of Π_1^1-determinacy implies that $\gamma_2^1 < \delta_3^1$. For more details, see Kechris *et al.* (1989) and a textbook of set theory such as Jech (2002).

Note, however, that it appears still not to be known whether *every* non-null ordinal $\gamma < \gamma_2^1$ is the Borel rank of a (lightface) Π_1^1 (or Σ_1^1) set. On the other hand, it is known that every ordinal $\gamma < \omega_1^{\mathrm{CK}}$ is the Borel rank of a (lightface) Δ_1^1 set. Moreover, for every non-null ordinal $\alpha < \omega_1^{\mathrm{CK}}$, there exist some $\mathbf{\Sigma}_\alpha^0$-complete and some $\mathbf{\Pi}_\alpha^0$-complete sets in the class Δ_1^1. Louveau gives the following argument: the natural universal set for the class $\mathbf{\Sigma}_\alpha^0$ (respectively, $\mathbf{\Pi}_\alpha^0$), where $\alpha < \omega_1^{\mathrm{CK}}$, is a $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete) set and it is in the class Δ_1^1 (Louveau 2005). The definition and the construction of a universal set for a given Borel class may be found in Moschovakis (1980).

4. Borel hierarchy of ω-languages in r-BCL$(8)_\omega$

It is well known that every Turing machine can be simulated by a (non-real-time) 2-counter automaton, see Hopcroft and Ullman (1979). Thus the Borel hierarchy of the class **BCL**$(2)_\omega$ is also the Borel hierarchy of the class of ω-languages accepted by Büchi Turing machines. We shall prove the following proposition.

Proposition 4.1. The Borel hierarchy of the class **r-BCL**$(8)_\omega$ is equal to the Borel hierarchy of the class **BCL**$(2)_\omega$.

We first sketch the proof of this result. We are going to find, from an ω-language $L \subseteq \Sigma^\omega$ in **BCL**$(2)_\omega$, another ω-language $\theta_S(L)$, which will be of the same Borel complexity but accepted by a *real-time* 8-counter Büchi automaton. The idea is first to add a storage type called a queue to a 2-counter Büchi automaton in order to read ω-words in real time. Then we shall see that a queue can be simulated by two pushdown stacks or by four counters. This simulation is not done in real time, but a crucial fact is that we can bound the number of transitions needed to simulate the queue. This allows us to pad the strings in L with enough extra letters to ensure that the new words will be read in real time by an 8-counter Büchi automaton (two counters are used to check that an ω-word really is obtained with the good padding, which is made in a regular way). The padding is obtained via the function θ_S, which we now define.

Let Σ be an alphabet having at least two letters, E be a new letter not in Σ, S be an integer $\geqslant 1$, and $\theta_S : \Sigma^\omega \to (\Sigma \cup \{E\})^\omega$ be the function defined, for all $x \in \Sigma^\omega$, by

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\ldots$$

We now state two lemmas.

Lemma 4.2. Let Σ be an alphabet having at least two letters and let $L \subseteq \Sigma^\omega$ be a subset of Σ^ω that is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α) for some ordinal $\alpha \geqslant 2$. Then the ω-language $\theta_S(L)$ is a subset of $(\Sigma \cup \{E\})^\omega$ that is $\mathbf{\Sigma}_\alpha^0$-complete (respectively, $\mathbf{\Pi}_\alpha^0$-complete, $\mathbf{\Sigma}_\alpha^0$ of rank α, $\mathbf{\Pi}_\alpha^0$ of rank α).

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Proof. Let Σ be an alphabet having at least two letters. It is easy to see that the function θ_S is continuous because if two ω-words x and y of Σ^ω have a common initial segment of length n, then the two ω-words $\theta_S(x)$ and $\theta_S(y)$ have a common initial segment of length greater than n.

Let $\varphi : (\Sigma \cup \{E\})^\omega \to (\Sigma \cup \{E\})^\omega$ be the mapping defined, for all $y \in (\Sigma \cup \{E\})^\omega$, by

$$\varphi(y) = y(1).y(S+2).y(S+S^2+3)\dots y(S+S^2+\dots+S^n+(n+1))\dots .$$

It is easy to see that the function φ is also continuous and that, for any $L \subseteq \Sigma^\omega$, we have $\theta_S(L) = \varphi^{-1}(L) \cap \theta_S(\Sigma^\omega)$.

Now let $L \subseteq \Sigma^\omega$ be a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of Σ^ω, and hence also of $(\Sigma \cup \{E\})^\omega$, for some ordinal $\alpha \geqslant 2$. Then $\varphi^{-1}(L)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$ because the class $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) is closed under inverse images by continuous functions. On the other hand, $\theta_S(\Sigma^\omega)$ is a closed set, so $\theta_S(L) = \varphi^{-1}(L) \cap \theta_S(\Sigma^\omega)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$ because the class $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) is closed under finite intersection.

Moreover, $L = \theta_S^{-1}(\theta_S(L))$. Thus, if L is assumed to be $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete), then $\theta_S(L)$ is a $\mathbf{\Sigma}^0_\alpha$-**complete** (respectively, $\mathbf{\Pi}^0_\alpha$-**complete**) subset of $(\Sigma \cup \{E\})^\omega$ because it is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) set, the function θ_S is continuous, and L is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete).

Now assume that L is a $\mathbf{\Sigma}^0_\alpha$-set of rank α (respectively, $\mathbf{\Pi}^0_\alpha$-set of rank α). We have already seen that $\theta_S(L)$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{E\})^\omega$. The Borel rank of $\theta_S(L)$ cannot be smaller than α, since otherwise $L = \theta_S^{-1}(\theta_S(L))$ would also be of Borel rank smaller than α. □

Lemma 4.3. Let Σ be an alphabet having at least two letters, and $L \subseteq \Sigma^\omega$ be an ω-language in the class $\mathbf{BCL}(2)_\omega$. Then there exists an integer $S \geqslant 1$ such that $\theta_S(L)$ is in the class $\mathbf{r\text{-}BCL}(8)_\omega$.

Proof. Let Σ be an alphabet having at least two letters and let $L \subseteq \Sigma^\omega$ be an ω-language accepted by a Büchi 2-counter automaton $\mathcal{A}$.

A way to construct a finite machine accepting the same ω-language but in *real time* would be to add a storage type called a *queue* (Engelfriet and Hoogeboom 1993).

Configurations of a queue are finite words over a finite alphabet Σ; a letter of Σ may be added to the rear of the queue or removed from the front; moreover, there are tests to determine the first letter of the queue.

The new machine will read words in real time. At every transition, a letter of the input ω-word is read, is added to the rear of the queue, and waits to be read (and then removed from the front of the queue) for the simulation of the reading of the input ω-word by the 2-counter automaton $\mathcal{A}$.

We are going to see that one can simulate a queue with four counters. This simulation will not be a *real-time* simulation, but we shall be able to get an upper bound on the number of transitions of the four counters that are necessary for the simulation of one transition of the queue. This upper bound will be useful later in the paper.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Claim 4.4. A queue can be simulated by two pushdown stacks.

Proof. Assume that the queue alphabet is $\Sigma = \{Z_2, Z_3, \ldots, Z_{k-1}\}$, for some integer $k \geqslant 3$.

The content of the queue can be represented by a finite word $Z_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$, where the letter Z_{i_m} is the first letter of the queue and Z_{i_1} is the last letter of the queue (the last added to the rear).

This content can be stored in a pushdown stack whose alphabet is $\Gamma = \Sigma \cup \{Z_1\}$, where Z_1 is the bottom symbol, which appears only at the bottom of the stack and always remains there. The stack content representing the queue content $Z_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$ will be simply $Z_1Z_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.

If the letter Z_{i_m} of the front of the queue is removed from the queue, it suffices to pop the same letter from the top of the stack.

To simulate the addition of a new letter Z_r to the rear of the queue we can use a second pushdown stack whose alphabet is also Γ.

In fact, we have to add the letter Z_r between the letters Z_1 and Z_{i_1} of the first pushdown stack. To achieve this, we successively pop letters from the top of the first stack, pushing them onto the second stack, which initially contains only the bottom symbol Z_1. After having done this operation for letters $Z_{i_m}, Z_{i_{m-1}}, \ldots Z_{i_2}, Z_{i_1}$, the content of the first stack is Z_1 and the content of the second stack is $Z_1Z_{i_m}Z_{i_{m-1}}\ldots Z_{i_2}Z_{i_1}$. We can then push the letter Z_r onto the top of the first stack. Then we successively pop letters $Z_{i_1}, Z_{i_2}, Z_{i_3}\ldots, Z_{i_m}$ from the top of the second stack, pushing them back onto the first stack. At the end of this operation the content of the first stack is $Z_0Z_rZ_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$ and it represents the new content of the queue. □

We now recall the following well-known property.

Claim 4.5 (Hopcroft and Ullman 1979). A pushdown stack can be simulated by two counters.

Proof. Consider a stack having $k-1$ symbols $Z_1, Z_2, \ldots, Z_{k-1}$. The stack content $Z_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$ can be represented by the integer j, which is given in base k by

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \ldots + k^{m-1}.i_1\,.$$

Note that, as mentioned in Hopcroft and Ullman (1979), not every integer represents a stack content. In particular, an integer whose representation in base k contains the digit 0 does not represent any stack content.

We are going to see how to use a second counter to determine which letter is at the top of the stack, and to simulate the operations of pushing a letter onto the stack or of popping a letter from the top of the stack.

Assume that the integer j representing the stack content $Z_{i_1}Z_{i_2}Z_{i_3}\ldots Z_{i_m}$ is stored in one of the two counters.

In order to determine which letter is at the top of the stack, we can copy the content j to the second counter, using the finite control of the finite machine to compute j modulo k.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

The integer j modulo k is equal to the integer i_m, which characterises the letter Z_{i_m}, and hence the letter at the top of the stack.

It would be possible to transfer the integer j back to the first counter again, but one can also leave it in the second counter and use the finite control to know in which counter the integer j is stored.

Note that this operation needs only j steps (and $2j$ steps if we transfer j back to the first counter).

If a letter Z_r is pushed onto the stack, the content of the stack becomes

$$Z_{i_1}Z_{i_2}Z_{i_3}\dots Z_{i_m}Z_r$$

and the integer associated with that content is $j.k + r$.

It is easy to store the integer $j.k$ in the second counter, by adding k to this counter each time the first counter is decreased by 1. When the content of the first counter is equal to zero, the content of the second counter is equal to $j.k$. One can then add r to the second counter by using the finite control of the machine. Again we can use the finite control to know that the content of the stack is now coded by the integer that is in the *second* counter.

Note that the whole operation needs only $j.k + r$ steps.

If, instead, the symbol Z_{i_m} is popped from the top of the stack, the new content of the stack is $Z_{i_1}Z_{i_2}Z_{i_3}\dots Z_{i_{m-1}}$, and it is represented by the integer $[\frac{j}{k}] = i_{m-1} + k.i_{m-2} + \dots + k^{m-2}.i_1$, which is the integer part of $\frac{j}{k}$.

To get the integer $[\frac{j}{k}]$ as the content of the second counter, we can decrease the first counter from j to zero, adding 1 to the second counter each time the first one is decreased by k.

Note that this operation needs only j steps.

Note also that we can achieve this operation in a non-deterministic way, checking at the end which letter was at the top of the stack. □

We have seen above that a queue can be simulated using two pushdown stacks, and hence also using four counters.

This simulation is not done in *real time*, but we shall see that we can get an upper bound on the number of transitions of the four counters simulating one transition of the queue. This upper bound will be crucial in view of Lemma 4.3.

Claim 4.6. Assume as above that the queue alphabet is $\Sigma = \{Z_2, Z_3, \dots, Z_{k-1}\}$ for some integer $k \geqslant 3$ and that at some time the content of the queue is a finite word $Z_{i_1}Z_{i_2}Z_{i_3}\dots Z_{i_m}$. Then the number of transitions of four counters that are needed to simulate the addition of a letter Z_r to the rear of the queue is smaller than $(2.k)^{m+2}$.

Proof. Recall that the content of the queue can be stored in a pushdown stack whose alphabet is $\Gamma = \Sigma \cup \{Z_1\}$, where Z_1 is the bottom symbol. The stack content representing the queue content $Z_{i_1}Z_{i_2}Z_{i_3}\dots Z_{i_m}$ is simply $Z_1Z_{i_1}Z_{i_2}Z_{i_3}\dots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

This stack content $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \dots Z_{i_m}$ can itself be represented by the integer j, which is given in base k by

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \dots + k^{m-1}.i_1 + k^m.1 \ \leqslant \ k^{m+1} .$$

We have seen that, considering the simulation of the addition of Z_r to the rear of a queue with two pushdown stacks, the first stack containing $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \dots Z_{i_m}$, we have first to successively pop letters $Z_{i_m}, \dots, Z_{i_3}, Z_{i_2}, Z_{i_1}$ from the top of the first stack and push them onto the second stack.

We have also seen above that popping a letter from the stack whose content is represented by the integer j needs only j transitions of two counters and that we can know at the end of this popping simulation which letter has just been popped.

Moreover, to push a letter Z_s onto the stack whose content is represented by an integer j' needs only $j'.k + s$ transitions of two counters.

Thus, at most $2.m.k^{m+1}$ transitions of four counters are necessary to simulate the operation of successively popping letters $Z_{i_m}, \dots, Z_{i_3}, Z_{i_2}, Z_{i_1}$ from the top of the first stack and then pushing them onto the second stack.

Two transitions of the counters are needed to check that the content of the first stack is now reduced to Z_1, which is simply represented by the integer 1, without changing this content; in one step the counter is reduced from one to zero then in a second step the counter is increased from zero to one.

To simulate the addition of the letter Z_r to the rear of the queue, we now push the letter Z_r onto the first stack; this is simulated by $k + r$ transitions of two counters.

Now we have to successively pop letters $Z_{i_1}, Z_{i_2}, Z_{i_3}, \dots, Z_{i_m}$ from the top of the second stack and push them again onto the first stack. This whole operation needs fewer than $m.k^{m+1} + m.k^{m+2}$ transitions of the four counters.

Finally, to simulate the addition of a letter Z_r to the rear of the queue, we need only

$$2.m.k^{m+1} + 2 + k + r + m.k^{m+1} + m.k^{m+2}$$

transitions of four counters. This number is smaller than

$$4.m.k^{m+2} + 3.k \leqslant (2.k)^{m+2}.$$ □

Claim 4.7. Assume as above that the queue alphabet is $\Sigma = \{Z_2, Z_3, \dots, Z_{k-1}\}$ for some integer $k \geqslant 3$ and that at some time the content of the queue is a finite word $Z_{i_1} Z_{i_2} Z_{i_3} \dots Z_{i_m}$. Then the number of transitions of four counters that are needed to determine the letter Z_{i_m} that is at the front of the queue is smaller than k^{m+1}. And the number of transitions of four counters that are needed to simulate the operation of removing the letter Z_{i_m} from the front of the queue is smaller than k^{m+1}.

Proof. The content $Z_{i_1} Z_{i_2} Z_{i_3} \dots Z_{i_m}$ of the queue is represented by a stack content $Z_1 Z_{i_1} Z_{i_2} Z_{i_3} \dots Z_{i_m}$, where Z_1 is at the bottom of the stack and Z_{i_m} is at the top of the stack.

This stack content is itself represented by the integer

$$j = i_m + k.i_{m-1} + k^2.i_{m-2} + \dots + k^{m-1}.i_1 + k^m.1 \ \leqslant \ k^{m+1} .$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

By the proof of Claim 4.5, $j \leqslant k^{m+1}$ transitions of four counters (and even of just two counters) suffice to determine the letter Z_{i_m} that is at the top of the stack or to pop it from the top of the stack. □

Claim 4.8. The ω-language $\theta_S(\Sigma^\omega)$ is in the class **r-BCL**$(2)_\omega$.

Proof. Recall that if Σ is an alphabet having at least two letters, E is a new letter not in Σ and S is an integer $\geqslant 1$, then an ω-word $y \in (\Sigma \cup \{E\})^\omega$ is in $\theta_S(\Sigma^\omega)$ if and only if it is in the form

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\dots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\dots$$

for some $x \in \Sigma^\omega$.

It is easy to construct a real-time Büchi 2-counter automaton $\mathscr{B}$ accepting $\theta_S(\Sigma^\omega)$. We now describe the behaviour of $\mathscr{B}$ when reading an ω-word $y \in (\Sigma \cup \{E\})^\omega$. After the reading of the first letter $y(1) \in \Sigma$, the automaton $\mathscr{B}$ adds one to the first counter for each letter E read, checking with the finite control that there are S letters E following $y(1)$. Then $\mathscr{B}$ reads a second letter of Σ, adds S to the second counter and decreases the first counter by one each time it reads S letters E. When the first counter content is equal to zero, the second counter content is equal to S^2 and $\mathscr{B}$ has read S^2 letters E. It then reads a third letter of Σ, adds S to the first counter and decreases the second counter by one each time it reads S letters E. When the second counter content is equal to zero, the first counter content is equal to S^3, and $\mathscr{B}$ has read S^3 letters E. Then $\mathscr{B}$ reads a fourth letter of Σ, and so on. The Büchi acceptance condition is used to check that the content of the first (and also the second) counter takes the value zero infinitely many times. □

Proof of Lemma 4.3 continued. Let $L \subseteq \Sigma^\omega$ be an ω-language accepted by a Büchi 2-counter automaton $\mathscr{A}$. We are going to explain the behaviour of a real-time Büchi 8-counter automaton $\mathscr{A}_1$ accepting $\theta_S(L)$, where $S = (3k)^3$ and $k = cardinal(\Sigma) + 2$.

As explained in the proof of Claim 4.8, two counters of $\mathscr{A}_1$ will be used, independently of the other six, to check that the input ω-word $y \in (\Sigma \cup \{E\})^\omega$ is in $\theta_S(\Sigma^\omega)$.

Now consider the reading by $\mathscr{A}_1$ of an ω-word $y \in (\Sigma \cup \{E\})^\omega$ in the form

$$y = \theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\dots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\dots$$

for some $x \in \Sigma^\omega$.

The automaton $\mathscr{A}_1$ will use four counters to simulate a queue in which letters

$$x(1), x(2), \dots, x(n), \dots$$

will be successively stored as soon as they are read. Two other counters of $\mathscr{A}_1$ will be used to simulate the reading of the ω-word x by the Büchi 2-counter automaton $\mathscr{A}$. Note that only letters $x(1), x(2), \dots, x(n), \dots$ will be added to the rear of the queue. Therefore, after reading the initial segment $x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\dots x(n).E^{S^n}$ of y, the content of the queue has cardinal smaller than or equal to n.

When $\mathscr{A}_1$ reads $x(n+1)$ it will first simulate the addition of the letter $x(n+1)$ to the rear of the queue, using four counters, and do this in real time while continuing to read some following letters E. By Claim 4.6, the number of transitions of four counters needed

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

to simulate the addition of $x(n+1)$ to the rear of the queue is smaller than $(2k)^{n+2}$, where $k = cardinal(\Sigma) + 2$. Next, $\mathscr{A}_1$ determines which letter is at the front of the queue. By Claim 4.7, this needs at most k^{n+2} transitions of the four counters (because there are now at most $(n+1)$ letters in the queue). Now the automaton $\mathscr{A}_1$ simulates, using two counters, only one transition of $\mathscr{A}$. This transition may be a λ-transition or not. In the second case, $\mathscr{A}_1$ simulates the reading by $\mathscr{A}$ of the letter at the front of the queue, so this letter is removed from the queue; by Claim 4.7, this again needs at most k^{n+2} transitions of the four counters. Now,

$$(2k)^{n+2} + k^{n+2} + k^{n+2} + 1 \leqslant ((3k)^3)^n .$$

Thus, if we set $S = (3k)^3$, all these transitions of the six counters can be achieved by the automaton $\mathscr{A}_1$ in *real time* during the reading of the letters E following $x(n+1)$ in y. Not all the S^{n+1} letters E are read during these transitions of the six counters. But $\mathscr{A}_1$ will read the others without changing the contents of the six counters, waiting for the reading of the next letter of Σ: the letter $x(n+2)$. It will then simulate the addition of this letter to the rear of the queue, and so on.

A Muller condition can be used to ensure that $y \in \theta_S(\Sigma^\omega)$, that is, $y = \theta_S(x)$ for some $x \in \Sigma^\omega$, *and* that $x \in L = L(\mathscr{A})$. As mentioned in Section 2, this can also be achieved with a Büchi acceptance condition. □

Note that in order to prove Lemma 4.3, we have not attempted to find the smallest possible integer S, but just one such integer.

5. Borel hierarchy of ω-languages in r-BCL$(1)_\omega$

We shall first prove the following result.

Proposition 5.1. Let $k \geqslant 2$ be an integer. If for some ordinal $\alpha \geqslant 2$ there is an ω-language in the class **r-BCL**$(k)_\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α), then there is some ω-language in the class **r-BCL**$(1)_\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

To simplify the exposition of the proof of this result, we are first going to give the proof for $k = 2$. Then we shall explain the modifications required to infer the result for the integer $k = 8$, which is in fact the only case we shall need later in the paper. (However, our main result will show that the proposition is true for every integer $k \geqslant 2$).

To that end, 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

$$h(x) = A.0^6.x(1).B.0^{6^2}.A.0^{6^2}.x(2).B.0^{6^3}.A.0^{6^3}.x(3).B \ldots B.0^{6^n}.A.0^{6^n}.x(n).B \ldots$$

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

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Lemma 5.2. Let Σ be a finite alphabet and $(h(\Sigma^\omega))^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$. If $\mathscr{L} \subseteq \Sigma^\omega$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α), for a countable ordinal $\alpha \geqslant 2$, then $h(\mathscr{L}) \cup h(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

Proof. The topological space Σ^ω is compact, so its image by the continuous function h is also a compact subset of the topological space $(\Sigma \cup \{A, B, 0\})^\omega$. The set $h(\Sigma^\omega)$ is compact, so it is a closed subset of $(\Sigma \cup \{A, B, 0\})^\omega$. Thus, its complement

$$(h(\Sigma^\omega))^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$$

is an open (that is, a $\mathbf{\Sigma}^0_1$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$.

On the other hand, the function h is also injective, and thus it is a bijection from Σ^ω onto $h(\Sigma^\omega)$. But a continuous bijection between two compact sets is a homeomorphism, so h induces a homeomorphism between Σ^ω and $h(\Sigma^\omega)$. Assume that $\mathscr{L}$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of Σ^ω. Then $h(\mathscr{L})$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $h(\Sigma^\omega)$ (where Borel sets of the topological space $h(\Sigma^\omega)$ are defined from open sets as in the case of the topological space Σ^ω).

The topological space $h(\Sigma^\omega)$ is a topological subspace of $(\Sigma \cup \{A, B, 0\})^\omega$ and its topology is induced by the topology on $(\Sigma \cup \{A, B, 0\})^\omega$: open sets of $h(\Sigma^\omega)$ are traces on $h(\Sigma^\omega)$ of open sets of $(\Sigma \cup \{A, B, 0\})^\omega$, and the same result holds for closed sets. Then one can easily show by induction that for every ordinal $\alpha \geqslant 1$, $\mathbf{\Pi}^0_\alpha$-subsets (respectively, $\mathbf{\Sigma}^0_\alpha$-subsets) of $h(\Sigma^\omega)$ are traces on $h(\Sigma^\omega)$ of $\mathbf{\Pi}^0_\alpha$-subsets (respectively, $\mathbf{\Sigma}^0_\alpha$-subsets) of $(\Sigma \cup \{A, B, 0\})^\omega$, that is, are intersections with $h(\Sigma^\omega)$ of $\mathbf{\Pi}^0_\alpha$-subsets (respectively, $\mathbf{\Sigma}^0_\alpha$-subsets) of $(\Sigma \cup \{A, B, 0\})^\omega$.

But $h(\mathscr{L})$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$)-subset of $h(\Sigma^\omega)$, for some ordinal $\alpha \geqslant 2$, so there exists a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset T of $(\Sigma \cup \{A, B, 0\})^\omega$ such that $h(\mathscr{L}) = T \cap h(\Sigma^\omega)$. But $h(\Sigma^\omega)$ is a closed, that is, $\mathbf{\Pi}^0_1$-subset of $(\Sigma \cup \{A, B, 0\})^\omega$ and the class of $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subsets of $(\Sigma \cup \{A, B, 0\})^\omega$ is closed under finite intersection, and thus $h(\mathscr{L})$ is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$.

Now $h(\mathscr{L}) \cup (h(\Sigma^\omega))^-$ is the union of a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset and a $\mathbf{\Sigma}^0_1$-subset of $(\Sigma \cup \{A, B, 0\})^\omega$, so it is a $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subset of $(\Sigma \cup \{A, B, 0\})^\omega$ because the class of $\mathbf{\Sigma}^0_\alpha$ (respectively, $\mathbf{\Pi}^0_\alpha$) subsets of $(\Sigma \cup \{A, B, 0\})^\omega$ is closed under finite union.

Now we assume first that $\mathscr{L}$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete). In order to prove that $h(\mathscr{L}) \cup (h(\Sigma^\omega))^-$ is $\mathbf{\Sigma}^0_\alpha$**-complete** (respectively, $\mathbf{\Pi}^0_\alpha$**-complete**), it is enough to note that

$$\mathscr{L} = h^{-1}[h(\mathscr{L}) \cup (h(\Sigma^\omega))^-] .$$

This implies that $h(\mathscr{L}) \cup (h(\Sigma^\omega))^-$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete) because $\mathscr{L}$ is assumed to be $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete).

On the other hand, if we assume only that $\mathscr{L}$ is a $\mathbf{\Sigma}^0_\alpha$-set of rank α (respectively, $\mathbf{\Pi}^0_\alpha$-set of rank α), we can infer that $h(\mathscr{L}) \cup (h(\Sigma^\omega))^-$ is also a $\mathbf{\Sigma}^0_\alpha$-set of rank α (respectively, $\mathbf{\Pi}^0_\alpha$-set of rank α). Indeed, if $h(\mathscr{L}) \cup (h(\Sigma^\omega))^-$ were of Borel rank $\beta < \alpha$, then $\mathscr{L} = h^{-1}[h(\mathscr{L}) \cup (h(\Sigma^\omega))^-]$ would also be of rank smaller than α because the class $\mathbf{\Sigma}^0_\beta$ (respectively, $\mathbf{\Pi}^0_\beta$) is closed under inverse images by continuous functions. □

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

In order to apply Lemma 5.2, we want to prove that if $L(\mathcal{A})\subseteq \Sigma^\omega$ is accepted by a real-time 2-counter automaton $\mathcal{A}$ with a Büchi acceptance condition, then $h(L(\mathcal{A}))\cup h(\Sigma^\omega)^-$ is accepted by a 1-counter automaton with a Büchi acceptance condition. We first prove the following lemma.

Lemma 5.3. Let Σ be a finite alphabet and h be the coding of ω-words over Σ defined as above. Then $h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$ is accepted by a real-time 1-counter Büchi automaton.

Proof. We can easily see that $h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega)$ is the set of ω-words in $(\Sigma \cup \{A, B, 0\})^\omega$ that belong to one of the following ω-languages:

— $\mathcal{D}_1$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ that have no initial segments in $A.0^6.\Sigma.B$. It is easy to see that $\mathcal{D}_1$ is in fact a regular ω-language.

— $\mathcal{D}_2$ is the complement of $(A.0^+.\Sigma.B.0^+)^\omega$ in $(\Sigma \cup \{A, B, 0\})^\omega$. However, the ω-language $(A.0^+.\Sigma.B.0^+)^\omega$ is regular, so its complement $\mathcal{D}_2$ is also a regular ω-language.

— $\mathcal{D}_3$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ that contain a segment in $B.0^n.A.0^m.\Sigma$ for some positive integers $n \neq m$. It is easy to see that this ω-language can be accepted by a real-time 1-counter Büchi automaton.

— $\mathcal{D}_4$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ that contain a segment in $A.0^n.\Sigma.B.0^m.A$ for some positive integers n and m with $m \neq 6n$. Again this ω-language can be accepted by a real-time 1-counter Büchi automaton.

The class **r-BCL**$(1)_\omega$ is closed under finite union because it is the class of ω-languages accepted by *non-deterministic* real-time 1-counter Büchi automata. On the other hand,

$$h(\Sigma^\omega)^- = (\Sigma \cup \{A, B, 0\})^\omega - h(\Sigma^\omega) = \cup_{1\leqslant i\leqslant 4}\mathcal{D}_i\,,$$

so $h(\Sigma^\omega)^-$ is accepted by a real-time 1-counter Büchi automaton. □

We would now like to prove that if $L(\mathcal{A})\subseteq \Sigma^\omega$ is accepted by a real-time 2-counter automaton $\mathcal{A}$ with a Büchi acceptance condition, then $h(L(\mathcal{A}))$ is in **BCL**$(1)_\omega$. We cannot show this, so we are first going to define another ω-language $\mathcal{L}(\mathcal{A})$ accepted by a 1-counter Büchi automaton and prove that $h(L(\mathcal{A}))\cup h(\Sigma^\omega)^- = \mathcal{L}(\mathcal{A})\cup h(\Sigma^\omega)^-$.

We shall need the following notion. Let $N \geqslant 1$ be an integer such that $N = 2^x.3^y.N_1$, where x, y are positive integers and $N_1 \geqslant 1$ is an integer that is not divisible by either 2 or 3. Then we set $P_2(N) = x$ and $P_3(N) = y$. So $2^{P_2(N)}$ is the greatest power of 2 that divides N and $2^{P_3(N)}$ is the greatest power of 3 that divides N.

Now we assume a 2-counter Büchi automaton $\mathcal{A}= (K, \Sigma, \Delta, q_0, F)$ accepting the ω-language $L(\mathcal{A})\subseteq \Sigma^\omega$. The ω-language $\mathcal{L}(\mathcal{A})$ is the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ in the form

$$A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A \ldots A.u_n.v_n.x_n.B.w_n.z_n.A \ldots$$

where, for all integers $i \geqslant 1$, $v_i, w_i \in 0^+$, $u_i, z_i \in 0^\star$, $x_i \in \Sigma$, $|u_1| = 5$, $|u_{i+1}| = |z_i|$ and there is a sequence $(q_i)_{i\geqslant 0}$ of states of K and integers $j_i, j_i' \in \{-1; 0; 1\}$, for $i \geqslant 1$, such that for all integers $i \geqslant 1$

$$x_i : (q_{i-1}, P_2(|v_i|), P_3(|v_i|)) \mapsto_{\mathcal{A}} (q_i, P_2(|v_i|) + j_i, P_3(|v_i|) + j_i')$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

and

$$|w_i| = |v_i|.2^{j_i}.3^{j'_i}.$$

Moreover, some state $q_f \in F$ occurs infinitely often in the sequence $(q_i)_{i\geqslant 0}$.

Note that the state q_0 of the sequence $(q_i)_{i\geqslant 0}$ is also the initial state of $\mathscr{A}$.

Lemma 5.4. Let $\mathscr{A}$ be a real-time 2-counter Büchi automaton accepting ω-words over the alphabet Σ and $\mathscr{L}(\mathscr{A}) \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ be defined as above. Then $\mathscr{L}(\mathscr{A})$ is accepted by a 1-counter Büchi automaton $\mathscr{B}$.

Proof. We shall explain informally the behaviour of a 1-counter Büchi automaton $\mathscr{B}$ accepting the ω-language $\mathscr{L}(\mathscr{A})$.

We first consider the reading of an ω-word $x \in (A.0^\star.\Sigma.B.0^\star)^\omega$ in the form

$$x = A.0^{n_1}x_1.B.0^{m_1}.A.0^{n_2}x_2.B.0^{m_2}.A \ldots A.0^{n_p}x_p.B.0^{m_p}.A \ldots$$

where, for all integers $i \geqslant 1$, we have n_i, m_i, are positive integers and $x_i \in \Sigma$.

Using the finite control, the automaton $\mathscr{B}$ first checks that the first six letters of x form the initial segment $A.0^5$. Then, when reading the following $(n_1 - 5)$ letters 0, the automaton $\mathscr{B}$, using the finite control, checks that $(n_1 - 5) > 0$ and determines whether $P_2(n_1 - 5) = 0$ and whether $P_3(n_1 - 5) = 0$. Moreover the counter content is increased by one for each letter 0 read. The automaton $\mathscr{B}$ now reads the letter x_1 and guesses a transition of $\mathscr{A}$ leading to

$$x_1 : (q_0, P_2(n_1 - 5), P_3(n_1 - 5)) \mapsto_{\mathscr{A}} (q_1, P_2(n_1 - 5) + j_1, P_3(n_1 - 5) + j'_1).$$

We set $v_1 = 0^{n_1-5}$ and $w_1 = 0^{(n_1-5).2^{j_1}.3^{j'_1}}$. The counter value is now equal to $(n_1 - 5)$ and, when reading letters 0 following x_1, the automaton $\mathscr{B}$ checks that $m_1 \geqslant (n_1 - 5).2^{j_1}.3^{j'_1}$ in such a way that the counter value becomes 0 after having read the $(n_1 - 5).2^{j_1}.3^{j'_1}$ letters 0 following the first letter B. For instance, if $j_1 = j'_1 = 1$, then $|w_1| = |v_1|.6$, so this can be done by decreasing the counter content by one each time six letters 0 are read. The other cases are treated in a similar way. The details are left to the reader.

Note also that the automaton $\mathscr{B}$ has kept in its finite control the value of the state q_1.

We now set $0^{m_1} = w_1.z_1$. We have seen that after having read w_1 the counter value is equal to zero. Now, when reading z_1 the counter content is increased by one for each letter read, so it becomes $|z_1|$ after reading z_1. The automaton $\mathscr{B}$ now reads a letter A and then decreases its counter by one for each letter 0 read, until the counter content is equal to zero. We set $0^{n_2} = u_2.v_2$ with $u_2 = z_1$. The automaton $\mathscr{B}$ now reads the segment v_2. Using the finite control, it checks that $|v_2| > 0$ and determines whether $P_2(|v_2|) = 0$ and whether $P_3(|v_2|) = 0$. Moreover, the counter content is increased by one for each letter 0 read. The automaton $\mathscr{B}$ now reads the letter x_2 and guesses a transition of $\mathscr{A}$ leading to

$$x_2 : (q_1, P_2(|v_2|), P_3(|v_2|)) \mapsto_{\mathscr{A}} (q_2, P_2(|v_2|) + j_2, P_3(|v_2|) + j'_2).$$

We set $w_2 = 0^{|v_2|.2^{j_2}.3^{j'_2}}$. The counter value is now equal to $|v_2|$. The automaton $\mathscr{B}$ now reads the second letter B and, when reading the m_2 letters 0 following this letter B, the automaton $\mathscr{B}$ checks that $m_2 \geqslant |v_2|.2^{j_2}.3^{j'_2}$ in such a way that the counter value becomes 0 after having read the $|v_2|.2^{j_2}.3^{j'_2}$ letters 0 following the second letter B.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

For instance, if $j_2 = 0$ and $j'_2 = -1$, then $|w_2| = |v_2|.3^{-1}$, so this can be done by decreasing the counter content by three each time one letter 0 is read.

And if $j_2 = -1$ and $j'_2 = -1$, then $|w_2| = |v_2|.2^{-1}.3^{-1} = |v_2|.6^{-1}$ so this can be done by decreasing the counter content by six each time one letter 0 is read. The other cases are treated in a similar way. The details are left to the reader.

Note that these different cases can be achieved with the use of λ-transitions, but in such a way that there will be at most 5 consecutive λ-transitions during a run of $\mathscr{B}$ on x. This will be an important useful fact later in the paper.

Note also that the automaton $\mathscr{B}$ has kept the value of the state q_2 in its finite control.

The reading of x by $\mathscr{B}$ continues in the same way. A Büchi acceptance condition can be used to ensure that some state $q_f \in K$ occurs infinitely often in the sequence $(q_i)_{i \geqslant 0}$.

To complete the proof, we note that $\mathscr{R}=(A.0^\star.\Sigma.B.0^\star)^\omega$ is a regular ω-language, so we have only considered the reading by $\mathscr{B}$ of ω-words $x \in \mathscr{R}$. Indeed, if the ω-language $L(\mathscr{B})$ were not included in $\mathscr{R}$, we could replace it by $L(\mathscr{B}) \cap \mathscr{R}$ because the class $\mathbf{BCL}(1)_\omega$ is closed under intersection with regular ω-languages (by a classical construction of the product of automata, the ω-language $\mathscr{R}$ being accepted by a deterministic Muller automaton). □

Lemma 5.5. Let $\mathscr{A}$ be a real-time 2-counter Büchi automaton accepting ω-words over the alphabet Σ, and let $\mathscr{L}(\mathscr{A}) \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ be defined as above. Then $L(\mathscr{A}) = h^{-1}(\mathscr{L}(\mathscr{A}))$, that is, $\forall x \in \Sigma^\omega \quad h(x) \in \mathscr{L}(\mathscr{A}) \longleftrightarrow x \in L(\mathscr{A})$.

Proof. Let $\mathscr{A}$ be a real-time 2-counter Büchi automaton accepting ω-words over the alphabet Σ, and $\mathscr{L}(\mathscr{A}) \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ be defined as above. Let $x \in \Sigma^\omega$ be an ω-word such that $h(x) \in \mathscr{L}(\mathscr{A})$. So $h(x)$ may be written

$$h(x) = A.0^6.x(1).B.0^{6^2}.A.0^{6^2}.x(2).B.0^{6^3}.A.0^{6^3}.x(3).B \ldots B.0^{6^n}.A.0^{6^n}.x(n).B \ldots$$

and, also,

$$h(x) = A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A \ldots A.u_n.v_n.x_n.B.w_n.z_n.A \ldots$$

where, for all integers $i \geqslant 1$, $v_i, w_i \in 0^+$, $u_i, z_i \in 0^\star$, $x_i = x(i) \in \Sigma$, $|u_1| = 5$, $|u_{i+1}| = |z_i|$, and there is a sequence $(q_i)_{i \geqslant 0}$ of states of K and integers $j_i, j'_i \in \{-1; 0; 1\}$, for $i \geqslant 1$, such that for all integers $i \geqslant 1$

$$x_i : (q_{i-1}, P_2(|v_i|), P_3(|v_i|)) \mapsto_{\mathscr{A}} (q_i, P_2(|v_i|) + j_i, P_3(|v_i|) + j'_i)$$

and

$$|w_i| = |v_i|.2^{j_i}.3^{j'_i}$$

with some state $q_f \in F$ occurring infinitely often in the sequence $(q_i)_{i \geqslant 0}$.

In particular, $u_1 = 0^5$ and $u_1.v_1 = 0^6$, so $|v_1| = 1 = 2^0.3^0$. We are going to prove by induction on the integer $i \geqslant 1$ that, for all integers $i \geqslant 1$, $|w_i| = |v_{i+1}| = 2^{P_2(|w_i|)}.3^{P_3(|w_i|)}$. Moreover, setting $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$, we are going to prove that for each integer $i \geqslant 1$

$$x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathscr{A}} (q_i, c_1^{i+1}, c_2^{i+1}) .$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

We have already seen that $|v_1| = 1 = 2^0.3^0$. By hypothesis, there is a state $q_1 \in K$ and integers $j_1, j'_1 \in \{-1; 0; 1\}$ such that $x_1 : (q_0, P_2(|v_1|), P_3(|v_1|)) \mapsto_{\mathscr{A}} (q_1, P_2(|v_1|)+j_1, P_3(|v_1|)+j'_1)$, that is, $x_1 : (q_0, 0, 0) \mapsto_{\mathscr{A}} (q_1, j_1, j'_1)$. So $|w_1| = |v_1|.2^{j_1}.3^{j'_1} = 2^{j_1}.3^{j'_1}$.

We now have $|w_1.z_1| = |u_2.v_2| = 0^{6^2}$ and $|u_2| = |z_1|$, so $|v_2| = |w_1| = 2^{j_1}.3^{j'_1}$. Setting $c_1^1 = 0$, $c_2^1 = 0$, $c_1^2 = j_1 = P_2(|v_2|)$ and $c_2^2 = j'_1 = P_3(|v_2|)$, we have $x_1 : (q_0, c_1^1, c_2^1) \mapsto_{\mathscr{A}} (q_1, c_1^2, c_2^2)$.

Assume now that, for all integers i, $1 \leqslant i \leqslant n-1$, we have

$$|w_i| = |v_{i+1}| = 2^{P_2(|w_i|)}.3^{P_3(|w_i|)}$$

and

$$x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathscr{A}} (q_i, c_1^{i+1}, c_2^{i+1})$$

where $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$.

We know that there is a state $q_n \in K$ and integers $j_n, j'_n \in \{-1; 0; 1\}$ such that

$$x_n : (q_{n-1}, P_2(|v_n|), P_3(|v_n|)) \mapsto_{\mathscr{A}} (q_n, P_2(|v_n|) + j_n, P_3(|v_n|) + j'_n),$$

that is,

$$x_n : (q_{n-1}, c_1^n, c_2^n) \mapsto_{\mathscr{A}} (q_n, c_1^n + j_n, c_2^n + j'_n).$$

So,

$$|w_n| = |v_n|.2^{j_n}.3^{j'_n} = 2^{c_1^n+j_n}.3^{c_2^n+j'_n}.$$

On the other hand, $|w_n.z_n| = |u_{n+1}.v_{n+1}| = 0^{6^{n+1}}$ and $|u_{n+1}| = |z_n|$, so

$$|v_{n+1}| = |w_n| = 2^{c_1^n+j_n}.3^{c_2^n+j'_n} = 2^{c_1^{n+1}}.3^{c_2^{n+1}}$$

by setting $c_1^{n+1} = P_2(|v_{n+1}|)$ and $c_2^{n+1} = P_3(|v_{n+1}|)$. So we have

$$x_n : (q_{n-1}, c_1^n, c_2^n) \mapsto_{\mathscr{A}} (q_n, c_1^{n+1}, c_2^{n+1}).$$

Finally, we have now proved the claim by induction. If for all integers $i \geqslant 1$ we set $c_1^i = P_2(|v_i|)$ and $c_2^i = P_3(|v_i|)$,

$$x_i : (q_{i-1}, c_1^i, c_2^i) \mapsto_{\mathscr{A}} (q_i, c_1^{i+1}, c_2^{i+1}).$$

But there is some state $q_f \in K$ that occurs infinitely often in the sequence $(q_i)_{i\geqslant 1}$. This implies that $(q_{i-1}, c_1^i, c_2^i)_{i\geqslant 1}$ is a successful run of $\mathscr{A}$ on x, so $x \in L(\mathscr{A})$.

Conversely, it is easy to see that if $x \in L(\mathscr{A})$, then $h(x) \in \mathscr{L}(\mathscr{A})$. This concludes the proof of Lemma 5.5. □

Remark 5.6. The simulation, during the reading of $h(x)$ by the 1-counter Büchi automaton $\mathscr{B}$, of the behaviour of the real-time 2-counter Büchi automaton $\mathscr{A}$ reading x, can be achieved, using a coding of the content (c_1, c_2) of two counters by a single integer $2^{c_1}.3^{c_2}$ and the **special shape** of ω-words in $h(\Sigma^\omega)$ that allows the propagation of the counter value of $\mathscr{B}$. This will be sufficient here, because of the previous lemmas, and, in particular, because of the fact that $h(\Sigma^\omega)^-$ is in the class **r-BCL**$(1)_\omega$, and we can now finish the proof of Proposition 5.1.

Proof of Proposition 5.1 continued. Let $\alpha \geqslant 2$ be an ordinal. Assume that there is an ω-language $L(\mathscr{A}) \subseteq \Sigma^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

of rank α) and is accepted by a real-time 2-counter Büchi automaton $\mathscr{A}$. By Lemma 5.2, $h(\mathscr{L}) \cup h(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α). On the other hand, Lemma 5.5 states that $L(\mathscr{A}) = h^{-1}(\mathscr{L}(\mathscr{A}))$, and this implies that $h(L(\mathscr{A})) \cup h(\Sigma^\omega)^- = \mathscr{L}(\mathscr{A}) \cup h(\Sigma^\omega)^-$. But we know by Lemmas 5.3 and 5.4 that the ω-languages $h(\Sigma^\omega)^-$ and $\mathscr{L}(\mathscr{A})$ are in the class $\mathbf{BCL}(1)_\omega$, so their union is also accepted by a 1-counter Büchi automaton. Therefore, $h(L(\mathscr{A})) \cup h(\Sigma^\omega)^-$ is an ω-language in the class $\mathbf{BCL}(1)_\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

We now want to find an ω-language in the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

On the one hand, we have proved that $h(\Sigma^\omega)^-$ is accepted by a *real-time* 1-counter Büchi automaton. On the other hand, we have proved that $\mathscr{L}(\mathscr{A})$ is accepted by a (non-real-time) 1-counter Büchi automaton $\mathscr{B}$. However, we have seen in the proof of Lemma 5.4, that at most 5 consecutive λ-transitions can occur during the reading of an ω-word x by $\mathscr{B}$.

Now consider the mapping $\phi : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ that is defined for all $x \in (\Sigma \cup \{A, B, 0\})^\omega$ by

$$\phi(x) = F^5.x(1).F^5.x(2).F^5.x(3) \dots F^5.x(n).F^5.x(n+1).F^5 \dots$$

The function ϕ is continuous and separates two successive letters of x by five letters F. We can prove, as in the proof of Lemma 4.2, that if $L \subseteq (\Sigma \cup \{A, B, 0\})^\omega$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α) for some ordinal $\alpha \geqslant 2$, then $\phi(L)$ is a subset of $(\Sigma \cup \{A, B, F, 0\})^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

Thus, the ω-language $\phi(\mathscr{L}(\mathscr{A}) \cup h(\Sigma^\omega)^-)$ is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

Moreover, it is easy to see that $\phi(\mathscr{L}(\mathscr{A}))$ is accepted by a *real-time* 1-counter Büchi automaton $\mathscr{B}'$. The automaton $\mathscr{B}'$ checks with its finite control that an input ω-word is in the form $\phi(x)$ for some $x \in (\Sigma \cup \{A, B, 0\})^\omega$. And $\mathscr{B}'$ simulates the reading of x by $\mathscr{B}$, the λ-transitions of $\mathscr{B}$ occuring during the reading, in *real time*, of letters F of the ω-word $\phi(x)$.

Finally, $\phi(\mathscr{L}(\mathscr{A}) \cup h(\Sigma^\omega)^-) = \phi(\mathscr{L}(\mathscr{A})) \cup \phi(h(\Sigma^\omega)^-))$ is the union of two ω-languages in $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$, so it is in $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ and is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α).

This concludes the proof of Proposition 5.1 for the integer $k = 2$.

We now explain the modifications required to prove Proposition 5.1 for the integer $k = 8$. We assume that $\alpha \geqslant 2$ is an ordinal and that there is an ω-language $L(\mathscr{A}) \subseteq \Sigma^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α) and is accepted by a real-time 8-counter Büchi automaton $\mathscr{A}$.

We first modify the coding of ω-words that was given by the mapping h. We replace the number $6 = 2.3$ by the product of the first eight prime numbers:

$$K = 2.3.5.7.11.13.17.19 = 9699690\,.$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Then an ω-word $x \in \Sigma^\omega$ will be coded by the ω-word

$$h_K(x) = A.0^K.x(1).B.0^{K^2}.A.0^{K^2}.x(2).B.0^{K^3}.A.0^{K^3}.x(3).B\ldots B.0^{K^n}.A.0^{K^n}.x(n).B\ldots$$

The mapping $h_K : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$ is continuous and we can prove, as in Lemma 5.2, that $h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^-$ is a subset of $(\Sigma \cup \{A, B, 0\})^\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α). As in Lemma 5.3, we can prove that $h_K(\Sigma^\omega)^-$ is in the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$.

Next, for each prime number $p \in \{2; 3; 5; 7; 11; 13; 17; 19\}$, and each positive integer $N \geqslant 1$, we use $P_p(N)$ to denote the positive integer such that $p^{P_p(N)}$ is the greatest power of p that divides N.

We define the ω-language $\mathscr{L}(\mathscr{A})$ as the set of ω-words over the alphabet $\Sigma \cup \{A, B, 0\}$ in the form

$$A.u_1.v_1.x_1.B.w_1.z_1.A.u_2.v_2.x_2.B.w_2.z_2.A\ldots A.u_n.v_n.x_n.B.w_n.z_n.A\ldots$$

where, for all integers $i \geqslant 1$, $v_i, w_i \in 0^+$, $u_i, z_i \in 0^\star$, $|u_1| = K - 1$, $|u_{i+1}| = |z_i|$ and there is a sequence $(q_i)_{i\geqslant 0}$ of states of K and integers $j_i^1, j_i^2, \ldots, j_i^8, \in \{-1; 0; 1\}$, for $i \geqslant 1$, such that for all integers $i \geqslant 1$

$$x_i : (q_{i-1}, P_2(|v_i|), P_3(|v_i|), \ldots, P_{19}(|v_i|)) \mapsto_{\mathscr{A}} \\ (q_i, P_2(|v_i|) + j_i^1, P_3(|v_i|) + j_i^2, \ldots, P_{19}(|v_i|) + j_i^8)$$

and

$$|w_i| = |v_i|.2^{j_i^1}.3^{j_i^2}.\ldots.19^{j_i^8}$$

and some state $q_f \in F$ occurs infinitely often in the sequence $(q_i)_{i\geqslant 0}$.

Applying the same ideas as in the proofs of Lemmas 5.4 and 5.5, we can prove that $\mathscr{L}(\mathscr{A})$ is accepted by a 1-counter Büchi automaton and that $L(\mathscr{A}) = h_K^{-1}(\mathscr{L}(\mathscr{A}))$.

The essential change is that now the content $(c_1, c_2, \ldots, c_8)$ of eight counters is coded by the product $2^{c_1}.3^{c_2}.\ldots.(17)^{c_7}.(19)^{c_8}$.

Note that $\mathscr{L}(\mathscr{A})$ is again accepted by a (non-real-time) 1-counter Büchi automaton $\mathscr{B}$. However, there are now at most $(K - 1)$ consecutive λ-transitions that can occur during the reading of an ω-word x by $\mathscr{B}$.

So we now define the mapping $\phi_K : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ by:

— For all $x \in (\Sigma \cup \{A, B, 0\})^\omega$,

$$\phi_K(x) = F^{K-1}.x(1).F^{K-1}.x(2).F^{K-1}.x(3)\ldots F^{K-1}.x(n).F^{K-1}.x(n+1).F^{K-1}\ldots$$

The function ϕ_K is continuous since the function ϕ was. The end of the proof is unchanged, so we infer that $\phi_K(h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^-)$ is an ω-language in the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ that is $\mathbf{\Sigma}^0_\alpha$-complete (respectively, $\mathbf{\Pi}^0_\alpha$-complete, $\mathbf{\Sigma}^0_\alpha$ of rank α, $\mathbf{\Pi}^0_\alpha$ of rank α). □

From the results of Section 4 and Proposition 5.1, we can now state the following result.

Theorem 5.7. Let $\mathscr{C}$ be a class of ω-languages such that

$$\mathbf{r}\text{-}\mathbf{BCL}(1)_\omega \subseteq \mathscr{C} \subseteq \Sigma^1_1.$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

(a) The Borel hierarchy of the class $\mathscr{C}$ is equal to the Borel hierarchy of the class Σ_1^1.
(b) $\gamma_2^1 = Sup \ \{\alpha \mid \exists L \in \mathscr{C}$ such that L is a Borel set of rank $\alpha\}$.
(c) For every non-null ordinal $\alpha < \omega_1^{\mathrm{CK}}$, there exists some $\mathbf{\Sigma}_\alpha^0$-complete and some $\mathbf{\Pi}_\alpha^0$-complete ω-languages in the class $\mathscr{C}$.

Note that (b) and (c) follow from (a) and from the known results about the Borel hierarchy of the class Σ_1^1.

6. Wadge hierarchy of ω-languages in r-BCL$(1)_\omega$

We now introduce the Wadge hierarchy, which is a great refinement of the Borel hierarchy defined via reductions by continuous functions, (Duparc 2001; Wadge 1983).

Definition 6.1 (Wadge 1983). 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' ($L \leqslant_W L'$) if and only if there exists a continuous function $f : X^\omega \to Y^\omega$ such that $L = f^{-1}(L')$.

L and L' are Wadge equivalent if and only if $L \leqslant_W L'$ and $L' \leqslant_W L$. This will be denoted by $L \equiv_W L'$. And we shall say that $L <_W L'$ if and only if $L \leqslant_W L'$ but not $L' \leqslant_W L$.

A set $L \subseteq X^\omega$ is said to be self-dual if and only if $L \equiv_W L^-$; otherwise it is said to be non-self-dual.

The relation $\leqslant_W$ 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 $\leqslant_W$ and $\equiv_W$.

For $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$, if $L \leqslant_W L'$ 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, this 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.

Note 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 6.2. Let L be a subset of X^ω. The Wadge class of L is

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

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

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

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Definition 6.3. 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 alternately write letters a_n of X for Player 1 and b_n of Y for Player 2.

After ω steps, Player 1 has written an ω-word $a \in X^\omega$ and Player 2 has written an ω-word $b \in Y^\omega$. Player 2 is allowed to skip, even infinitely often, provided he really writes an ω-word in ω steps.

Player 2 wins the game if and only if $[a \in L \leftrightarrow b \in L']$, that is, if and only if

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

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 $f : X^+ \to Y \cup \{s\}$.

σ is a winning stategy for Player 1 if and only if he always wins a game when he uses the strategy σ, that is, when the n^{th} letter he writes is given by $a_n = \sigma(b_1 \ldots b_{n-1})$, 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 Kechris (1995)), with X a Borel set, is determined, and this implies the following theorem.

Theorem 6.4 (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 $L \leqslant_W L'$ if and only if Player 2 has a winning strategy in the game $W(L, L')$.

Theorem 6.5 (Wadge). Up to the complement and $\equiv_W$, the class of Borel subsets of X^ω for a finite alphabet X is a well-ordered hierarchy. There is an ordinal $|WH|$, called the length of the hierarchy, and a map d_W^0 from WH onto $|WH| - \{0\}$, such that for all $L, L' \subseteq X^\omega$:

$$d_W^0 L < d_W^0 L' \leftrightarrow L <_W L'$$

and

$$d_W^0 L = d_W^0 L' \leftrightarrow [L \equiv_W L' \text{ or } L \equiv_W L'^-] .$$

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. This is described in Wadge (1983) and Duparc (2001) by the use of Veblen functions.

It is natural to ask for the Wadge hierarchy of classes of ω-languages accepted by finite machines, like **X**-automata. The Wadge hierarchy of regular ω-languages, now

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

called the Wagner hierarchy, has been effectively determined by Wagner; it has length ω^ω (Wagner 1979; Selivanov 1995; 1998). Wilke and Yoo proved in Wilke and Yoo (1995) that one can compute in polynomial time the Wadge degree of an ω-regular language. The Wadge hierarchy of ω-languages accepted by Muller *deterministic* one blind (that is, without zero-test) counter automata is an effective extension of the Wagner hierarchy studied in Finkel (2001c). Its extension to *deterministic* context free ω-languages has been determined by Duparc, its length is $\omega^{(\omega^2)}$ (Duparc *et al.* 2001; Duparc 2003) but we do not yet know whether it is effective. Selivanov has recently determined the Wadge hierarchy of ω-languages accepted by *deterministic* Turing machines; its length is $(\omega_1^{\mathrm{CK}})^\omega$ (Selivanov 2003a; 2003b).

In previous papers (Finkel 2001a; 2001d; 2003a; 2003b) we have used the work of Duparc on the Wadge hierarchy of Borel sets (Duparc 2001) to give inductive constructions for some $\mathbf{\Delta}^0_\omega$ context free ω-languages in ε_ω Wadge degrees, where ε_ω is the ω^{th} fixed point of the ordinal exponentiation of base ω, and also some $\mathbf{\Sigma}^0_\omega$-complete context free ω-languages. Note that the Wadge hierarchy of *non-deterministic* context-free ω-languages is not effective.

Here we will now show the following very surprising result, which extends Theorem 5.7.

Theorem 6.6. The Wadge hierarchy of the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$, hence also of the class $\mathbf{CFL}_\omega$, or of every class $\mathscr{C}$ such that $\mathbf{r}$-$\mathbf{BCL}(1)_\omega \subseteq \mathscr{C} \subseteq \Sigma^1_1$, is the Wadge hierarchy of the class Σ^1_1 of ω-languages accepted by Turing machines with a Büchi acceptance condition.

To prove this result, we will first consider non-self-dual sets. We recall the definition of Wadge degrees introduced by Duparc in Duparc (2001), which is a slight modification of the previous one.

Definition 6.7.

(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, $L >_W \emptyset$).

We are now going to introduce the operation of the sum of sets of infinite words, which has as counterpart the ordinal addition over Wadge degrees.

Definition 6.8 (Wadge, see Duparc (2001)). Assume that $X \subseteq Y$ are two finite alphabets, that $Y - X$ contains 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

$$L' + L =_{df} L \cup \{u.a.\beta \mid u \in X^\star, (a \in X_+ \text{ and } \beta \in L') \text{ or } (a \in X_- \text{ and } \beta \in L'^-)\}.$$

This operation is closely related to the *ordinal sum*, as stated in the following theorem.

Theorem 6.9 (Wadge, see Duparc (2001)). Let $X \subseteq Y$, with $Y - X$ containing at least two elements, and let $L \subseteq X^\omega$ and $L' \subseteq Y^\omega$ be non-self-dual Borel sets. Then $(L + L')$ is a non-self-dual Borel set and $d_w(L' + L) = d_w(L') + d_w(L)$.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

A player in charge of a set $L' + L$ in a Wadge game is like a player in charge of the set L except that he can, at any step of the game, erase his previous play and choose to be this time in charge of L' or L'^{-}. Note that he can do this only one time during a play. We shall use this property below.

Lemma 6.10. Let $L \subseteq \Sigma^\omega$ be a non-self-dual Borel set such that $d_w(L) \geqslant \omega$. Then $L \equiv_W \emptyset + L$.

Note that in the above lemma, $\emptyset$ is viewed as the empty set over an alphabet Γ such that $\Sigma \subseteq \Gamma$ and cardinal $(\Gamma - \Sigma) \geqslant 2$.

Proof. Assume that $L \subseteq \Sigma^\omega$ is a non-self-dual Borel set and that $d_w(L) \geqslant \omega$. We know that $\emptyset$ is a non-self-dual Borel set and that $d_w(\emptyset) = 1$. Thus, by Theorem 6.9, we have $d_w(\emptyset + L) = d_w(\emptyset) + d_w(L) = 1 + d_w(L)$. But, by hypothesis, $d_w(L) \geqslant \omega$, and this implies that $1 + d_w(L) = d_w(L)$. So we have proved that $d_w(\emptyset + L) = d_w(L)$.

On the other hand, L is non-self-dual and $d_w(\emptyset + L) = d_w(L)$ implies that only two cases may arise: either $\emptyset + L \equiv_W L$ or $\emptyset + L \equiv_W L^-$.

But it is easy to see that $L \leqslant_W \emptyset + L$. To this end, consider the Wadge game $W(L, \emptyset + L)$. Player 2 clearly has a winning strategy that consists of copying the play of Player 1, so $L \leqslant_W \emptyset + L$. This implies that $\emptyset + L \equiv_W L^-$ cannot hold, so $\emptyset + L \equiv_W L$. □

Lemma 6.11. Let $L \subseteq \Sigma^\omega$ be a non-self-dual Borel set accepted by a Turing machine with a Büchi acceptance condition. Then there is an ω-language $L' \in \mathbf{r\text{-}BCL}(8)_\omega$ such that $L \equiv_W L'$.

Proof. It is well known that there are regular ω-languages of every finite Wadge degree (Staiger 1997; Selivanov 1998). These ω-languages are Boolean combinations of open sets. So we have only to consider the case of non-self-dual Borel sets of Wadge degrees greater than or equal to ω.

So, let $L \subseteq \Sigma^\omega$ be a non-self-dual Borel set accepted by a Turing machine with a Büchi acceptance condition (in particular, L is in the class $\mathbf{BCL}(2)_\omega$) such that $d_w(L) \geqslant \omega$.

Lemma 4.3 states that there exists an integer $S \geqslant 1$ such that $\theta_S(L)$ is in the class $\mathbf{r\text{-}BCL}(8)_\omega$, where E is a new letter not in Σ and $\theta_S : \Sigma^\omega \to (\Sigma \cup \{E\})^\omega$ is the function defined, for all $x \in \Sigma^\omega$, by

$$\theta_S(x) = x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}.x(n+1).E^{S^{n+1}}\ldots$$

We are going to prove that $\theta_S(L) \equiv_W L$.

First, it is easy to see that $L \leqslant_W \theta_S(L)$. In order to prove this, we can consider the Wadge game $W(L, \theta_S(L))$. It is easy to see that Player 2 has a winning strategy in this game that consists of copying the play of Player 1, except that Player 2 adds letters E in such a way that he has written the initial word

$$x(1).E^S.x(2).E^{S^2}.x(3).E^{S^3}.x(4)\ldots x(n).E^{S^n}$$

while Player 1 has written the initial word

$$x(1).x(2).x(3).x(4)\ldots x(n)\,.$$

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Note that one can admit that a player writes a finite word at each step of the play instead of a single letter. This does not change the winner of a Wadge game.

To prove that $\theta_S(L) \leqslant_W L$, it suffices to prove that $\theta_S(L) \leqslant_W \emptyset + L$ because Lemma 6.10 states that $\emptyset + L \equiv_W L$. Consider the Wadge game $W(\theta_S(L), \emptyset + L)$.

Player 2 has a winning strategy in this game that consists of first copying the play of Player 1 except that Player 2 skips when Player 1 writes a letter E. He continues forever with this strategy if the word written by Player 1 is always a prefix of some ω-word of $\theta_S(\Sigma^\omega)$. Then, after ω steps Player 1 has written an ω-word $\theta_S(x)$ for some $x \in \Sigma^\omega$, and Player 2 has written x. So in that case $\theta_S(x) \in \theta_S(L)$ if and only if $x \in L$ if and only if $x \in \emptyset + L$.

However, if at some step of the game, Player 1 'goes out of' the closed set $\theta_S(\Sigma^\omega)$ because the word he has now written is not a prefix of any ω-word of $\theta_S(\Sigma^\omega)$, its final word will certainly be outside $\theta_S(\Sigma^\omega)$, and hence also outside $\theta_S(L)$. Player 2 can now write a letter of $\Gamma - \Sigma$ in such a way that he is now like a player in charge of the emptyset, so he can now write an ω-word u such that his final ω-word will be outside $\emptyset + L$. Thus, Player 2 wins this game too.

Finally, we have proved that $L \leqslant_W \theta_S(L) \leqslant_W L$, so $\theta_S(L) \equiv_W L$, which concludes the proof. □

Lemma 6.12. Let $L \subseteq \Sigma^\omega$ be a non-self-dual Borel set in the class $\mathbf{r\text{-}BCL}(8)_\omega$. Then there is an ω-language $L' \in \mathbf{r\text{-}BCL}(1)_\omega$ such that $L \equiv_W L'$.

Proof. As in the preceding proof, we only need to consider ω-languages with Wadge degrees greater than or equal to ω.

So, let $L = L(\mathscr{A}) \subseteq \Sigma^\omega$ be a non-self-dual Borel set acccepted by a real-time 8-counter Büchi automaton $\mathscr{A}$ such that $d_w(L) \geqslant \omega$. We have shown in the preceding section that $\phi_K(h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^-)$ is in the class $\mathbf{r\text{-}BCL}(1)_\omega$, where h_K is the continuous mapping $h_K : \Sigma^\omega \to (\Sigma \cup \{A, B, 0\})^\omega$ defined by:

— For all $x \in \Sigma^\omega$,

$$h_K(x) = A.0^K.x(1).B.0^{K^2}.A.0^{K^2}.x(2).B.0^{K^3}.A.0^{K^3}.x(3).B \dots B.0^{K^n}.A.0^{K^n}.x(n).B \dots$$

And the mapping $\phi_K : (\Sigma \cup \{A, B, 0\})^\omega \to (\Sigma \cup \{A, B, F, 0\})^\omega$ is defined by:

— For all $x \in (\Sigma \cup \{A, B, 0\})^\omega$,

$$\phi_K(x) = F^{K-1}.x(1).F^{K-1}.x(2).F^{K-1}.x(3) \dots F^{K-1}.x(n).F^{K-1}.x(n+1).F^{K-1} \dots$$

We can now prove, using the fact that $d_w(L) \geqslant \omega$, and by a very similar reasoning to that in the proof of Lemma 6.11, that

$$L \equiv_W h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^- \equiv_W \phi_K(h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^-).$$

But $\phi_K(h_K(L(\mathscr{A})) \cup h_K(\Sigma^\omega)^-)$ is in the class $\mathbf{r\text{-}BCL}(1)_\omega$, which concludes the proof. □

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Proof of Theorem 6.6 continued. Let $L \subseteq \Sigma^\omega$ be a Borel set accepted by a Turing machine with a Büchi acceptance condition (in particular, L is in the class $\mathbf{BCL}(2)_\omega$). 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 $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$. If L is non-self-dual and its Wadge degree is greater than or equal to ω, we can infer from Lemmas 6.11 and 6.12 that there is an ω-language $L' \in \mathbf{r}$-$\mathbf{BCL}(1)_\omega$ such that $L \equiv_W L'$.

We still need 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 $\Sigma_1.L_1 \cup \Sigma_2.L_2$, 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 $L_{(u_1)} = u_1.\Sigma^\omega \cap L$ and $L_{(u_2)} = u_2.\Sigma^\omega \cap L$ for some $u_1, u_2 \in \Sigma^\star$, see Duparc (2003). So, if $L \subseteq \Sigma^\omega$ is a self-dual Borel set accepted by a Turing machine with a Büchi acceptance condition, then $L \equiv_W \Sigma_1.L_1 \cup \Sigma_2.L_2$, 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 a Turing machine with a Büchi acceptance condition. We have already proved that there is an ω-language $L_1' \in \mathbf{r}$-$\mathbf{BCL}(1)_\omega$ such that $L_1' \equiv_W L_1$ and an ω-language $L_2' \in \mathbf{r}$-$\mathbf{BCL}(1)_\omega$ such that $L_2'^- \equiv_W L_2$. Thus, $L \equiv_W \Sigma_1.L_1 \cup \Sigma_2.L_2 \equiv_W \Sigma_1.L_1' \cup \Sigma_2.L_2'$ and $\Sigma_1.L_1' \cup \Sigma_2.L_2'$ is in the class $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$. □

Remark 6.13. In the above we have only considered the Wadge hierarchy of **Borel sets**. If we assume the axiom of $\mathbf{\Sigma}_1^1$-determinacy, then Theorem 6.5 can be extended by considering the class of analytic sets instead of the class of Borel sets. In fact, in that case any set that is analytic but not Borel is $\mathbf{\Sigma}_1^1$-complete, see Kechris (1995). So there is only one more Wadge degree containing $\mathbf{\Sigma}_1^1$-complete sets. We have already proved in Finkel (2003a) that there is a $\mathbf{\Sigma}_1^1$-complete set accepted by a Büchi 1-counter automaton, and it is easy to see from the proof that one can find such a $\mathbf{\Sigma}_1^1$-complete set accepted by a Büchi 1-counter *real-time* automaton.

Remark 6.14. The result given by Theorem 5.7 can now be deduced from Theorem 6.6, and can be seen as a particular case of this result because the Wadge hierarchy is a refinement of the Borel hierarchy and, for each countable non-null ordinal γ, $\mathbf{\Sigma}_\gamma^0$-complete sets (respectively, $\mathbf{\Pi}_\gamma^0$-complete sets) form a single equivalence class of $\equiv_W$, that is, a single Wadge degree (Kechris 1995). However, we have preferred to present the results given in this paper by first considering the Borel hierarchy. In this way the reader who is only interested in the Borel hierarchy of ω-languages can read this part and skip Section 6 on the Wadge hierarchy.

7. Concluding remarks

We have proved that the Borel and the Wadge hierarchies of classes $\mathbf{r}$-$\mathbf{BCL}(1)_\omega$ and $\mathbf{CFL}_\omega$ are also the Borel and Wadge hierarchies of the class Σ_1^1. The methods used in this paper are different from those used in previous papers on context free ω-languages (Finkel 2001a; 2001d; 2003a; 2003b), where we gave an inductive construction of some $\mathbf{\Delta}_\omega^0$ context free ω-languages of a given Borel rank or Wadge degree, using work of Duparc on the Wadge hierarchy of $\mathbf{\Delta}_\omega^0$ Borel sets (Duparc 2001). However, it will be possible to combine both methods for the effective construction of ω-languages in the

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

class **r-BCL**$(1)_\omega$, and of 1-counter Büchi automata accepting them, of a given Wadge degree among the ε_ω degrees obtained in Finkel (2001d) for $\mathbf{\Delta}^0_\omega$ context free ω-languages.

Finally, in another paper (Finkel 2006), using the results of this paper and applying similar methods to the study of topological properties of infinitary rational relations, we prove that their Wadge and Borel hierarchies are equal to the corresponding hierarchies of the classes **r-BCL**$(1)_\omega$, $\mathbf{CFL}_\omega$ or Σ^1_1.

Acknowledgements

We are grateful to the anonymous referee for useful comments on a preliminary version of this paper.

References

Autebert, J.-M., Berstel, J. and Boasson, L. (1996) Context free languages and pushdown automata. In: *Handbook of formal languages* **1**, Springer-Verlag.

Berstel, J. (1979) *Transductions and context free languages*, Teubner Studienbücher Informatik.

Cachat, T. (2002) Symbolic strategy synthesis for games on pushdown graphs. In: Proceedings of the International Conference ICALP 2002. *Springer-Verlag Lecture Notes in Computer Science* **2380** 704–715.

Cachat, T., Duparc, J. and Thomas, W. (2003) Solving pushdown games with a Σ_3-winning condition. In: Proceedings of the 11th Annual Conference of the European Association for Computer Science Logic, CSL 2002. *Springer-Verlag Lecture Notes in Computer Science* **2471** 322–336.

Carton, O. and Perrin, D. (1997) Chains and superchains for ω-rational sets, automata and semigroups. *International Journal of Algebra and Computation* **7** (7) 673–695.

Carton, O. and Perrin, D. (1999) The Wagner hierarchy of ω-rational sets. *International Journal of Algebra and Computation* **9** (5) 597–620.

Cohen, R. S. and Gold, A. Y. (1977) Theory of ω-languages, parts one and two. *Journal of Computer and System Science* **15** 169–208.

Cohen, R. S. and Gold, A. Y. (1978) ω-computations on deterministic pushdown machines. *Journal of Computer and System Science* **16** 275–300.

Duparc, J. (2001) Wadge hierarchy and Veblen hierarchy. Part 1: Borel sets of finite rank. *Journal of Symbolic Logic* **66** (1) 56–86.

Duparc, J. (2003) A hierarchy of deterministic context free ω-languages. *Theoretical Computer Science* **290** (3) 1253–1300.

Duparc, J. and Riss, M. (2005) The missing link for ω-rational sets, automata, and semigroups. *International Journal of Algebra and Computation* (to appear).

Duparc, J., Finkel, O. and Ressayre, J.-P. (2001) Computer science and the fine structure of Borel sets. *Theoretical Computer Science* **257** (1–2) 85–105.

Engelfriet, J. and Hoogeboom, H. J. (1993) X-automata on ω-words. *Theoretical Computer Science* **110** (1) 1–51.

Finkel, O. (2001a) Topological properties of omega context free languages. *Theoretical Computer Science* **262** (1–2) 669–697.

Finkel, O. (2001b) Wadge hierarchy of omega context free languages. *Theoretical Computer Science* **269** (1–2) 283–315.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Finkel, O. (2001c) An effective extension of the Wagner hierarchy to blind counter automata. In: Proceedings of Computer Science Logic, 15th International Workshop, CSL 2001. *Springer-Verlag Lecture Notes in Computer Science* **2142** 369–383.

Finkel, O. (2001d) On the Wadge hierarchy of omega context free languages. In: *Proceedings of the International Workshop on Logic and Complexity in Computer Science, Créteil, France: held in Honor of Anatol Slissenko for his 60th Birthday* 69–79.

Finkel, O. (2003a) Borel hierarchy and omega context free languages. *Theoretical Computer Science* **290** (3) 1385–1405.

Finkel, O. (2003b) On omega context free languages which are Borel sets of infinite rank. *Theoretical Computer Science* **299** (1–3) 327–346.

Finkel, O. (2005) Borel ranks and Wadge degrees of context free ω-languages. In: Proceedings of New Computational Paradigms: First Conference on Computability in Europe, CiE 2005, Amsterdam, The Netherlands. *Springer-Verlag Lecture Notes in Computer Science* **3526** 129–138.

Finkel, O. (2006) On the accepting power of two-tape Büchi automata. In: Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science, STACS 2006. *Springer-Verlag Lecture Notes in Computer Science* **3884** 301–312.

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

Hopcroft, J. E. and Ullman, J. D. (1979) *Introduction to automata theory, languages, and computation*, Addison-Wesley.

Jech, T. (2002) *Set theory, third edition*, Springer-Verlag.

Kechris, A. S. (1995) *Classical descriptive set theory*, Springer-Verlag.

Kechris, A. S., Marker, D. and Sami, R. L. (1989) Π_1^1 Borel sets. *Journal of Symbolic Logic* **54** (3) 915–920.

Lescow, H. and Thomas, W. (1994) Logical specifications of infinite computations. In: de Bakker, J. W., de Roever, W. P. and Rozenberg, G. (eds.) A Decade of Concurrency. *Springer-Verlag Lecture Notes in Computer Science* **803** 583–621.

Linna, M. (1975) On ω-words and ω-computations. *Ann. Univ. Turku. Ser AI* **168** 53.

Louveau, A. (2005) Personal communication.

Moschovakis, Y. N. (1980) *Descriptive set theory*, North-Holland.

Nivat, M. (1977) Mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique et Applications* **11** 311–327.

Nivat, M. (1978) Sur les ensembles de mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique et Applications* **12** (3) 259–278.

Perrin, D. and Pin, J.-E. (2004) *Infinite words, automata, semigroups, logic and games*, Pure and Applied Mathematics **141**, Elsevier.

Selivanov, V. L. (1995) Fine hierarchy of regular ω-languages. In: Proceedings of the International Joint Conference on the Theory and Practice of Software Development TAPSOFT-95, Aarhus, Denmark. *Springer-Verlag Lecture Notes in Computer Science* **915** 277–287.

Selivanov, V. L. (1998) Fine hierarchy of regular ω-languages. *Theoretical Computer Science* **191** 37–59.

Selivanov, V. L. (2003a) 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. *Springer-Verlag Lecture Notes in Computer Science* **2607** 97–108.

Selivanov, V. L. (2003b) Wadge degrees of ω-languages of deterministic Turing machines. *RAIRO-Theoretical Informatics and Applications* **37** (1) 67–83.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press

Serre, O. (2004a) *Contribution à l'étude des jeux sur des graphes de processus à pile*, Ph.D. thesis, Université Paris VII.

Serre, O. (2004b) Games with winning conditions of high Borel complexity. In: Proceedings of the International Conference ICALP 2004. *Springer-Verlag Lecture Notes in Computer Science* **3142** 1150–1162.

Simonnet, P. (1992) *Automates et théorie descriptive*, Ph.D. thesis, Université Paris VII.

Staiger, L. (1986) Hierarchies of recursive ω-languages. *Elektronische Informationsverarbeitung und Kybernetik* **22** (5-6) 219–241.

Staiger, L. (1987) Research in the theory of ω-languages. *Journal of Information Processing and Cybernetics* **23** (8-9) 415–439.

Staiger, L. (1997) ω-languages In: *Handbook of formal languages* **3**, Springer-Verlag 339–387.

Staiger, L. and Wagner, K. (1974) Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. *Elektron. Informationsverarbeit. Kybernetik* **10** 379–392.

Thomas, W. (1990) Automata on infinite objects In: van Leeuwen, J. (ed.) *Handbook of Theoretical Computer Science* **B**: *Formal models and semantics*, Elsevier 135–191.

Thomas, W. (2002) Infinite games and verification (extended abstract of a tutorial). In: Proceedings of the International Conference on Computer Aided Verification CAV 2002. *Springer-Verlag Lecture Notes in Computer Science* **2404** 58–64.

Wadge, W. (1983) *Reducibility and determinateness in the Baire space*, Ph.D. thesis, University of California, Berkeley.

Wagner, K. (1979) On ω-regular sets. *Information and Control* **43** (2) 123–177.

Walukiewicz, I. (2000) Pushdown processes: games and model checking. *Information and Computation* **157** 234–263.

Wilke, T. and Yoo, H. (1995) Computing the Wadge degree, the Lifschitz degree, and the Rabin index of a regular language of infinite words in polynomial time. In: Mosses, P. D., Nielsen, M. and I Schwartzbach, M. (eds.) TAPSOFT 95. *Springer-Verlag Lecture Notes in Computer Science* **915** 288–302.

https://doi.org/10.1017/S0960129506005597 Published online by Cambridge University Press