# REGULAR TREE LANGUAGES IN LOW LEVELS
# OF THE WADGE HIERARCHY

MIKOŁAJ BOJAŃCZYK, FILIPPO CAVALLARI, THOMAS PLACE, AND MICHAŁ SKRZYPCZAK

Institute of Informatics, University of Warsaw
*e-mail address*: bojan@mimuw.edu.pl

University of Turin and University of Lausanne
*e-mail address*: filippo.cavallari@unito.it

LaBRI, Bordeaux University
*e-mail address*: thomas.place@labri.fr

Institute of Informatics, University of Warsaw
*e-mail address*: mskrzypczak@mimuw.edu.pl

ABSTRACT. In this article we provide effective characterisations of regular languages of infinite trees that belong to the low levels of the Wadge hierarchy. More precisely we prove decidability for each of the finite levels of the hierarchy; for the class of the Boolean combinations of open sets $\mathsf{BC}(\mathbf{\Sigma}_1^0)$ (i.e. the union of the first $\omega$ levels); and for the Borel class $\mathbf{\Delta}_2^0$ (i.e. for the union of the first $\omega_1$ levels).

## 1. INTRODUCTION

The space of all infinite trees over a finite alphabet is homeomorphic to the Cantor space. Therefore, it makes sense to ask if a language of infinite trees — in our setting, we are interested in regular ones — is for instance open, Borel, or of specific descriptive set theoretical complexity. As witnessed by a number of conjectures and results [Sku93, Mur08b, FMM16, SW16, CMS17], topologically defined classes often have natural automata counterparts. For instance, in the case of $\omega$-words, the structure of parity *deterministic* automata (defined in terms of Wagner hierarchy) is strictly connected to the Wadge hierarchy, see [Wag79]. In the case of regular tree languages that are Borel, there is a strong connection between the *Borel rank* and priorities used by *weak alternating automata* (see [DM07] and [CMS17]).

Algorithms that determine if a regular language belongs to a subclass $\mathcal{L}$ of regular languages are known as *effective characterisations*. Typically, an effective characterisation comes with a structural description of automata (or algebras) that recognise languages

$$\overbrace{\phantom{\qquad\qquad\qquad}}^{\displaystyle \mathsf{BC}(\boldsymbol{\Sigma}^0_1)} \qquad \overbrace{\phantom{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}}^{\displaystyle \boldsymbol{\Delta}^0_2}$$

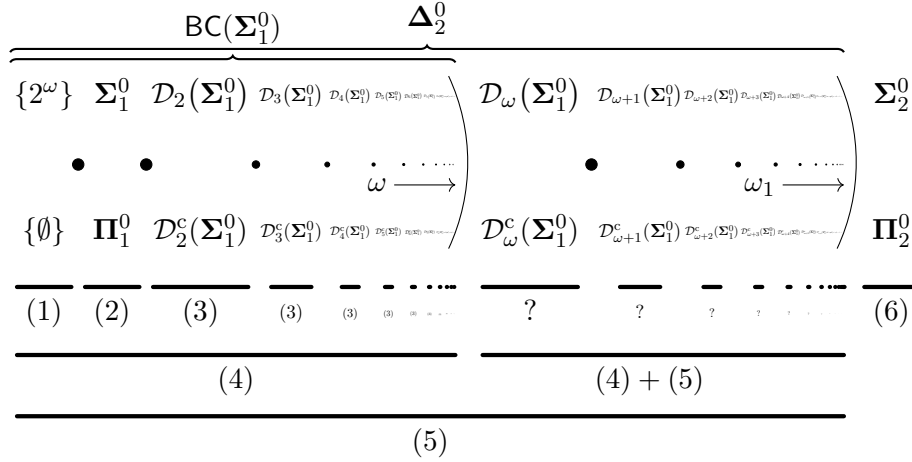| $\{2^\omega\}$ | $\boldsymbol{\Sigma}^0_1$ | $\mathcal{D}_2\big(\boldsymbol{\Sigma}^0_1\big)$ | $\mathcal{D}_3(\boldsymbol{\Sigma}^0_1)$ | $\mathcal{D}_4(\boldsymbol{\Sigma}^0_1)$ $\mathcal{D}_5(\boldsymbol{\Sigma}^0_1)$ ... | $\mathcal{D}_\omega\big(\boldsymbol{\Sigma}^0_1\big)$ | $\mathcal{D}_{\omega+1}(\boldsymbol{\Sigma}^0_1)$ $\mathcal{D}_{\omega+2}(\boldsymbol{\Sigma}^0_1)$ ... | $\boldsymbol{\Sigma}^0_2$ |
|---|---|---|---|---|---|---|---|
| • | • | • | • | $\cdots$ $\omega \longrightarrow$ | • | • $\cdots$ $\omega_1 \longrightarrow$ | |
| $\{\emptyset\}$ | $\boldsymbol{\Pi}^0_1$ | $\mathcal{D}^c_2(\boldsymbol{\Sigma}^0_1)$ | $\mathcal{D}^c_3(\boldsymbol{\Sigma}^0_1)$ | $\mathcal{D}^c_4(\boldsymbol{\Sigma}^0_1)$ $\mathcal{D}^c_5(\boldsymbol{\Sigma}^0_1)$ ... | $\mathcal{D}^c_\omega\big(\boldsymbol{\Sigma}^0_1\big)$ | $\mathcal{D}^c_{\omega+1}(\boldsymbol{\Sigma}^0_1)$ $\mathcal{D}^c_{\omega+2}(\boldsymbol{\Sigma}^0_1)$ ... | $\boldsymbol{\Pi}^0_2$ |
| (1) | (2) | (3) | (3) | (3) (3) ... | ? | ? ? ... | (6) |
| | | (4) | | | | (4) + (5) | |
| | | | | (5) | | | |

Figure 1: The first $\omega_1 + 1$ levels of the Wadge hierarchy for infinite trees; together with their decidability results. The self-dual classes are depicted by •, they are formed by the intersection of the two consecutive non self-dual classes.

from $\mathcal{L}$. The seminal example is Schützenberger's Theorem [Sch65], which says that a regular language is star-free if and only if it is recognised by an aperiodic semigroup. For other examples about finite words, see the survey [PZ15], which discusses effective characterisations for the low levels of the quantifier alternation hierarchy. For examples on $\omega$-words, including topologically motivated ones, see [PP04]. For examples on finite trees, see e.g. [BW08] or a survey [Boj10].

Most of the known effective characterisations speak about languages of words (finite or infinite) or finite trees. The case of regular languages of infinite trees seems to be much more difficult, mainly because of the inherent non-determinism needed to recognise these languages [Blu11, BS13]. Thus, the known examples of effective characterisations are usually limited either to simple classes of sets (e.g. open sets [KW02, Wal02a]) or to restricted classes of languages given as the input (e.g. recognised by deterministic automata [Mur08a]).

In this work we focus on the very low levels of the Borel hierarchy: the class $\mathsf{BC}(\boldsymbol{\Sigma}^0_1)$ of Boolean combinations of open sets; and the self-dual class $\boldsymbol{\Delta}^0_2$ at the second level of the hierarchy. We use algebraic methods for infinite trees, i.e. our characterisations are defined by equations which must be satisfied by the syntactic algebra of a language.

This paper continues a line of work aimed at understanding the algebraic theory of regular languages of infinite trees [Blu11, BS13, BI09a]. The obtained results show that even simple algebras (i.e. not strong enough to distinguish all regular languages or not *complete*, see Subsection 7.1) can be adequate for characterising classes of languages that are sufficiently simple. This opens the possibility that a bit more complex algebraic structure (but a priori not complete) might be enough for the successive levels of the Borel hierarchy, like $\boldsymbol{\Delta}^0_3$.

**Contribution of this article.** Consider a class $\Gamma$ of languages (e.g. the class of open sets $\boldsymbol{\Sigma}^0_1$). Then, an *effective characterisation* of $\Gamma$ is an algorithm for the following decision problem:

**Problem 1.1** (Effective Characterisation of $\Gamma$)**.** Given a representation of a regular language $L$, decide if $L \in \Gamma$.

As explained above, there are multiple results providing effective characterisations for various classes of languages. In this article we focus on the classes of the Wadge hierarchy inside $\mathbf{\Delta}_2^0$, see Figure 1 (the relevant technical notions are introduced in Section 2). Notice that, since regular languages are effectively closed under complement, an effective characterisation of $\Gamma$ provides at the same time an effective characterisation of $\Gamma^c$ and $\Gamma \cap \Gamma^c$. Thus, we will focus on non self-dual classes on one *side* of the hierarchy. Since $\mathrm{Tr}_A$ is homeomorphic to $2^\omega$ (see Fact 2.2) all the results from Subsections 2.4 and 2.5 apply. Therefore, the Wadge hierarchy over $\mathrm{Tr}_A$ introduces the following classes of sets:

(1) The class $\{\mathrm{Tr}_A\}$. A language $L$ belongs to $\{\mathrm{Tr}_A\}$ if and only if $L = \mathrm{Tr}_A$, thus solving the effective characterisation for that class boils down to checking universality of $L$, which reduces to non-emptiness of the complement of $L$ [Rab69].
(2) The class of open sets $\mathbf{\Sigma}_1^0$. That characterisation follows from [KW02, Wal02a].
(3) The classes of the difference hierarchy $\mathcal{D}_n(\mathbf{\Sigma}_1^0)$ for $2 \leq n < \omega$. These classes are characterised in this paper, see Theorem 3.8.
(4) The class $\mathsf{BC}(\mathbf{\Sigma}_1^0) = \bigcup_{n \in \omega} \mathcal{D}_n(\mathbf{\Sigma}_1^0)$ of Boolean combinations of open sets. This is the main contribution of this paper, see Theorem 5.1.
(5) The self-dual class $\mathbf{\Delta}_2^0 = \bigcup_{\xi < \omega_1} \mathcal{D}_\xi(\mathbf{\Sigma}_1^0)$ of the Borel hierarchy. This result was claimed in [FM14], however the arguments there contain a flow, see discussion in Section 6. In this paper we provide a complete argument, see Theorem 6.1.
(6) The class $\mathbf{\Sigma}_2^0$ from the second level of the Borel hierarchy. This class seems to be out of reach of the algebras considered in this paper, see Subsection 7.1. However, an effective characterisation for that class exists, see [CMS17].

What remains open is how to characterise the specific classes $\mathcal{D}_\xi(\mathbf{\Sigma}_1^0)$ for $\omega \leq \xi < \omega_1$. Notice that there are only countably many regular languages and therefore there must exist $\xi_0 < \omega_1$ such that no regular language belongs to $\mathcal{D}_\xi(\mathbf{\Sigma}_1^0)$ for $\xi \geq \xi_0$. However, the value of $\xi_0$ is not known. Duparc and Murlak [DM07] have proved that there exist regular languages in any Wadge degree with Wadge rank less than $\omega^\omega$ (i.e. $\xi_0 \geq \omega^\omega$). We do not know if $\xi_0 = \omega^\omega$, even if this is a quite reasonable conjecture.

**Related work.** First, a series of works [NW05, NW03, Mur08b, FMM16] provide effective characterisations for almost all natural classes when the input is restricted to deterministic automata or their dualised variant — the so-called game automata. These results are based on the *pattern method* saying that the language recognised by a deterministic automaton is complex if and only if the automaton itself contains a *complex pattern*. Unfortunately, there is no known method allowing to extend these techniques to languages involving non-determinism.

Recently, certain new techniques have been developed that show how to deal with the non-determinism of regular languages of infinite trees. The first result of this kind is the reduction of the general Rabin-Mostowski index problem to a certain boundedness problem for cost automata [CL08]. Unfortunately, the latter problem is not known to be decidable. However, the game approach used in the above reduction turned out to work for the lowest indices [CKLV13]. By adopting these techniques, the authors of [SW16] provided a characterisation of Borel sets among languages recognisable by Büchi automata. A similar approach used in [CMS17] provided an effective characterisation of the Borel class

$\mathbf{\Pi}^0_2$ among all regular tree languages. An effective (but not algebraic) characterisation of the class $\mathbf{\Delta}^0_2$ follows directly from that result, however it does not solve the more difficult case of $\mathsf{BC}(\mathbf{\Sigma}^0_1)$.

The paper is based on the conference papers [BP12] and [FM14], see Conclusions for a discussion on relations between the new paper and the original ones.

**Structure.** The paper is structured as follows. In Section 2 we recall some basic notions about words and trees and we set the notation used throughout the article, with a special emphasis on the topological hierarchies involved. In Section 3 we describe a topological game that will be used to obtain effective characterisations of the levels of the Wadge hierarchy up to $\omega$. Subsection 3.1 provides an easy application of the game to prove decidability of each of the first $\omega$ levels of the Wadge hierarchy (i.e. all the finite levels). Subsection 3.2 extends the game to an infinite variant used later to characterise $\mathbf{\Delta}^0_2$. In Section 4 we present the algebraic structure used in this paper to represent regular languages of infinite trees. In Section 5 we state Theorem 5.1 characterising the class $\mathsf{BC}(\mathbf{\Sigma}^0_1)$ in terms of equations defined in the syntactic algebra. The proof of this theorem is spread across Subsections 5.2, 5.3, 5.4 and 5.5. Finally, in Section 6 we state and prove Theorem 6.1 that uses the algebraic tools from Theorem 5.1 to characterise the Borel class $\mathbf{\Delta}^0_2$.

## 2. Basic notions

If $f$ is a function, by $\mathrm{dom}(f)$ we denote the domain of $f$. We denote by $\omega$ the first infinite ordinal and by $\omega_1$ the first uncountable ordinal.

2.1. **Words and trees.** Consider a non-empty set $A$. We call $A$ an *alphabet* if $A$ is finite. Let $A^n$ be the space of the functions of the form $s\colon \{0,\ldots,n-1\} \to A$. Such a function can be represented as a word $s = (s(0),\ldots,s(n-1)) = s_0 \ldots s_{n-1}$ over $A$. If $s = s_0 s_1 \ldots s_{n-1}$ then we say that $n$ is the *length* of $s$, and we denote it by $lt(s)$. The empty word is denoted by $\epsilon$, i.e. $lt(\epsilon) = 0$ and $A^0 = \{\epsilon\}$. By $A^{\leq n}$ we denote the set of words over $A$ of length at most $n$, i.e. $A^{\leq n} \stackrel{\mathrm{def}}{=} A^0 \cup A^1 \cup \ldots \cup A^n$. We denote by $A^*$ the set of all the finite words over $A$: $A^* \stackrel{\mathrm{def}}{=} \bigcup_{n \in \omega} A^n$. By $A^\omega$ we denote the set of infinite words over the alphabet $A$, formally the elements of this space are functions of the form $\alpha\colon \omega \to A$. Such a function can be represented as an infinite sequence $(\alpha(0),\alpha(1),\alpha(2),\ldots) = \alpha_0\alpha_1\alpha_2\ldots$ Finally, we set $A^{\leq\omega} \stackrel{\mathrm{def}}{=} A^* \cup A^\omega$.

If $\alpha \in A^{\leq\omega}$ and $n \in \omega$, we define $\alpha{\upharpoonright}n \stackrel{\mathrm{def}}{=} \alpha_0\alpha_1\ldots\alpha_{n-1} \in A^n$ (if $\alpha$ is finite this definition makes sense only if $n \leq lt(\alpha)$). We say that $s \in A^*$ is a *prefix* of $\alpha \in A^{\leq\omega}$ if $s = \alpha{\upharpoonright}n$ for some $n$; in symbols $s \preceq \alpha$. We write $s \prec \alpha$ if $s \preceq \alpha$ but $s \neq \alpha$. The *concatenation* of $s,t \in A^*$, where $s = s_0 \ldots s_{n-1}$ and $t = t_0 \ldots t_{m-1}$, is the word $s{\char`^}t = st = s_0 \ldots s_{n-1}t_0 \ldots t_{m-1}$. We can also consider the concatenation $s{\char`^}\alpha$ of a finite word $s$ and an infinite word $\alpha$ defined in the obvious way: $s{\char`^}\alpha = s_0 s_1 s_2 \ldots s_{lt(s)-1}\alpha_0\alpha_1\ldots$

Now let us generalise these notions to trees. In this article, we focus on trees with binary branching, where the two *directions* are *left* L and *right* R. A *partial tree* over an alphabet $A$ is a partial function $t\colon \{\text{L},\text{R}\}^* \rightharpoonup A$ with a non-empty prefix-closed domain $\mathrm{dom}(t)$ (i.e. if $s \in \mathrm{dom}(t)$ and $s' \preceq s$ then $s' \in \mathrm{dom}(t)$). A node $u \in \mathrm{dom}(t)$ is either an *internal node* (i.e. both $u{\char`^}\text{L}$ and $u{\char`^}\text{R}$ belong to $\mathrm{dom}(t)$), a *unary node* (i.e. exactly one of $u{\char`^}\text{L}$ and $u{\char`^}\text{R}$ belongs to $\mathrm{dom}(t)$), or a *leaf* (i.e. none of $u{\char`^}\text{L}$ and $u{\char`^}\text{R}$ belongs to $\mathrm{dom}(t)$). For the sake of

readability, we write $u \in t$ to denote that $u \in \mathrm{dom}(t)$ is a node of $t$. The empty sequence $\epsilon$ belongs to every partial tree and it is called the *root* of a tree. A *branch* of a partial tree $t$ is a word $\pi$ such that $\pi{\upharpoonright}n \in t$ for any $n \leq lt(\pi)$ if $\pi$ is finite (resp. for any $n \in \omega$ if $\pi$ is infinite). An infinite branch of a partial tree $t$ is called a *path* of $t$. A node $u$ is *on a branch* (finite or infinite) $\pi$ if it is a prefix of $\pi$, i.e. if $u \preceq \pi$. A partial tree $t$ is *finite* if its domain $\mathrm{dom}(t)$ is finite. A *tree* is a partial tree $t$ with $\mathrm{dom}(t) = \{\mathtt{L},\mathtt{R}\}^*$ (i.e. a *complete tree*). The set of all trees over an alphabet $A$ is denoted $\mathrm{Tr}_A$, that is $\mathrm{Tr}_A \stackrel{\text{def}}{=} \{t \mid t\colon \{\mathtt{L},\mathtt{R}\}^* \to A\}$.

If $p$ and $t$ are partial trees, we say that $p$ is a *prefix* of $t$, and we denote it by $p \sqsubseteq t$, if $\mathrm{dom}(p) \subseteq \mathrm{dom}(t)$ and $p(u) = t(u)$ for any $u \in \mathrm{dom}(p)$. We write $p \sqsubset t$ if $p \sqsubseteq t$ but $p \neq t$. Finally, if $t$ is a partial tree and $u$ is a node of $t$, by $t.u$ we indicate the partial tree $t$ *truncated* in $u$: for any $w$ such that $uw \in \mathrm{dom}(t)$, we have $t.u(w) = t(uw)$.

For $d > 0$ a *d-prefix* of a tree $t$ is the prefix $p \stackrel{\text{def}}{=} t{\upharpoonright}\{\mathtt{L},\mathtt{R}\}^{<d}$, i.e. the prefix of $t$ containing all the nodes at depths smaller than $d$. For instance, the 1-prefix of $t$ consists of the root of $t$ only.

A subset $L \subseteq \mathrm{Tr}_A$ is a *tree language*. *Regular* tree languages are the ones recognised by parity non-deterministic automata or, equivalently, definable in Monadic Second-order Logic (for this equivalence see for example [GTW02]).

2.2. **Polish spaces.** The subsequent subsections recall the topological notions coming from Descriptive Set Theory that we will use throughout the article. We do not aim for completeness, for more details we refer the reader to [Kec95]. In the first subsection we define Polish spaces, that are the main objects studied in Descriptive Set Theory. In the remaining section we introduce the main hierarchies usually considered for Polish spaces, i.e. the Borel, difference, and Wadge hierarchies.

We denote a topological space by $(X, \tau)$, where $X$ is a non-empty set and $\tau$ is a family of subsets of $X$ called *open sets*. If $\tau$ is understood from the context we write just $X$ and suppress $\tau$ from the notation. We say that $X$ is a *Polish space* if $\tau$ is completely metrizable (i.e. there exists a complete metric on $X$ that generates $\tau$) and separable (i.e. there exists a countable dense subset of $X$).

If a space $X$ is known from the context and $A \subseteq X$ then by $A^{\mathrm{c}} \stackrel{\text{def}}{=} X \setminus A$ we denote the complement of $A$ in $X$. Similarly, if $\Gamma$ is a family of subsets of $X$ then $\Gamma^{\mathrm{c}} \stackrel{\text{def}}{=} \{A^{\mathrm{c}} \mid A \in \Gamma\}$.

Consider a non-empty at most countable set $B$. We will now introduce the so-called *prefix topologies* on the spaces $B^\omega$ and $\mathrm{Tr}_B$ by providing explicitly their bases. However, it is worth noticing that these topologies coincide with the *Tychonoff topologies* when $B$ is considered as a discrete topological space, see [Eng89, Section 2.3].

The *prefix topology* on the space of infinite words $B^\omega$ over $B$ is generated by the basic open sets of the form:
$$N_s = \{\alpha \in B^\omega \mid s \prec \alpha\},$$
with $s \in B^*$. When $B = \{0, 1\}$, we obtain the *Cantor space*, denoted by $2^\omega$. When $B = \omega$, we obtain the *Baire space*, denoted by $\omega^\omega$. Every space of the form $B^\omega$ with the prefix topology is Polish. It is easy to check that the prefix topology is completely metrizable: the metric $d(\alpha, \beta) = 2^{-n}$, where $n$ is the minimum index such that $\alpha(n) \neq \beta(n)$ (and $d(\alpha, \beta) = 0$ if $\alpha = \beta$), is complete and it generates the prefix topology. Moreover, if we fix a symbol $c \in B$, the set
$$D = \{s\,\hat{}\,ccc\cdots \mid s \in B^*\}$$

is countable (since $B^*$ is countable) and dense, so $B^\omega$ is separable. In particular, the Cantor space and the Baire space are Polish spaces.

The prefix topology can easily be generalised to trees $\mathrm{Tr}_A$, with basic open sets of the form:
$$N_p = \{t \in \mathrm{Tr}_A \mid p \subset t\},$$
where $p$ is a finite partial tree. Every $N_p$ is actually a clopen (i.e. both open and closed). We denote this topology by $\tau_{\mathrm{pref}}$. The topology $\tau_{\mathrm{pref}}$ is generated by several metrics. The usual metric considered to generate $\tau_{\mathrm{pref}}$ is $d_{\mathrm{pref}}(t_1, t_2) = 2^{-n}$ where $n$ is the minimum length of a node $u$ such that $t_1(u) \neq t_2(u)$ (and $d_{\mathrm{pref}}(t_1, t_2) = 0$ for $t_1 = t_2$). Each open ball of $d_{\mathrm{pref}}$ is a basic clopen set $N_p$ for a certain finite partial tree $p$.

Notice that the metric $d_{\mathrm{pref}}$ satisfies the following strengthening of the triangle inequality:
$$d_{\mathrm{pref}}(x, z) \leq \max\big(d_{\mathrm{pref}}(x, y), d_{\mathrm{pref}}(y, z)\big).$$
Such a metric is called an *ultrametric*, see [Kec95, Exercise 2.2]. This property makes $d_{\mathrm{pref}}$ too rigid for our way of choosing optimal witnesses (see Definition 5.16 of optimal strategy trees). Therefore, we will also consider a different metric, denoted by $\lambda$ and called the *discounted distance*. This metric also generates $\tau_{\mathrm{pref}}$ but has a less intuitive family of open balls. Fix some enumeration $u_0, u_1, \ldots$ of all the nodes in $\{\mathrm{L}, \mathrm{R}\}^*$. Given two trees $t_1$ and $t_2$, for any node $u$, define $\mathsf{dist}(t_1(u), t_2(u)) = 0$ if $t_1(u) = t_2(u)$, 1 in the other case. Let
$$\lambda(t_1, t_2) \overset{\mathrm{def}}{=} \sum_{n \geq 0} \frac{1}{2^n} \cdot \mathsf{dist}\big(t_1(u_n), t_2(u_n)\big).$$

**Fact 2.1.** Regardless of the enumeration $(u_0, u_1, \ldots)$, the prefix and discounted distances yield the same topology.

*Proof.* It is enough to observe that $\tau_{\mathrm{pref}}$ is exactly the product topology obtained by starting from the discrete topology and the discounted distance is exactly the product metric. $\square$

**Fact 2.2.** $\mathrm{Tr}_A$ with the topology $\tau_{\mathrm{pref}}$ is a Polish space homeomorphic to the Cantor space $2^\omega$.

*Proof.* The proof is a standard encoding of one compact product space into another. One can also use a characterisation of the Cantor space, see [Kec95, Theorem 7.4, page 35]. $\square$

2.3. **The Borel hierarchy.** Let $(X, \tau)$ be a topological space. Recall that $\omega_1$ is the first uncountable ordinal. We define, by a transfinite recursion on $1 \leq \xi < \omega_1$, the following classes:
$$\mathbf{\Sigma}_1^0(X) \overset{\mathrm{def}}{=} \{A \subseteq X \mid A \text{ is open}\};$$
$$\mathbf{\Pi}_\xi^0(X) \overset{\mathrm{def}}{=} \{A^{\mathrm{c}} \subseteq X \mid A \in \mathbf{\Sigma}_\xi^0(X)\} = \big(\mathbf{\Sigma}_\xi^0(X)\big)^{\mathrm{c}};$$
$$\mathbf{\Sigma}_\xi^0(X) \overset{\mathrm{def}}{=} \big\{\bigcup_n A_n \mid A_n \in \mathbf{\Pi}_{\xi_n}^0(X),\ 1 \leq \xi_n < \xi,\ n \in \omega\big\}.$$

Moreover, for $1 \leq \xi < \omega_1$, we define the intersection of the two classes $\mathbf{\Delta}_\xi^0(X) \overset{\mathrm{def}}{=} \mathbf{\Sigma}_\xi^0(X) \cap \mathbf{\Pi}_\xi^0(X)$.

$$\begin{array}{ccccc}
\mathbf{\Sigma}_1^0 & & \mathbf{\Sigma}_2^0 & & \cdots \\
& \diagdown & \diagup \quad \diagdown & & \\
\mathsf{BC}(\mathbf{\Sigma}_1^0) \;\text{---}\; \mathbf{\Delta}_2^0 & & \mathsf{BC}(\mathbf{\Sigma}_2^0) \;\text{---}\; \mathbf{\Delta}_3^0 \\
& \diagup & \diagdown \quad \diagup & & \\
\mathbf{\Pi}_1^0 & & \mathbf{\Pi}_2^0 & & \cdots
\end{array}$$
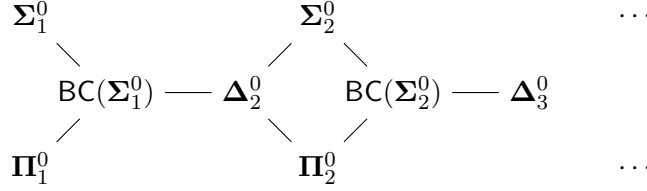
Figure 2: The first levels of the Borel hierarchy.

**Fact 2.3.** For each $1 \leq \xi < \omega_1$, the classes $\mathbf{\Sigma}_\xi^0(X)$ and $\mathbf{\Pi}_\xi^0(X)$ are closed under finite unions and finite intersections. The class $\mathbf{\Delta}_\xi^0(X)$ is also closed under complement and therefore forms a Boolean algebra.

The smallest Boolean algebra containing all the sets from $\mathbf{\Sigma}_\xi^0(X)$ is denoted $\mathsf{BC}(\mathbf{\Sigma}_\xi^0)(X)$. The above fact implies that $\mathsf{BC}(\mathbf{\Sigma}_\xi^0)(X) \subseteq \mathbf{\Delta}_{\xi+1}^0(X)$. For uncountable Polish spaces the inclusion is strict, this fact follows from [Kec95, Exercise 22.26(iii)]).

The *Borel sets* of $X$ are:
$$\mathcal{B}(X) = \bigcup_{\xi \in \omega_1} \mathbf{\Sigma}_\xi^0(X) = \bigcup_{\xi \in \omega_1} \mathbf{\Pi}_\xi^0(X) = \bigcup_{\xi \in \omega_1} \mathbf{\Delta}_\xi^0(X).$$
When the space $X$ is clear from the context, we omit it and write just $\mathbf{\Sigma}_\xi^0$, $\mathbf{\Pi}_\xi^0$, etc. . .

Notice that if $\xi \leq \xi' < \omega_1$ then directly from the definition we know that $\mathbf{\Sigma}_\xi^0 \subseteq \mathbf{\Sigma}_{\xi'}^0$. The following fact shows that the opposite containment does not hold in general.

**Fact 2.4** ([Kec95, Theorem 22.4]). Let $(X, \tau)$ be an uncountable Polish space. Then the Borel hierarchy of $X$ does not collapse i.e. every class $\mathbf{\Sigma}_\xi^0$ is properly contained in $\mathbf{\Sigma}_{\xi+1}^0$.

For the rest of the article we will focus on the first two levels of the Borel hierarchy, as depicted in Figure 2.

2.4. **The difference hierarchy.** The Borel hierarchy is refined by the so-called the *difference hierarchy*, see [Kec95, Section 22.E]. First notice that every ordinal $\theta$ can be uniquely written as $\lambda + n$, where $\lambda$ is 0 or a limit ordinal and $n \in \omega$. We say that the *parity* of $\theta$ is *even* (resp. *odd*) if $n$ is even (resp. odd).

**Definition 2.5.** Let $X$ be a topological space, $\Gamma$ a family of subsets of $X$, and $\theta < \omega_1$ a countable ordinal. A set $A \subseteq X$ is called a $\theta$-*difference of* $\Gamma$ *sets* if and only if there exists a $\theta$-indexed sequence of sets $(A_\eta)_{\eta < \theta} \subseteq \Gamma$ that is non-decreasing (i.e. $A_\eta \subseteq A_{\eta'}$ if $\eta \leq \eta'$) and:
$$x \in A \iff \text{the minimum } \eta < \theta \text{ such that } x \in A_\eta,$$
$$\text{has parity opposite to that of } \theta.$$
The family of all $\theta$-differences of $\Gamma$ sets is denoted $\mathcal{D}_\theta(\Gamma)$. In particular, for each $\xi < \omega_1$ the class $\mathcal{D}_\theta(\mathbf{\Sigma}_\xi^0)(X)$ is the family of all $\theta$-differences of sets from the Borel class $\mathbf{\Sigma}_\xi^0(X)$.

The above definition requires the sequence $(A_\eta)_{\eta < \theta}$ to be non-decreasing and we will focus on $\Gamma = \mathbf{\Sigma}_\xi^0(X)$. These are important assumptions, because of the following remark.

**Remark 2.6.** For each $\xi < \omega_1$ we have $\mathcal{D}_\omega(\mathbf{\Pi}_\xi^0) = \mathbf{\Sigma}_{\xi+1}^0$.
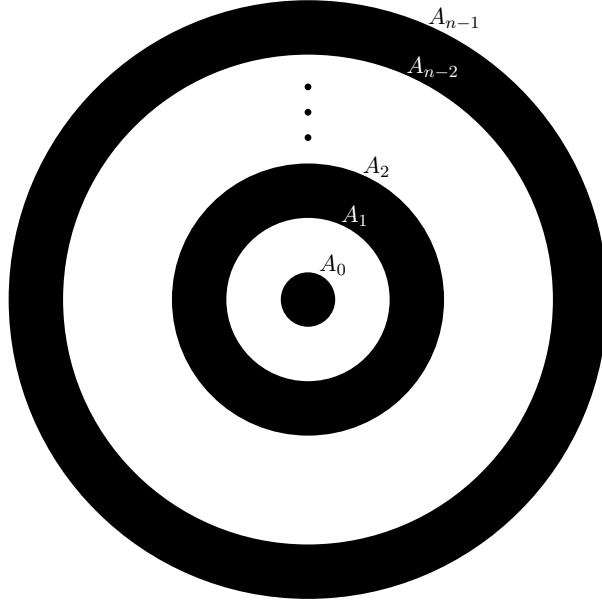
Figure 3: A set $A = A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{n-1} \setminus A_{n-1})$ ($A$ is the union of the black parts) with $A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_{n-2} \subseteq A_{n-1}$ and $A_i \in \mathbf{\Sigma}^0_\xi$ for every $i$.

*Proof.* The inclusion $\mathcal{D}_\omega\big(\mathbf{\Pi}^0_\xi\big) \subseteq \mathbf{\Sigma}^0_{\xi+1}$ follows directly from the definition. For the opposite direction, consider a set $A \in \mathbf{\Sigma}^0_{\xi+1}$. It is easy to check that $A$ can be written as $\bigcup_{n\in\omega} A_n$ with $(A_n)_{n\in\omega}$ non-decreasing and each $A_n$ in $\mathbf{\Pi}^0_\xi$. Then take $A'_n \stackrel{\text{def}}{=} A_{\lfloor n \rfloor}$ and notice that this sequence is also non-decreasing and contains only $\mathbf{\Pi}^0_\xi$ sets. However, $x \in A$ iff $\exists n.\ x \in A_n$ iff the minimum $n$ such that $x \in A'_n$ is even. Therefore, $A$ is an $\omega$-difference of $\mathbf{\Pi}^0_\xi$ sets and $A \in \mathcal{D}_\omega\big(\mathbf{\Pi}^0_\xi\big)$.  $\square$

Notice that for a natural number $n$ we have $A \in \mathcal{D}_n\big(\mathbf{\Sigma}^0_\xi\big)$ if and only if it can be written as follows (see Figure 3):

$$A = A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{n-1} \setminus A_{n-2}) \qquad \text{if } n \text{ is odd,} \qquad (2.1)$$

$$A = (A_1 \setminus A_0) \cup \ldots \cup (A_{n-1} \setminus A_{n-2}) \qquad \text{if } n \text{ is even,} \qquad (2.2)$$

with $A_0, \ldots, A_{n-1}$ belonging to $\mathbf{\Sigma}^0_\xi$.

As one can easily check from the definition, the classes $\mathcal{D}_\eta\big(\mathbf{\Sigma}^0_\xi\big)$ are monotone both in $\eta$ and in $\xi$.

**Fact 2.7.** For each $\eta \leq \eta'$ and $\xi \leq \xi'$ we have

$$\mathcal{D}_\eta\big(\mathbf{\Sigma}^0_\xi\big) \subseteq \mathcal{D}_{\eta'}\big(\mathbf{\Sigma}^0_{\xi'}\big).$$

The following theorem shows that the difference hierarchy over $\mathbf{\Sigma}^0_\xi$ saturates the successive class $\mathbf{\Delta}^0_{\xi+1}$.

**Theorem 2.8** (Hausdorff, Kuratowski, see [Kec95, Theorem 22.27, page 176])**.** In Polish spaces and for any $1 \leq \xi < \omega_1$ we have that

$$\mathbf{\Delta}^0_{\xi+1} = \bigcup_{1 \leq \theta < \omega_1} \mathcal{D}_\theta(\mathbf{\Sigma}^0_\xi).$$

Similarly, the first $\omega$ levels of the hierarchy coincide with the class $\mathsf{BC}(\mathbf{\Sigma}^0_\xi)$:

**Theorem 2.9** ([Kec95, Exercise 22.30, page 177])**.** In Polish spaces and for any $1 \leq \xi < \omega_1$ we have that

$$\mathsf{BC}(\mathbf{\Sigma}^0_\xi) = \bigcup_{1 \leq \theta < \omega} \mathcal{D}_\theta(\mathbf{\Sigma}^0_\xi).$$

2.5. **The Wadge hierarchy.** We are now in the position to define the Wadge hierarchy of a general topological space. Later in the article we will focus on the specific case of the Wadge hierarchy of the Cantor space $2^\omega$.

**Definition 2.10.** Let $X, Y$ be two topological spaces. We say that a set $A \subseteq X$ *continuously reduces* to $B \subseteq Y$ if there is a continuous function $f \colon X \to Y$ such that the pre-image $f^{-1}(B) = \{x \in X \mid f(x) \in B\}$ is equal to $A$ (i.e. $x \in A \Leftrightarrow f(x) \in B$ for every $x \in X$).

The following proposition shows that continuous reductions preserve the topological classes defined above.

**Proposition 2.11.** Let $\Gamma$ be a level of the Borel hierarchy or of the difference hierarchy. Then $\Gamma$ is closed under continuous preimages: if $B$ is a subset of a topological space $Y$ such that $B \in \Gamma(Y)$ and $f \colon X \to Y$ is a continuous function from a topological space $X$ to $Y$, then $f^{-1}(B) \in \Gamma(X)$.

Now we are in place to define the Wadge order.

**Definition 2.12.** Let $X$ and $Y$ be two topological spaces and let $A \subseteq X$ and $B \subseteq Y$. We say that $A$ is *Wadge reducible* to $B$, and we denote it by $A \leq_{\mathrm{W}} B$, if there exists a continuous reduction of $A$ to $B$. We say that $A$ is *Wadge equivalent* to $B$, in symbols $A \equiv_{\mathrm{W}} B$, if $A \leq_{\mathrm{W}} B$ and $B \leq_{\mathrm{W}} A$. Finally, we write $A <_{\mathrm{W}} B$ if $A \leq_{\mathrm{W}} B$ and $B \leq_{\mathrm{W}} A$ does not hold.

**Fact 2.13.** $\leq_{\mathrm{W}}$ is an equivalence relation.

The relation $\leq_{\mathrm{W}}$ induces a partial order between the $\equiv_{\mathrm{W}}$-classes, called *Wadge degrees*, of subsets of topological spaces. If we fix a space $X$ and we restrict the ordering induced by $\leq_{\mathrm{W}}$ to the sets of $X$, we obtain the *Wadge hierarchy* of $X$. If $A$ is a subset of $X$, then by $[A]_{\mathrm{W}}$ we denote its Wadge degree:

$$[A]_{\mathrm{W}} = \{B \subseteq X \mid B \equiv_{\mathrm{W}} A\}.$$

Even tough the Wadge hierarchy can be defined for any topological space, its shape for a generic space can be very complicated (for example the Wadge hierarchy of many non zero-dimensional topological spaces, including the space of real numbers, is very complicated: see for instance [MRS14] and [MRSS15]). Also, the good properties of the hierarchy (like its width or well-foundedness) depend on the determinacy of related games. Therefore, in our work we will restrict our attention to the order $\leq_{\mathrm{W}}$ restricted to the first levels of the Borel hierarchy of the Cantor space $2^\omega$. We refer the reader to [AC13] for a description of the structure of the Wadge hierarchy for $2^\omega$.

**Theorem 2.14** (Wadge's lemma, see [Kec95, Theorem 21.14, page 156]). For any $A, B \in \mathcal{B}(2^\omega)$ it holds that

$$A \leq_{\mathrm{W}} B \quad \text{or} \quad B^{\mathrm{c}} \leq_{\mathrm{W}} A.$$

**Theorem 2.15** (Wadge, Martin, Monk, see [Kec95, Theorem 21.15, page 158]). The ordering $\leq_{\mathrm{W}}$ among the Borel sets of $2^\omega$ is well-founded.

**Definition 2.16.** A set which is Wadge reducible to its complement is called *self-dual*, otherwise it is called *non self-dual*.

**Example 2.17.** It is easy to check that every non-trivial (i.e. different than $\emptyset$ and $2^\omega$) clopen subset of $2^\omega$ is self-dual.

Since the notion of self-duality is invariant under $\equiv_{\mathrm{W}}$ we can speak of self-dual and non self-dual Wadge degrees. If $[A]_{\mathrm{W}}$ is a non self-dual Wadge degree then we say that the pair $\{[A]_{\mathrm{W}}, [A^{\mathrm{c}}]_{\mathrm{W}}\}$ is a *non self-dual pair*.

**Corollary 2.18.** The anti-chains in the Wadge degrees have length at most 2 and are of the form

$$\big\{[A]_{\mathrm{W}}, [A^{\mathrm{c}}]_{\mathrm{W}}\big\},$$

with $A$ non self-dual.

Notice that technically every Wadge degree does not contain the elements contained in the previous degrees of the hierarchy. For example, the Wadge degree $[C]_{\mathrm{W}}$, where $C$ is a clopen set different from $2^\omega$ and $\emptyset$, contains all the clopen sets except the whole space $2^\omega$ and the empty set $\emptyset$. This is obvious, since any Wadge degree is an equivalence class of the relation $\equiv_{\mathrm{W}}$. We define the *Wadge class* of a set as the union of its Wadge degree with all its predecessors in the hierarchy. For example the Wadge class $\mathbf{\Delta}_1^0$ is obtained by taking the union of the Wadge degree $\mathbf{\Delta}_1^0 \setminus \{\emptyset, 2^\omega\}$ with its predecessors $\{\emptyset\}$ and $\{2^\omega\}$. It is clear that the two hierarchies, the one of Wadge degrees and the one of Wadge classes, are isomorphic as orders, so we can treat both the hierarchies in the same way. In the pictures of this section we show the hierarchy of the Wadge classes, because they are more intuitive and easier to describe.

**Theorem 2.19** (See [AC13]). In the Cantor space, $[2^\omega]_{\mathrm{W}} = \{2^\omega\}$ and $[\emptyset]_{\mathrm{W}} = \{\emptyset\}$ are the two minimal Wadge degrees and they clearly form a non self-dual pair. Then we have the Wadge degree formed by any clopen set different from $2^\omega$ and $\emptyset$ and this is a self-dual Wadge degree. The hierarchy continues with a constant alternation of a non self-dual pair and one self-dual Wadge degree. All the limit levels of the Wadge hierarchy consist of a non self-dual pair. Certain specific Wadge classes coincide with the levels of the Borel hierarchy.

Hence, the Wadge hierarchy of the Cantor space has the shape as depicted in Figure 1.

Now we can assign an ordinal to any level of the hierarchy. This ordinal is the *Wadge rank* of a Wadge degree (or of the corresponding Wadge class). The two bottom Wadge degrees $\{\emptyset\}$ and $\{2^\omega\}$ have Wadge rank 1, the Wadge degree $\mathbf{\Delta}_1^0 \setminus \{\emptyset, 2^\omega\}$ has Wadge rank 2 (so it has the Wadge class $\mathbf{\Delta}_1^0$), and so on.

Among the non self-dual Wadge classes we find the classes $\mathbf{\Sigma}_n^0$ and $\mathbf{\Pi}_n^0$. The classes $\mathbf{\Sigma}_1^0$ and $\mathbf{\Pi}_1^0$ are immediately after the Wadge class $\mathbf{\Delta}_1^0$, so their Wadge rank is 3. Then, between the non self-dual pair

$$\{\mathbf{\Sigma}_n^0, \mathbf{\Pi}_n^0\}$$
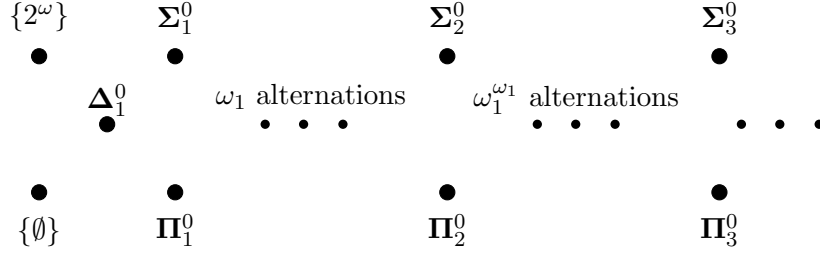
Figure 4: An initial fragment of the Wadge hierarchy inside the Borel hierarchy of $2^\omega$.

and the successive

$$\{\mathbf{\Sigma}^0_{n+1}, \mathbf{\Pi}^0_{n+1}\}$$

there are $\omega_1^{\omega_1^{\cdot^{\cdot^{\omega_1}}}} {\scriptstyle n \text{ times}}$ Wadge classes. In particular, there are $\omega_1$ Wadge classes between the pair $\{\mathbf{\Sigma}^0_1, \mathbf{\Pi}^0_1\}$ and $\{\mathbf{\Sigma}^0_2, \mathbf{\Pi}^0_2\}$. Hence, the Wadge rank of the Wadge classes $\mathbf{\Sigma}^0_2$ and $\mathbf{\Pi}^0_2$ is $\omega_1$, and the Borel class $\mathbf{\Delta}^0_2$ contains $\omega_1$ different levels of the Wadge hierarchy.

Now we focus on the segment that we will study in this article, that is the initial segment from the beginning of the hierarchy until the Wadge classes $\mathbf{\Sigma}^0_2$ and $\mathbf{\Pi}^0_2$. Using this notion we can express a characterisation of the first $\omega_1$ levels of the Wadge hierarchy in $2^\omega$ in terms of the difference hierarchy.

**Theorem 2.20** (See [AC13])**.** For every $m \geq 1$ every class $\mathcal{D}_\theta\big(\mathbf{\Sigma}^0_m\big)$ corresponds to a Wadge class of a non self-dual pair. For $m = 1$ these are essentially all Wadge classes: there is no non self-dual Wadge class between $\mathcal{D}_\theta\big(\mathbf{\Sigma}^0_1\big)$ and $\mathcal{D}_{\theta+1}\big(\mathbf{\Sigma}^0_1\big)$. Therefore, Figure 1 depicts the first $\omega_1$ levels of the Wadge hierarchy.

Hence, in the Cantor space the difference hierarchy is an important tool to understand the Wadge hierarchy, especially in the initial part up to $\mathbf{\Sigma}^0_2$. Beyond that level, the difference hierarchy becomes much coarser (i.e. it skips a lot of Wadge degrees).

## 3. The game for Wadge ranks

In this section we define a game that we will use in this article to obtain results of decidability of Wadge degrees with Wadge ranks up to $\omega$ (i.e. the classes $\mathcal{D}_n\big(\mathbf{\Sigma}^0_1\big)$ for $n \in \omega$). This game is played by two players, named Alternator and Constrainer and it is a finite duration game. In this article we work on the space $\mathrm{Tr}_A$, but *a priori* this game can be defined in any topological space and the characterization that it gives holds in general. Yet, in the case of regular languages of trees, it is possible to decide which player wins the game. This fact will be crucial to state the results about decidability.

Let us describe the game. Let $X$ be a topological space, $U_0 \subseteq X$ open and non-empty, and let $X_1, \ldots, X_n$ be arbitrary subsets of $X$. We define the game

$$\mathcal{H}_{U_0}(X_1, \ldots, X_n)$$

played by Constrainer (choosing open subsets of $X$) and Alternator (choosing points of $X$). The game will last for $n$ rounds, a round $i$ for $1, \ldots, n$ of the game is played as follows:

(1) Alternator chooses a point $x_i \in U_{i-1} \cap X_i$. If there is no such point $x_i$, the game is interrupted and Constrainer wins immediately.

(2) Constrainer chooses an open set $U_i \subseteq U_{i-1}$ that contains $x_i$ and the next round is played.

If Alternator manages to survive $n$ rounds then he wins, otherwise Constrainer wins.

A special variant of the game, when $U_0 = X$ is the whole space, is denoted simply as $\mathcal{H}(X_1, \ldots, X_n)$. Now we prove some properties of this game.

Let $(X, \tau)$ be a topological space, $U \subseteq X$ open non-empty, and let $X_1, \ldots, X_n$ be subsets of $X$. Consider the game $\mathcal{H}_U(X_1, \ldots, X_n)$. In this framework we can represent a position of a play trough a tuple

$$\langle U_0, x_1, U_1, x_2, U_2, \ldots, x_i, U_i \rangle,$$

where $x_1, \ldots, x_i \in X$ and $U_0, U_1, \ldots, U_i \in \tau$ with $U_0 = U$. A *strategy for Constrainer* in the game $\mathcal{H}_U(X_1, \ldots, X_n)$ is a function

$$\sigma \colon X^{\leq n} \to \tau.$$

If $s$ is a word belonging to $X^{\leq n}$ compatible with the game $\mathcal{H}$ and with $\sigma$, then $\sigma(s)$ is the open set played by Constrainer in the position

$$\langle s_0, \sigma(s_0), s_1, \sigma(s_0 s_1), \ldots, s_{lt(s)-1}, \sigma(s) \rangle.$$

If $s$ does not represent a position compatible with the game $\mathcal{H}$ and with $\sigma$ (for example because the second letter of $s$ is not an element of $X_2$ or it is not an element of $\sigma(s_0)$), then we put $\sigma(s) = \emptyset$ by convention. As usual, a strategy $\sigma$ for Constrainer is *winning* if Constrainer wins every play where he follows $\sigma$. In a specular way we could define strategies for Alternator, but we will not use them in this article.

Since the duration of the game is finite, it is determined — one of the players has a winning strategy. The following remark is not used in this article but provides a simplified form for the strategies of the players.

**Remark 3.1.** The game $\mathcal{H}_U(X_1, \ldots, X_n)$ is positionally determined: the winner of the game does not change if we insist that the players' strategies depend only on the number of the round and the last move of the opponent. This means that we can freely assume that Alternators's point $x_i$ depends only on $i$ and the current open set $U_{i-1}$; while Constrainer's open set $U_i$ depends only on $i$, the current point $x_i$, and the previous open set $U_{i-1}$.

The first property we prove is Refinement Lemma, that states that if the sets $X_1, \ldots, X_n$ are split into finitely many parts each and Alternator wins $\mathcal{H}(X_1, \ldots, X_n)$ then he can win for some choice of parts of $X_1, \ldots, X_n$.

**Lemma 3.2** (Refinement Lemma). Let $X_1, \ldots, X_n$ be subsets of a topological space $X$. For $i \in \{1, \ldots, n\}$, let $\mathcal{Y}_i$ a finite family of sets partitioning $X_i$. For any non-empty open $U \subseteq X$, if Alternator wins

$$\mathcal{H}_U(X_1, \ldots, X_n)$$

then there exist $Y_1 \in \mathcal{Y}_1, \ldots, Y_n \in \mathcal{Y}_n$ such that Alternator wins

$$\mathcal{H}_U(Y_1, \ldots, Y_n).$$

*Proof.* We prove the theorem by induction on $n$. The induction base is immediate, because Alternator always wins when $n = 0$ and he wins when $n = 1$ if and only if $X_1 \neq \emptyset$. Now prove the induction step. Consider the first move by Alternator, and assume that he chooses the same point $x \in U$ as chosen by his winning strategy in $\mathcal{H}_U(X_1, \ldots, X_n)$. This point

necessarily belongs to some $Y_1 \in \mathcal{Y}_1$. For $i \in \omega$, let $U_i$ be the open ball around $x$ of radius $\frac{1}{i}$. By the definition of the game, we know that for every $i < \omega$ Alternator wins

$$\mathcal{H}_{U_i}(X_1, \ldots, X_n).$$

Thus, he also wins $\mathcal{H}_{U_i}(X_2, \ldots, X_n)$ and by the inductive assumption, we know that for every $i$ there exist $Y_2^{(i)} \in \mathcal{Y}_2, \ldots, Y_n^{(i)} \in \mathcal{Y}_n$ such that Alternator wins

$$\mathcal{H}_{U_i}(Y_2^{(i)}, \ldots, Y_n^{(i)}).$$

By Pigeon-hole Principle, there must be some $Y_2, \ldots, Y_n$ such that

$$(Y_2, \ldots, Y_n) = (Y_2^{(i)}, \ldots, Y_n^{(i)})$$

holds for infinitely many $i$. Since the game $\mathcal{H}_V(Y_2, \ldots, Y_n)$ grows more difficult for Alternator as the open set $V$ becomes smaller, and since every open set $V$ that contains $x$ contains some $U_i$, we conclude that Alternator wins

$$\mathcal{H}_V(Y_2, \ldots, Y_n)$$

for every $V$ that contains $x$. By viewing $V$ as a response of Constrainer to Alternator's move $x \in Y_1$, we conclude that Alternator wins the game

$$\mathcal{H}_U(Y_1, \ldots, Y_n).$$

$\square$

Now let $Y$ be a subset of our topological space $X$ and consider a particular case of the game where the sets $X_1, \ldots, X_n$ alternate between $Y$ and its complement, i.e. we consider $\mathcal{H}(X_1, \ldots, X_n)$ where $X_i$ is $Y$ if $i$ is odd, $Y^c$ otherwise. We denote by $\mathcal{H}^{\in, \notin}(Y, n)$ that game and by $\mathcal{H}_U^{\in, \notin}(Y, n)$ the variant relativised to a non-empty open set $U \subseteq X$.

**Example 3.3.** Consider the game where the topological space $X$ is the space of real numbers $\mathbb{R}$ and $Y = \mathbb{Q}$, i.e. the rational numbers. Then for every $n$, Alternator wins the game $\mathcal{H}^{\in, \notin}(Y, n)$.

**Remark 3.4.** Notice that if Alternator wants to survive in $\mathcal{H}^{\in, \notin}(Y, n)$ as long as possible, he has to avoid to play points in the interior[1] of $Y$ and the interior of $Y^c$. For example, if at the first round Alternator plays $x$ belonging to the interior of $Y$ then Constrainer can play (a subset of) the interior of $Y$ and Alternator loses because he cannot go outside $Y$ any more.

**Example 3.5.** In the real numbers $\mathbb{R}$, let $Y$ be the complement of $\{\frac{1}{n} \in \mathbb{R} \mid n \in \omega\}$. Alternator wins $\mathcal{H}^{\in, \notin}(Y, 3)$. Indeed, in the first round Alternator can play $0 \in Y$. In the second round, Alternator plays $\frac{1}{n} \notin Y$ for some large $n$ depending on Constrainer's move. In the third round, Alternator plays $\frac{1}{n} + \epsilon \in Y$, for some small $\epsilon$ depending on Constrainer's move.

We now argue that Constrainer wins $\mathcal{H}^{\in, \notin}(Y, n)$ for $n \geq 4$. Notice that since $x_2$ must belong to $Y^c = \{\frac{1}{n} \in \mathbb{R} \mid n \in \omega\}$, Constrainer can choose $U_2 \subseteq U_1$ in such a way to guarantee that $x_2 \in U_2$ but $0 \notin U_2$. Then Alternator chooses a point $x_3 \in Y \cap U_2$ that must be distinct from 0. Thus, there exists an open set $U_3$ such that $x_3 \in U_3 \subseteq U_2$ but $U_3 \subseteq Y$. This guarantees that Alternator is not able to choose $x_4 \in Y^c \cap U_3$, making him lose the game $\mathcal{H}^{\in, \notin}(Y, 4)$.

---

[1]Recall that the interior of a set $Y$ is the union of all open sets contained in $Y$.

**Remark 3.6.** Let $\sigma_1$ and $\sigma_2$ be two strategies for Constrainer such that $\sigma_1(s) \subseteq \sigma_2(s)$ for any finite word $s \in X^{\leq n}$. If $\sigma_2$ is winning for Constrainer then $\sigma_1$ is winning for Constrainer too.

**Lemma 3.7.** Choose some basis $B$ for the topology of the topological space $X$. If Constrainer has a winning strategy in $\mathcal{H}^{\in,\notin}(Y, n)$ then he has a winning strategy which uses only basic open sets from $B$.

*Proof.* Consider any function $f$ that to every pair $(U, x)$, where $U$ is an open set and $x \in U$, assigns a basic open set $V \in B$ such that $x \in V$ and $V \subseteq U$. Let $\sigma$ be a winning strategy for Constrainer. We can define another winning strategy $\bar{\sigma}$ that always takes sets from $B$: given a finite word $s$ of length $i$, we define $\bar{\sigma}(s) = f\big(\sigma(s), s_i\big)$. Since $\sigma$ was winning for Constrainer, by Remark 3.6 also $\bar{\sigma}$ is winning. $\square$

Now we are ready to give a characterization of $\mathcal{D}_n\big(\boldsymbol{\Sigma}_1^0\big)$ sets in terms of the game $\mathcal{H}$.

**Theorem 3.8.** Let $X$ be a topological space and let $Y \subseteq X$. The following conditions are equivalent:
(1) $Y$ belongs to $\mathcal{D}_n\big(\boldsymbol{\Sigma}_1^0\big)$.
(2) Constrainer wins the game $\mathcal{H}^{\in,\notin}(Y, n+1)$.

*Proof.* We have to prove both directions separately.

**Implication** $1 \Rightarrow 2$. Suppose that $n$ is odd and let

$$Y = A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{n-1} \setminus A_{n-2}), \tag{3.1}$$

where $A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_{n-2} \subseteq A_{n-1}$ and every $A_i$ is open for $0 \leq i \leq n-1$. Then, a winning strategy for Constrainer in $\mathcal{H}^{\in,\notin}(Y, n)$ is to play $A_{n-1}$ as the first move, $A_{n-2}$ as the second move, and so on. Equation (3.1) implies that this is a valid strategy of Constrainer. The $n$th move will be $A_0$, and since $A_0 \subseteq Y$, at that point Alternator loses at the $(n+1)$th round. The case of $n$ even is completely dual, in that case $A_0 \subseteq Y^c$.

**Implication** $2 \Rightarrow 1$. Suppose that $n$ is odd, the opposite case can be solved by an entirely dual argument. Our aim is to present $Y$ as in (3.1). For $i = 0, 1, 2, \ldots$ define sets

$$A_i \overset{\text{def}}{=} \bigcup \big\{ U \mid \text{Constrainer wins } \mathcal{H}_U^{\in,\notin}(Y, i+1) \big\} \qquad \text{for odd } i$$

$$A_i \overset{\text{def}}{=} \bigcup \big\{ U \mid \text{Constrainer wins } \mathcal{H}_U^{\in,\notin}(Y^c, i+1) \big\} \qquad \text{for even } i$$

where the unions range over open sets $U \subseteq X$. Notice that if Constrainer wins $\mathcal{H}_U^{\in,\notin}(Y, i)$ then he also wins $\mathcal{H}_U^{\in,\notin}(Y^c, i+1)$ by the same strategy, just playing $U$ in the first round. Therefore, the family $(A_i)$ is increasing. By the definition, all the sets $A_i$ are open. Clearly, the assumption that Constrainer has a winning strategy in $\mathcal{H}^{\in,\notin}(Y, n+1)$ implies that $A_n$ is the whole space. Thus, it is enough to inductively prove the following claim.

**Claim 3.9.** For $i = 0, 1, \ldots$ the following holds

$$Y \cap A_i = A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{i-1} \setminus A_{i-2}) \qquad \text{for odd } i$$

$$Y^c \cap A_i = (A_1 \setminus A_0) \cup (A_3 \setminus A_2) \cup \ldots \cup (A_{i-1} \setminus A_{i-2}) \qquad \text{for even } i$$

Notice that by the definition $A_0 \subseteq Y$ — whenever there exists $x \in U \cap Y^c$ then Alternator wins $\mathcal{H}_U^{\in,\notin}(Y^c, 1)$ by playing $x$. Therefore, the above claim holds for $i = 0$ as $Y^c \cap A_0 = \emptyset$.

Assume that Claim 3.9 holds for $i-1$ and consider the two cases for $i$.

**The case of odd $i$.** In that case we need to prove that $Y \cap A_i$ is of the form from (3.1). Consider a point $x \in A_i$. First consider the case that $x \in A_{i-1}$. Then by the inductive assumption $x \in Y$ if and only if

$$x \notin (A_1 \setminus A_0) \cup (A_3 \setminus A_2) \cup \ldots \cup (A_{i-2} \setminus A_{i-3})$$

what is equivalent to the disjunction of $x \in (A_{i-1} \setminus A_{i-2})$ or

$$x \in A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{i-3} \setminus A_{i-4}).$$

Thus, $x \in Y$ if and only if

$$x \in A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{i-1} \setminus A_{i-2}).$$

Thus, the statement of Claim 3.9 holds in that case.

Now assume that $x \notin A_{i-1}$. We will prove that in that case $x \notin Y$. Assume contrarily that $x \in Y$. Since $x \in A_i$, there exists an open set $U$ such that $x \in U \subseteq A_i$ and Constrainer wins $\mathcal{H}_U^{\in,\notin}(Y, i+1)$. Consider the first round of $\mathcal{H}_U^{\in,\notin}(Y, i+1)$ in which Alternator plays $x$ and a winning strategy of Constrainer replies with $V \ni x$. This means that Constrainer has a winning strategy in $\mathcal{H}_V^{\in,\notin}(Y^c, i)$ and therefore by the definition $x \in V \subseteq A_{i-1}$, a contradiction.

**The case of even $i$.** This case is entirely dual: we take $x \in A_i$ and consider the case that $x \in A_{i-1}$. Then the following conditions are equivalent:

$$x \notin Y$$
$$x \notin A_0 \cup (A_2 \setminus A_1) \cup \ldots \cup (A_{i-2} \setminus A_{i-3})$$
$$x \in (A_1 \setminus A_0) \cup (A_3 \setminus A_2) \cup \ldots \cup (A_{i-3} \setminus A_{i-4}) \text{ or } x \in (A_{i-1} \setminus A_{i-2})$$
$$x \in (A_1 \setminus A_0) \cup (A_3 \setminus A_2) \cup \ldots \cup (A_{i-1} \setminus A_{i-2})$$

and therefore Claim 3.9 holds in that case. If $x \notin A_{i-1}$ then we need to prove that $x \in Y$. Since $x \in A_i$, we know that $x \in U \subseteq A_i$ with Constrainer winning $\mathcal{H}_U^{\in,\notin}(Y^c, i+1)$ for some open $U$. Assume contrarily that $x \notin Y$ and as before we see a contradiction, as $x$ is a valid move of Alternator in $\mathcal{H}_U^{\in,\notin}(Y^c, i+1)$ and therefore $x \in A_{i-1}$. $\qquad\square$

**Corollary 3.10.** The following conditions are equivalent for a set $Y$:
(1) $Y$ belongs to $\bigcup_{n \in \omega} \mathcal{D}_n(\mathbf{\Sigma}_1^0) = \mathsf{BC}(\mathbf{\Sigma}_1^0)$.
(2) Constrainer wins the game $\mathcal{H}^{\in,\notin}(Y, n)$ for all but finitely many $n$.

*Proof.* It follows directly from Theorem 3.8 and Fact 2.7. $\qquad\square$

Since the family of sets defined by prefixes $N_p$ for all finite prefixes $p$ is a basis of the topology on $\mathrm{Tr}_A$, we obtain the following corollary for the case $X = \mathrm{Tr}_A$.

**Corollary 3.11.** Assume that $X = \mathrm{Tr}_A$ is the space of all trees and $L \subseteq \mathrm{Tr}_A$. Then, when considering strategies of Constrainer in $\mathcal{H}^{\in, \notin}(L, n)$ we can assume that each open set $U_i$ played by him is a basic open set, i.e. $U_{i+1} = N_p$ for a finite prefix $p$ of the currently played tree $t_i$. Then the condition that $U_{i+2} \subseteq U_{i+1}$ boils down to the assumption that $p_{i+2} \supseteq p_{i+1}$.

3.1. **Decidability of finite levels of the Wadge hierarchy.** In this short subsection we use the game $\mathcal{H}(L_1, \ldots, L_n)$ to show that, given a natural number $n$, it is decidable if a regular language $L$ is an $n$-difference of open sets (i.e. belongs to $\mathcal{D}_n(\boldsymbol{\Sigma}_1^0)$). Recall that, without loss of generality (see Corollary 3.11) we can assume that Constrainer plays finite prefixes of the trees played by Alternator, and Alternator has to extend the finite prefixes played by Constrainer.

**Example 3.12.** Consider the language

$$L = \{t \in \mathrm{Tr}_A \mid \text{infinitely many letters } a \text{ appear in } t\}.$$

$L$ is regular and it is easy to check that Alternator wins the game $\mathcal{H}^{\in, \notin}(L, n)$ for every $n \in \omega$. This is because every finite prefix can be extended to a tree with finitely many $a$ or to a tree with infinitely many $a$. So $L$ is not a Boolean combination of open sets (it is easy to see, indeed, that $L$ is a $\boldsymbol{\Pi}_2^0$ set but it is not a $\boldsymbol{\Sigma}_2^0$ set).

**Lemma 3.13.** Given regular tree languages $L_1, \ldots, L_n$, one can decide who wins the game $\mathcal{H}(L_1, \ldots, L_n)$. In particular, given $L$ and $n$, one can decide who wins $\mathcal{H}^{\in, \notin}(L, n)$.

*Proof.* We prove the statement for two regular languages $L_1, L_2$. It is easy to generalise it to $n$ regular languages. The sentences "Alternator wins the game $\mathcal{H}(L_1, L_2)$" and "Constrainer wins the game $\mathcal{H}(L_1, L_2)$" can be effectively formalized in Monadic Second-order Logic on the complete binary tree. For instance, the sentence for

"Alternator wins the game $\mathcal{H}(L_1, L_2)$"

is:

there exists a tree $t_1 \in L_1$ such that for any finite prefix $p$ of $t_1$

there exists a tree $t_2 \in L_2$ that extends $p$.

In a similar way we can write the sentence that says that Constrainer wins for $n > 2$. $\square$

So we obtain:

**Corollary 3.14.** It is decidable, given a regular tree language $L$ and $n \in \omega$, whether $L$ is an $n$-difference of open sets.

*Proof.* It directly follows from Theorem 3.8 and Lemma 3.13. $\square$

Hence, since Wadge degrees with Wadge ranks below $\omega$ are formed by Boolean combinations of the levels of the difference hierarchy, we easily obtain:

**Theorem 3.15.** Given a regular language $L$ and a Wadge degree $[A]_W$ with Wadge rank less than $\omega$, it is decidable if $L$ belongs to $[A]_W$.

3.2. **The infinite variant of the game.** We denote by $\mathcal{H}^\infty(Y)$ the infinite variant of $\mathcal{H}^{\in,\notin}(Y,n)$: $\mathcal{H}^\infty(Y)$ is the infinite duration game played the same way as $\mathcal{H}^{\in,\notin}(Y,n)$ but the winning condition for Alternator is that he has to survive for infinitely many turns. By $\mathcal{H}^\infty_U(Y)$ we denote the relativised game.

**Remark 3.16.** The game $\mathcal{H}^\infty(Y)$ is determined: every play where Constrainer wins is finite. Therefore, the winning condition for Constrainer is an open condition, while the winning condition for Alternator is a closed condition, both in the space[2] $\left(X \times \tau\right)^\omega$, where the sets $X$ and $\tau$ are taken with the discrete topology. Hence, the game is determined by Gale–Stewart Theorem [GS53] (see also [Mar75] for the more general result for Borel sets).

**Remark 3.17.** In the same fashion as in the case of the finite game, without loss of generality we can assume that Constrainer in his strategies uses only basic open sets, see Corollary 3.11.

The following fact follows directly from the definition of the two variants of the game.

**Fact 3.18.** If Alternator wins $\mathcal{H}^\infty(Y)$ then he wins $\mathcal{H}^{\in,\notin}(Y,n)$ for any $n$.

**Proposition 3.19.** The converse to Fact 3.18 is not true, even for regular tree languages $L$.

This proposition follows a posteriori from Theorem 5.1 and Proposition 3.20, using the fact that the involved games are determined and there exist sets in $\mathbf{\Delta}^0_2 \setminus \mathcal{BC}(\mathbf{\Sigma}^0_1)$. However, the proof provided here is supposed to provide an informative illustration of the considered games.

*Proof.* We have to exhibit a counterexample. To do that it is convenient to work with an alphabet with three different symbols, so let $A$ be the alphabet $\{a,b,c\}$. For the sake of this example, assume that if $t_1,\ldots,t_n$ are trees, by $[t_1,\ldots,t_n]$ we denote the tree
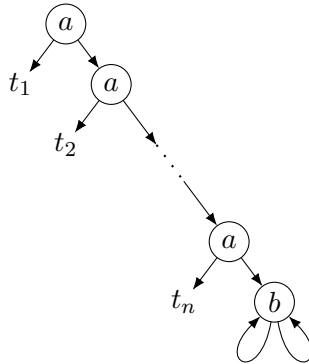


---

Now let $L$ be the language of all the trees of the form $[t_1, \ldots, t_n]$ where for every $i \in \{1, \ldots, n\}$ the tree $t_i$ is

(1) either a tree with every node labelled by $a$,

(2) or a tree that contains only finitely many letters different than $c$, i.e. a tree for which there exists a finite prefix $p$ of $t_i$ such that $t_i(u) = c$ for any node $u \notin p$.

If $t_i$ respects the first condition we say that $t_i$ is a *first case tree*, if it respects the second condition we say it is a *second case tree*. We first prove that Alternator wins $\mathcal{H}^{\in, \notin}(L, n)$ for any $n$. Fix a natural number $n$, we provide a winning strategy for Alternator for the game $\mathcal{H}^{\in, \notin}(L, 2n)$. We define the following sets of trees:

- For $i \in \{1, \ldots, n\}$ let $L_i$ be the set of trees $[t_1, \ldots, t_n]$ such that the trees $t_1, \ldots, t_{i-1}$ are second case trees and the trees $t_i, \ldots, t_n$ are first case trees. It is clear that $L_i \subseteq L$ for any $i$.

- We define $L_i'$ as $L_i$ except that the tree $t_i$ contains both $a$ and $b$ nodes, but no $c$ nodes. Obviously $L_i'$ is disjoint from $L$.

Every prefix of a tree in $L_i$ can be completed into a tree in $L_i'$ and every prefix of a tree in $L_i'$ can be completed into a tree in $L_{i+1}$. It follows that Alternator wins the game

$$\mathcal{H}(L_1, L_1', L_2, L_2', \ldots, L_n, L_n')$$

and therefore also Alternator wins $\mathcal{H}^{\in, \notin}(L, 2n)$.

Now we move to a proof that Alternator loses $\mathcal{H}^{\infty}(L)$. Consider the tree played by Alternator in the first round. Since this tree belongs to $L$, it must be of the form $[t_1, \ldots, t_n]$ with $t_1, \ldots, t_n$ either first case trees or second case trees. Let $p_1$ be a finite prefix of this tree which contains the node $\mathtt{r}^n$. Constrainer uses a strategy, which preserves the following properties:

(1) All prefixes played by Constrainer extend the prefix $p_1$. Consequently, all the trees played by Alternator are of the form $[s_1, \ldots, s_n]$. Indeed, the prefix $p_1$ guarantees that the played trees $t$ satisfy $t(\mathtt{r}^n) = b$ and $t(\mathtt{r}^k) = a$ for $k < n$. Hence, all the modifications done by Alternator are relative to the trees $t_1, \ldots, t_n$ and therefore for $k \in \{1, \ldots, n\}$ it is meaningful to talk about the $k$th coordinate of the tree played by Alternator in a round which refers to the tree $s_k$.

(2) Suppose that Alternator plays a tree $[s_1, \ldots, s_n]$ in some round $i$. Let $p_i$ be a finite prefix of this tree such that for every coordinate $k \in \{1, \ldots, n\}$ we have:
   - If $s_k$ is a second case tree then $p_i$ contains a prefix of $s_k$ such that under that prefix every node is labelled by $c$.
   - If $s_k$ contains some $b$ then $p_i$ contains some $b$ in the subtree $s_k$.

   In the next round Constrainer chooses $p_i$. Consequently, if $i, j$ are rounds with $i < j$ and $k \in \{1, \ldots, n\}$ then
   - If the $k$th coordinate of Alternator's tree in round $i$ is a second case tree then also the $k$th coordinate of Alternator's tree in round $j$ has to be a second case tree.
   - If the $k$th coordinate of Alternator's tree in round $i$ contains a $b$ then also the $k$th coordinate of Alternator's tree in round $j$ contains a $b$.

   So, in an odd-numbered round, Alternator's tree belongs to the language and therefore all the coordinates with a $b$ are second case trees. In an even-numbered round, Alternator's tree is outside the language. Therefore, when going from an odd-numbered round to the next even-numbered round, Alternator must change some coordinate from a first case tree without $b$ to a tree with $b$. It follows that the number coordinates

with $b$ increases in each even-numbered round. Since this can happen at most $n$ times, Alternator must lose after at most $2n$ rounds.

The proof is complete. $\qquad\square$

Now we can give a non-effective characterisation of the class $\mathbf{\Delta}_2^0$ for topological spaces that are completely metrizable (for the levels $\mathcal{D}_n(\mathbf{\Sigma}_1^0)$ we gave a characterisation that holds in general for any topological space, but here we are forced to require complete metrizability).

**Proposition 3.20.** Let $X$ be a completely metrizable topological space with a countable basis of the topology (i.e. $X$ is *second-countable*). The following conditions are equivalent for a subset $Y$ of $X$:

(1) Constrainer wins $\mathcal{H}^\infty(Y)$.
(2) $Y \in \mathbf{\Delta}_2^0(X)$.

*Proof.* The proof is very similar to the analysis of other games of this kind, for instance Banach–Mazur game, see [Kec95, Section 8.H].

**Implication** $1 \Rightarrow 2$. Assume that Constrainer has a winning strategy $\sigma$ in $\mathcal{H}^\infty(Y)$. By Remark 3.17 we can assume that $\sigma$ plays only basic open sets.

Notice that $\sigma$ (seen as a tree) is well-founded because the strategy is winning and therefore it admits no infinite play. We will prove by induction on the structure of $\sigma$ that if $\langle U_0, x_1, U_1, \ldots, U_{i-1} \rangle$ is a position compatible with $\sigma$ then $Y \cap U_{i-1} \in \mathbf{\Delta}_2^0$. Consider such a position $P = \langle U_0, x_1, U_1, \ldots, U_{i-1} \rangle$ and assume that the thesis holds for all the positions extending that one. If the position $P$ is instantly winning for Constrainer (i.e. Alternator cannot play a single round from $P$) then, depending on parity of $i$, either $U_{i-1} \subseteq Y^c$ or $U_{i-1} \subseteq Y$. In both cases the inductive thesis holds. Now assume that $P$ is not instantly winning for Constrainer. By the symmetry lets assume that $i$ is odd, i.e. Alternator is forced to play $x_i \in U_{i-1} \cap Y$. Let $(B_x)_{x \in U_{i-1} \cap Y}$ be the indexed family of basic open sets $B_x$ played by $\sigma$ as a response to Alternator playing $x$. By the inductive assumption we know that for each $x \in U_{i-1} \cap Y$ we have $Y \cap B_x \in \mathbf{\Delta}_2^0$.

By the definition of the family $B_x$ we know that

$$Y \cap U_{i-1} = \bigcup_{x \in U_{i-1} \cap Y} (B_x \cap Y),$$

where the union is in fact countable since there is only countably many basic open sets in $X$. As every set taken in the union is $\mathbf{\Sigma}_2^0$ (in fact $\mathbf{\Delta}_2^0$) we know that $Y \cap U_{i-1}$ is $\mathbf{\Sigma}_2^0$. Dually

$$Y^c \cap U_{i-1} = \left( U_{i-1} \setminus \bigcup_{x \in U_{i-1} \cap Y} B_x \right) \cup \bigcup_{x \in U_{i-1} \cap Y} (B_x \cap Y^c),$$

which again is a $\mathbf{\Sigma}_2^0$ presentation of $Y^c \cap U_{i-1}$.

Thus, the above induction implies that $Y \cap U_0 = Y \cap X = Y$ is $\mathbf{\Delta}_2^0$.

**Implication** $2 \Rightarrow 1$**.** We need to prove that if $Y \in \mathbf{\Delta}_2^0$ then Constrainer wins $\mathcal{H}^\infty(Y)$. Indeed, if $Y$ is in $\mathbf{\Delta}_2^0$ we can write $Y$ and its complement as

$$Y = \bigcap_{j \in \omega} A_j \quad \text{and} \quad Y^{\mathrm{c}} = \bigcap_{j \in \omega} B_j,$$

where all the sets $A_i$ and $B_i$ are open. Now we can describe a winning strategy for Constrainer in $\mathcal{H}^\infty(Y)$. Suppose $i = 1, 2, \ldots$ is the round we are playing and $i$ is odd (resp. $i$ is even). Let $j = \lfloor \frac{i-1}{2} \rfloor$. Assume that $U_{i-1}$ is the open set that was played last ($U_0 = X$) and let $x_i$ be the point played by Alternator in the current round. By the definition of the game, if $i$ is odd then $x_i \in Y$ and otherwise $x_i \in Y^{\mathrm{c}}$. Let Constrainer play $U_i$ such that $\overline{U_i} \subseteq U_{i-1}$; $U_i \subseteq A_j$ (resp. $U_i \subseteq B_j$); and the diameter of $U_i$ is smaller than $2^{-i}$. Such a set exists because $x_i \in U_{i-1} \cap A_j$ (resp. $x_i \in U_{i-1} \cap B_j$).

Clearly it is a valid strategy of Constrainer. Consider an infinite play consistent with this strategy. Since $X$ is Polish and the sets $U_i$ are of decreasing diameter with $\overline{U_i} \subseteq U_{i-1}$, there must exists $x \in \bigcap_{n \in \omega} U_i$. But by the construction of $U_i$, such $x$ must belong to both $\bigcap_{j \in \omega} A_j = Y$ and $\bigcap_{j \in \omega} B_j = Y^{\mathrm{c}}$, a contradiction. Thus, each play consistent with the above strategy is finite and therefore winning for Constrainer. $\qquad\square$

**Corollary 3.21.** Let $X$ be a completely metrizable topological space and $Y$ be a subset of $X$. Then $Y \in \mathbf{\Delta}_2^0 \setminus \mathsf{BC}(\mathbf{\Sigma}_1^0)$ if and only if Alternator wins $\mathcal{H}^{\in, \notin}(Y, n)$ for all $n$ but he loses $\mathcal{H}^\infty(Y)$.

## 4. The algebra on trees

Our next goal is decidability of the class $\bigcup_{n \in \omega} \mathcal{D}_n(\mathbf{\Sigma}_1^0) = \mathsf{BC}(\mathbf{\Sigma}_1^0)$. Notice that Corollary 3.14 gives us a semi-algorithm for deciding if a regular language is in $\bigcup_{n \in \omega} \mathcal{D}_n(\mathbf{\Sigma}_1^0)$. Indeed, for $n = 1, 2, \ldots$ we can use Corollary 3.14 to decide if $L \in \mathcal{D}_n(\mathbf{\Sigma}_1^0)$. If for some $n$ this is the case then $L \in \bigcup_{n \in \omega} \mathcal{D}_n(\mathbf{\Sigma}_1^0)$ and the algorithm terminates. Otherwise, the algorithm does not terminate. Section 5 provides an alternative algebraic algorithm that always terminates and solves the above decision problem.

The algebraic approach we define in this section is based on the so-called *Myhill–Nerode equivalence* that allows to distinguish trees based on their behaviour when put into certain *contexts*.

**Definition 4.1.** A *multicontext* over an alphabet $A$ is a partial tree $t$ over $A \sqcup \{\square\}$ where $\square \notin A$ such that: $t$ does not contain any unary nodes and a node of $t$ is a leaf if and only if it is labelled $\square$. A *port* of a multicontext $C$ is any node of $t$ that is labelled $\square$ (i.e. any leaf of $t$).

The number of ports is called the *arity* of the multicontext. *A priori* a multicontext may have infinitely many ports and in that case the arity is $\infty$. A multicontext with exactly one port is called a *context*. Given a multicontext $C$ and a valuation $\eta$ which maps ports of $C$ to trees in $\mathrm{Tr}_A$, we write $C[\eta]$ for the tree obtained by replacing each port $u$ by the tree $\eta(u)$. The tree $C[\eta]$ is said to *extend* the multicontext $C$. If $L$ is a set of trees and $C$ is a multicontext then by $C[L]$ we denote the set of trees obtained by plugging trees of $L$ in the ports of $C$ in all the possible ways. The set of all trees extending a multicontext $C$ is denoted by $C[*]$. If $C$ is a multicontext, possibly with infinitely many ports, and $t$ is a tree, we denote by $C[t]$ the tree obtained by putting $t$ in every port of $C$.

**Example 4.2.** The multicontext $C_0$ consists of only one node — the root. It is called the *trivial context* and denoted $\Box$. $C_1$ is a tree, it has no ports, and $C_1[*]$ is $\{C_1\}$. The multicontext $C_2$ is a context and if we complete $C_2$ with a tree we obtain a tree where the root label is $a$ and the left subtree of the root is labelled with only letters $b$. Finally, $C_3$ is a finite multicontext and $C_3[*] = \{t \mid t(\epsilon) = a\}$.



$$C_0 \qquad\qquad C_1 \qquad\qquad C_2 \qquad\qquad C_3$$
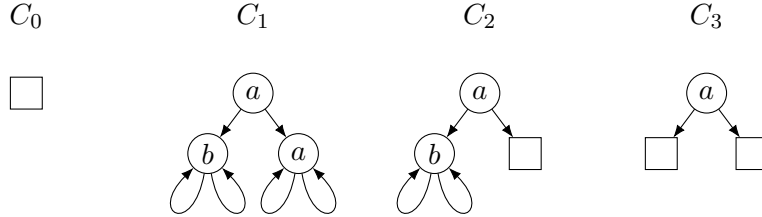
Now we focus on contexts, i.e. multicontexts with exactly one port. We write $\mathsf{Ct}_A$ for the set of all non-trivial contexts over $A$. Given two contexts $C, D$ we write $C \cdot D$ for the context obtained by replacing the port of $C$ with $D$. Moreover, if $C \neq \Box$ (i.e. $C$ is non-trivial) then we write $C^\infty$ for the infinite tree

$$C \cdot C \cdot C \cdot C \cdots$$

Notice that the above definition does not construct a tree for $C = \Box$ (the trivial context). That is why we restrict the set of contexts $\mathsf{Ct}_A$ to only non-trivial ones.

**Remark 4.3.** It is easy to verify that $\cdot$ is *associative*, therefore $\big(\mathsf{Ct}_A \sqcup \{\Box\}, \cdot\big)$ is an infinite monoid, with $\cdot$ associative and the trivial context $\Box$ as the neutral element.

Now we define the two Myhill–Nerode equivalence relations: one for trees and one for contexts. These equivalence relations depend on a fixed language $L \subseteq \mathrm{Tr}_A$.

**Definition 4.4.** In the Myhill–Nerode equivalence for trees, we say that two trees $t$ and $t'$ are *L-equivalent* (denoted $t \overset{\mathrm{Tr}}{\approx}_L t'$) if

$$C[t] \in L \Longleftrightarrow C[t'] \in L \quad \text{ for every multicontext } C.$$

**Remark 4.5.** Notice that because of the duality between $L$ and the complement $L^c$ in the above definition, we know that the equivalence relations $\overset{\mathrm{Tr}}{\approx}_L$ and $\overset{\mathrm{Tr}}{\approx}_{L^c}$ coincide. Moreover, since $\Box$ (i.e. the trivial context) is a multicontext we know that if $t \in L$ and $t' \in L^c$ then $t \overset{\mathrm{Tr}}{\not\approx}_L t'$.

The above remark says that the equivalence relation $\overset{\mathrm{Tr}}{\approx}_L$ is always able to distinguish trees from $L$ from those not in $L$. In the following example these are the only two classes of $\overset{\mathrm{Tr}}{\approx}_L$, i.e. $\overset{\mathrm{Tr}}{\approx}_L$ is not able to distinguish anything else.

**Example 4.6.** Consider the language $L = \{t \in \mathrm{Tr}_A \mid t(\epsilon) = a\}$. In this case we have just two equivalence classes that are $L$ and the complement $L^c$. The multicontext that establishes if a tree belongs to $L$ or $L^c$ is the trivial context $\Box$.

To give a similar definition for contexts, we use a variant of multicontexts where the ports can be substituted by contexts and not trees.

**Definition 4.7.** A *context environment* over an alphabet $A$ is a partial tree labelled by $A \sqcup \{\Box\}$ such that: $t$ has no leaves and a node of $t$ is unary if and only if it is labelled $\Box$. A *port* of a context environment is any node of $t$ that is labelled $\Box$ (i.e. any unary node of $t$).

Given a context environment $E$ and a non-trivial context $C$, we write $E[C]$ for the tree obtained by substituting $C$ for every port of $E$ in the following way. For each port $v$ of $E$ we first plug a copy of $C$ into that port and then plug the subtree of $E$ below that port into the port of $C$. Notice that there may be more than one occurrence of $\Box$ on a branch of $E$ and the above plugging needs to be performed in each of them, see the left-most branch of $E[C]$ in Example 4.8. This guarantees that $E[C]$ is a tree — it does not contain any port of $E$ nor any of the copies of the port of $C$.

The assumption that $C$ is non-trivial is important in the above construction, because if $E$ is a unary tree labelled everywhere $\Box$ (i.e. a $\Box$-labelled infinite path) and $C = \Box$ is the trivial context, then $E[C]$ (if defined at all) does not contain any node.

**Example 4.8.** In the following figure, $C$ is a context, $E$ is a context environment, and $E[C]$ is a tree.
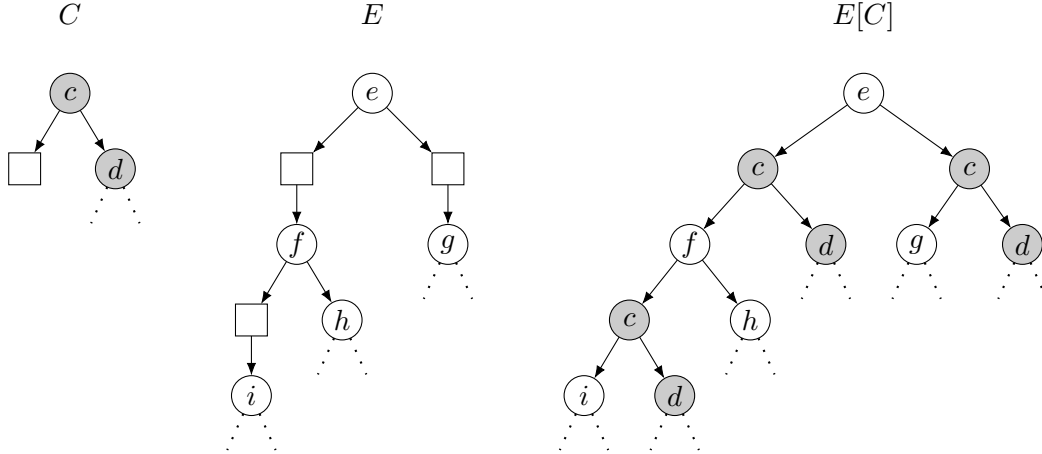


**Definition 4.9.** We define two non-trivial contexts $C$ and $C'$ to be *L-equivalent* (denoted $C \overset{\mathrm{Ct}}{\approx}_L C'$) if

$$E[C] \in L \Longleftrightarrow E[C'] \in L \quad \text{for every context environment } E.$$

4.1. **The syntactic tree algebra $(H_L, V_L)$.** We denote by $H_L$ the set of the equivalence classes of trees with respect to $L$, the elements of $H_L$ are called *tree types*. Similarly we denote by $V_L$ the set of the equivalence classes of non-trivial contexts with respect to $L$, the elements of $V_L$ are called *context types*. By $1_L$ we denote an additional type corresponding to the trivial context $\Box$. Since $V_L$ denotes types of the non-trivial contexts, in general $1_L \notin V_L$.

We will now prove a number of results, showing that the sets $H_L$ and $V_L$ bear certain algebraic structure. We start with a simple fact following directly from the definition of $L$-equivalence.

**Fact 4.10.** For every multicontext $D$ and context environment $E$, the following two operations preserve the $L$-equivalence:

(1) the operation $t \to D[t]$, i.e. if $t \overset{\mathrm{Tr}}{\approx}_L t'$ then $D[t] \overset{\mathrm{Tr}}{\approx}_L D[t']$,
(2) the operation $C \to E[C]$, i.e. if $C \overset{\mathrm{Ct}}{\approx}_L C'$ then $E[C] \overset{\mathrm{Tr}}{\approx}_L E[C']$.

The following corollary follows directly from Fact 4.10.

**Corollary 4.11.** Given a multicontext $D$, a context environment $E$, and elements $h \in H_L$, $v \in V_L$, the sets

$$D[h] \overset{\text{def}}{=} \{D[t] \mid t \in h\} \subseteq \mathrm{Tr}_A,$$

$$E[v] \overset{\text{def}}{=} \{E[C] \mid C \in v\} \subseteq \mathrm{Tr}_A.$$

are $L$-equivalence classes, i.e. elements of $H_L$.

Thus, each multicontext $D$ (resp. context environment $E$) induces an operation $H_L \to H_L$ (resp. $V_L \to H_L$).

As expressed by the next lemma, the following natural operations on contexts and trees respect the $L$-equivalence.

**Lemma 4.12.** The following operations respect $L$-equivalence.
(1) The composition of contexts $(C_1, C_2) \to C_1 \cdot C_2$.
(2) Substituting a tree in the port of a context $(C, t) \to C[t]$.
(3) Infinite iteration of a non-trivial context $C \to C^\infty$.
(4) For every symbol $c \in A$, the operations $t \mapsto c(t, \square)$ and $t \mapsto c(\square, t)$, that produce new contexts with roots labelled $c$, and $t$ plugged as the left or the right subtree under the root, respectively.

*Proof.* We prove all the items.

**Item 1.** Let $C_1$, $C_2$, $D_1$, $D_2$ be contexts such that $C_1$ is $L$-equivalent with $D_1$ and $C_2$ is $L$-equivalent with $D_2$. Let $E$ be some context environment, we prove that

$$E[C_1 \cdot C_2] \in L \Leftrightarrow E[D_1 \cdot D_2] \in L.$$

We construct from $E$, $C_1$ and $E$, $D_2$ respectively two new context environments $E_C$ and $E_D$ such that

$$E_C[C_2] = E[C_1 \cdot C_2] \text{ and } E_D[D_1] = E[D_1 \cdot D_2].$$

By the symmetry, lets focus on $E_C$. This context environment is obtained from $E$ by inserting into each occurrence of $\square$ a whole copy of $C_1$ (including its own $\square$). Thus, the overall number of holes of $E_C$ and $E$ is the same and the above equality follows.

Using the $L$-equivalence, we have:

$$E_D[C_1] \in L \quad \Leftrightarrow \quad E_D[D_1] \in L$$
$$E_C[C_2] \in L \quad \Leftrightarrow \quad E_C[D_2] \in L$$

Note that by the definition of $E_C$ and $E_D$ we have $E_C[D_2] = E_D[C_1]$. It follows that

$$E[C_1 \cdot C_2] \in L \Leftrightarrow E_C[C_2] \in L \Leftrightarrow E_C[D_2] \in L \Leftrightarrow E_D[C_1] \in L \Leftrightarrow E_D[D_1] \in L.$$

Therefore, $C_1 \cdot C_2$ and $D_1 \cdot D_2$ are $L$-equivalent.

**Item 2.** Let $C$, $C'$ be $L$-equivalent contexts and $t$, $t'$ be $L$-equivalent trees. We want to prove that $C[t]$ and $C'[t']$ are two $L$-equivalent trees. Let $D$ be a generic multicontext. We prove that

$$D[C[t]] \in L \Leftrightarrow D[C'[t']] \in L.$$

Let $D'$ be a multicontext such that $D'[t] = D[C[t]]$ and $E$ a context environment such that $E[C'] = D[C'[t']]$. Using the $L$-equivalence we have:

$$D'[t] \in L \quad \Leftrightarrow \quad D'[t'] \in L,$$
$$E[C] \in L \quad \Leftrightarrow \quad E[C'] \in L.$$

By the definition of $E$ and $D'$ we have $D'[t'] = E[C]$. It follows that

$$D[C[t]] \in L \Leftrightarrow D'[t] \in L \Leftrightarrow D'[t'] \in L \Leftrightarrow E[C] \in L \Leftrightarrow E[C'] \in L \Leftrightarrow D[C'[t']] \in L.$$

Therefore, $C[t]$ and $C'[t']$ are $L$-equivalent.

**Item 3.** Let $C$, $C'$ be $L$-equivalent non-trivial contexts. We want to prove that $C^\infty$ and $C'^\infty$ are two $L$-equivalent trees. Let $D$ be some multicontext, we prove that

$$D[C^\infty] \in L \Leftrightarrow D[C'^\infty] \in L.$$

Consider a context environment $E$ constructed from $D$ by replacing each port of $D$ with an infinite chain of ports. Using $L$-equivalence of $C$ and $C'$ we get:

$$E[C] \in L \Leftrightarrow E[C'] \in L$$
By definition of $E$ we have $E[C] = D[C^\infty]$ and $E[C'] = D[C'^\infty]$. It follows that
$$D[C^\infty] \in L \Leftrightarrow D[C'^\infty] \in L.$$

Therefore, $C^\infty$ and $C'^\infty$ are $L$-equivalent.

**Item 4.** We only do the proof for $t \mapsto c(t, \square)$, the other operations is handled symmetrically. Let $t, t'$ be $L$-equivalent trees. Let $E$ be some context environment, we prove that

$$E[c(t, \square)] \in L \Leftrightarrow E[c(t', \square)] \in L.$$

By inserting $c$ into the ports of $E$ we construct a multicontext $C$ such that for all trees $s$, $C[s] = E[c(s, \square)]$. Using $L$-equivalence of $t$ and $t'$ we get:

$$C[t] \in L \Leftrightarrow C[t'] \in L$$

Therefore, $c(t, \square)$ and $c(t', \square)$ are $L$-equivalent. $\qquad\square$

**Corollary 4.13.** The above operations on $\mathrm{Tr}_A$ and $\mathsf{Ct}_A$ induce the following algebraic structure on $(H_L, V_L)$:

- composition of context types $V_L \ni u, v \longmapsto u \cdot v \in V_L$,
- action of $V_L$ on $H_L$ i.e. $V_L \times H_L \ni (u, h) \longmapsto u \cdot h \in H_L$,
- infinite composition $V_L \ni u \longmapsto u^\infty \in H_L$,
- creation of contexts $H_L \ni h \longmapsto c(\square, h), c(h, \square) \in V_L$ for $c \in A$.

Moreover, $(V_L, \cdot)$ is a semigroup acting over $H_L$ via $\cdot$, $(V_L \sqcup \{1_L\}, \cdot)$ is a monoid, and $(H_L, V_L)$ satisfy the axioms of a Wilke algebra [Wil93].

A pair of sets $(H, V)$, equipped with the operations as in Corollaries 4.11 and 4.13, is called a *tree algebra*. Once restricted to the operations from Corollary 4.13, it becomes a *thin algebra*[3]. An advantage of a thin algebra is that (given that $H$ and $V$ are finite) it can be represented effectively on a computer by giving the sets and the multiplication tables for the five involved operations. Because of the simplicity of the considered classes of topological complexity, the operations involved in our characterisations involve only operations of thin algebras, see Equations (5.1) and (5.2).

---

[3]The name comes from the theory of *thin trees*, sometimes called *scattered trees*, see [ISB16].

Notice that the pair $(\mathrm{Tr}_A, \mathsf{Ct}_A)$ with the actual operations of composition of trees and contexts is in fact a tree algebra, we call it the *free* tree algebra.

The *syntactic morphism* of a language $L$, denoted by $\alpha_L$, is the two-sorted function

$$\alpha_L \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to (H_L, V_L),$$

that maps a tree $t \in \mathrm{Tr}_A$ into its $\overset{\mathrm{Tr}}{\approx}_L$-equivalence class $\alpha_L(t) \in H_L$ and a context $C \in \mathsf{Ct}_A$ into its $\overset{\mathrm{Ct}}{\approx}_L$-equivalence class $\alpha_L(C) \in V_L$.

The next fact is considered folklore, see for instance [PP04, Proposition 1.11 in Annex A on page 442]. The usual notation for the number $\sharp$ is $\omega$, however it is better not to use that symbol in this paper to avoid confusion with the infinite repetition.

**Fact 4.14.** Given any finite semigroup $V$, there is a number $\sharp_V$ (denoted just $\sharp$ if $V$ is known from the context) such that for each element $v$ of $V$ the element $v^\sharp$ is an idempotent, i.e. $v^\sharp = v^\sharp \cdot v^\sharp$.

4.2. **Congruences.** To be able to compute the syntactic tree algebra $(H_L, V_L)$ we will need to start with its approximation. This notion is formalised as follows.

**Definition 4.15.** Consider a two-sorted function $\alpha \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to (H, V)$ that is surjective onto a pair of sets $(H, V)$. We say that it is a *congruence* if the following two conditions hold:

- for every multicontext $D$ and every pair of trees $t, t' \in \mathrm{Tr}_A$ such that $\alpha(t) = \alpha(t')$ we have
$$\alpha\big(D[t]\big) = \alpha\big(D[t']\big),$$

- for every context environment $E$ and every pair of contexts $C, C' \in \mathsf{Ct}_A$ such that $\alpha(C) = \alpha(C')$ we have
$$\alpha\big(E[C]\big) = \alpha\big(E[C']\big).$$

We say that $\alpha$ *recognises* a set of trees $L \subseteq \mathrm{Tr}_A$ if $L = \alpha^{-1}(S)$ for some subset $S \subseteq H$.

**Remark 4.16.** If $\alpha$ is a congruence that recognises $L$ then its kernel is finer than the $L$-equivalence in the sense that for each pair of trees $t$, $t'$ and pair of contexts $C$, $C'$ we have:

$$\alpha(t) = \alpha(t') \Rightarrow t \overset{\mathrm{Tr}}{\approx}_L t' \quad \text{and} \quad \alpha(C) = \alpha(C') \Rightarrow C \overset{\mathrm{Ct}}{\approx}_L C'.$$

*Proof.* Assume that $S \subseteq H$ is such that $\alpha^{-1}(S) = L$. Consider the first claim and take two trees $t, t' \in \mathrm{Tr}_A$ such that $\alpha(t) = \alpha(t')$. We need to prove that $t \overset{\mathrm{Tr}}{\approx}_L t'$. Take a multicontext $D$ and assume by the symmetry that $D[t] \in L$, i.e. $\alpha\big(D[t]\big) \in S$. However, the assumption that $\alpha$ is a congruence implies that $\alpha\big(D[t']\big) = \alpha\big(D[t]\big)$ and therefore $D[t'] \in \alpha^{-1}(S) = L$. $\square$

Corollary 4.11 implies that the syntactic morphism $\alpha_L \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to (H_L, V_L)$ is a congruence. Lemma 4.12 implies that $\alpha_L$ is a homomorphism of thin algebras, i.e. it commutes with the operations of these algebras — for instance for the operation $v \cdot v'$ one needs to observe that for every two contexts $C, C' \in \mathsf{Ct}_A$ we have

$$\alpha_L(C) \cdot \alpha_L(C') = \alpha_L\big(C \cdot C'\big).$$

Additionally, Remark 4.5 implies that $\alpha_L$ recognises $L$.

4.3. **Computing** $(H_L, V_L)$**.** We will now prove that the syntactic algebras can be constructed effectively, as stated by the following proposition.

**Proposition 4.17.** If $L$ is a regular language then both $H_L$ and $V_L$ are finite. Moreover, given a parity non-deterministic automaton $\mathcal{A}$ recognising $L$, one can compute in EXPTIME the syntactic algebra $(H_L, V_L)$ for $L$, together with the structure of thin algebra on $(H_L, V_L)$. Additionally, both $H_L$ and $V_L$ have at most exponentially many elements in the number of states of $\mathcal{A}$.

The following remark shows that finiteness of $H_L$ and $V_L$ is not sufficient for regularity (the remark is motivated by Running Example 2 in [Boj15a]).

**Remark 4.18.** Both $H_L$ and $V_L$ are finite for any language defined in MSO+U (see [Boj04]).

The rest of this subsection is devoted to a proof of Proposition 4.17. We follow implicitly the approach from [Boj15a], see [Boj15b, Theorem 3.1] in the full version of the paper. A very similar construction is also given in [BI09b, Appendices D.1 and D.2] or [Idz12].

The proof is divided into two stages. The first stage shows how to construct some congruence $\alpha \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to (H, V)$ recognising $L$, such that the sets $H$ and $V$ are at most exponential in the number of states of $\mathcal{A}$ — a non-deterministic automaton recognising $L$. The second stage merges certain elements of that congruence to obtain the syntactic algebra $(H_L, V_L)$.

**The first stage — a congruence $\alpha$.** We assume the notions of parity automata over infinite trees as in [Tho96]: such an automaton is a tuple $\mathcal{A} = \langle A, Q, q_\mathrm{I}, \delta, \Omega \rangle$, where $A$ is an alphabet, $Q$ is a set of states, $q_\mathrm{I} \in Q$ is an initial state, $\delta \subseteq Q \times A \times Q^2$ is a transition relation and $\Omega \colon Q \to \{i, \dots, j\}$ is a priority assignment. We use the standard concepts of an accepting run of a given automaton over a given tree.

Define a two-sorted function $\alpha \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to \left( 2^Q, 2^{Q \times \{i, \dots, j\} \times Q} \right)$ as follows:

$$\alpha(t) \stackrel{\mathrm{def}}{=} \{q \in Q \mid \mathcal{A} \text{ has an accepting run over } t \text{ from } q\},$$

$$\alpha(C) \stackrel{\mathrm{def}}{=} \{(q, \ell, q') \in Q \times \{i, \dots, j\} \times Q \mid$$
$$\mathcal{A} \text{ has an accepting run over } C \text{ that:}$$
$$\text{starts from } q \text{ in the root of } C,$$
$$\text{reaches } q' \text{ in the port of } C,$$
$$\text{and the maximal priority seen in that run}$$
$$\text{on the path to from the root to the port equals } \ell\}.$$

Let $(H, V)$ be the range of $\alpha$. Intuitively, the value $\alpha(t)$ says from which states the automaton $\mathcal{A}$ can accept a given tree. In particular, $t \in L$ if and only if $q_\mathrm{I} \in \alpha(t)$ (thus, $\alpha$ recognises $L$). Similarly, the value $\alpha(C)$ for a context $C$ contains information about the relation between the states $\mathcal{A}$ can have in the root and in the port of $C$. However, as our algebra needs to deal with the operation $C^\infty$, we additionally need to know what is the maximal priority on the considered path, to make sure that it satisfies the parity condition once repeated infinitely many times.

**Claim 4.19.** $\alpha$ is a congruence.

*Proof.* We start with the first bullet of the definition (the second bullet speaking about contexts is analogous). Consider two trees $t, t' \in \mathrm{Tr}_A$ such that $\alpha(t) = \alpha(t') = h \in H$ and let $D$ be a multicontext. Our aim is to prove that $\alpha(D[t]) = \alpha(D[t'])$. Assume by the symmetry that $q \in \alpha(D[t])$, i.e. the tree $D[t]$ can be accepted from a state $q \in Q$. We need to prove that $q \in \alpha(D[t'])$ (the symmetric case is analogous). There exists a run of $\mathcal{A}$ over $D[t]$ that is accepting and starts from the state $q$ in the root of $D[t]$. By the assumption that $\alpha(t) = \alpha(t')$ we can construct a new run of $\mathcal{A}$ over $D[t']$ that is also accepting and starts from $q$, it is enough to replace the runs over $t$ by the respective runs over $t'$, below each of the ports of $D$. $\qquad\square$

We will now define explicitly the operations of thin algebra on $(H, V)$, for $h \in H_{\mathcal{A}}$, $v, v' \in V_{\mathcal{A}}$, and $c \in A$:

$$v \cdot v' \stackrel{\text{def}}{=} \{(q, \max(\ell, \ell'), q'') \mid (q, \ell, q') \in v, (q', \ell', q'') \in v'\},$$

$$v \cdot h \stackrel{\text{def}}{=} \{q \in Q \mid (q, \ell, q') \in v, q' \in h\},$$

$$v^{\infty} \stackrel{\text{def}}{=} (v^{\sharp})^{\infty} \qquad\qquad\qquad\qquad \text{see Fact 4.14,}$$

$$e^{\infty} \stackrel{\text{def}}{=} \{q \in Q \mid (q, \ell, q') \in e, (q', 2\ell', q') \in e\} \qquad \text{for } e \in V \text{ idempotent,}$$

$$c(\square, h) \stackrel{\text{def}}{=} \{(q, \Omega(q), q_{\mathsf{L}}) \mid (q, c, q_{\mathsf{L}}, q_{\mathsf{R}}) \in \delta, q_{\mathsf{R}} \in h\},$$

$$c(h, \square) \stackrel{\text{def}}{=} \{(q, \Omega(q), q_{\mathsf{R}}) \mid (q, c, q_{\mathsf{L}}, q_{\mathsf{R}}) \in \delta, q_{\mathsf{L}} \in h\}.$$

It is relatively easy to check that the above operations are compatible with $\alpha$, i.e. $\alpha$ is a homomorphism of thin algebras. Thus, we have constructed a congruence $\alpha \colon (\mathrm{Tr}_A, \mathsf{Ct}_A) \to (H, V)$ that recognises $L$ and additionally we have given explicitly a structure of thin algebra on $(H, V)$.

**The second stage — a quotient of $(H, V)$.** We now provide a sketch of the second stage of the construction: computing the equivalence relation $\approx$ on $(H, V)$ defined as:

$$t \stackrel{\mathrm{Tr}}{\approx}_L t' \Leftrightarrow \alpha(t) \approx \alpha(t') \quad \text{and} \quad C \stackrel{\mathrm{Ct}}{\approx}_L C' \Leftrightarrow \alpha(C) \approx \alpha(C'). \qquad (4.1)$$

First notice that this is a correct definition that does not depend on the choice of witnesses because of Remark 4.16.

For the sake of simplicity we will focus on the case of trees, i.e. we need to compute in EXPTIME if $h \approx h'$. However, Equation (4.1) says that $h \not\approx h'$ if and only if there exists a multicontext $D$ such that $D[h] \in L \Leftrightarrow D[h'] \notin L$ (the question whether $D[h] \in L$ is well-posed because $\alpha$ is a congruence recognising $L$).

This goal is achieved by constructing a non-deterministic parity tree automaton $\mathcal{C}_h$ of size polynomial in $\mathcal{A}$, that recognises the language of those multicontexts $D$ such that $D[h] \in L$. The automaton $\mathcal{C}_h$ non-deterministically guesses an accepting run of $\mathcal{A}$ over $D[t]$ for some hypothetical tree $t$. When it reaches a port of $D$, it verifies if the current state $q$ of $\mathcal{A}$ belongs to $h$. Once these automata are constructed, the question whether $h \approx h'$ boils down to checking if $\mathrm{L}(\mathcal{C}_h) = \mathrm{L}(\mathcal{C}_{h'})$, which can be done in EXPTIME.

Now, that the equivalence relation $\approx$ is computed, it is enough to divide $(H, V)$ by it. Because of (4.1), there is a bijection between the quotient $(H, V)/\approx$ and the syntactic algebra $(H_L, V_L)$. Moreover, since both $\alpha$ and $\alpha_L$ are homomorphisms of thin algebras, the thin algebra operations in $(H, V)$ must preserve $\approx$. This means that we can obtain the

structure of a thin algebra on $(H,V)/\approx$ that corresponds to the thin algebra structure of $(H_L, V_L)$. This concludes the proof of Proposition 4.17.

4.4. **Quotients.** Similarly as in the case of finite words, the syntactic algebra induces a natural notion of a quotient of a language: given a multicontext $D$ with $n$ holes and a language $K$ of trees, by $D^{-1}(K)$ we denote the set of tuples $(t_1, \ldots, t_n)$ such that the valuation $\eta$ mapping the $i$th port of $D$ into $t_i$ satisfies

$$D[\eta] \in K.$$

Notice that if $D$ is a context then in fact $D^{-1}(K)$ is a set of 1-tuples of trees, which we identify with trees, i.e. $D^{-1}(K) \subseteq \mathrm{Tr}_A$.

If $L$ is a language recognised by a morphism $\alpha_L$ then the fact whether $D[t] \in L$ depends only on $\alpha_L(D) \in V_L \sqcup \{1_L\}$ and $\alpha_L(t) \in H_L$. Therefore, it makes sense to write $v^{-1}(L)$ for $v \in V_L \sqcup \{1_L\}$. Also in that case $v^{-1}(L)$ is a language recognised by the morphism $\alpha_L$. Directly from the definition we get that

$$(vu)^{-1}(L) = u^{-1}\big(v^{-1}(L)\big). \tag{4.2}$$

4.5. **The game on types.** Now we want to extend the definition of the game $\mathcal{H}$ to sets of contexts. Recall that contexts are defined as a special case of partial trees, with an additional port label that appears in exactly one leaf. Notice that if $p$ is a finite partial context (i.e. $p$ has exactly one leaf labelled $\square$) then every context in the basic open set $N_p$ must have the same port as $p$ — in such a case we say that $p$ *fixes the port*. Also, if $U$ is an open set of contexts and $C \in U$ then there exists a sufficiently big prefix $p \subset C$ such that $C \in N_p \subseteq U$ and $p$ fixes the port. Thus, without loss of generality we can assume that we consider only those basic open sets $N_p$ where $p$ does fix the port. This means that every two contexts $C, C' \in N_p$ must have the same position of the port.

This yields the definition of a game $\mathcal{V}(K_1, \ldots, K_n)$ for a sequence $K_1, \ldots, K_n$ of context languages (i.e. subsets of $\mathrm{Ct}_A \sqcup \{\square\}$), which is played by Alternator and Constrainer. The game is played in $n$ rounds. Round $i = 1$ is special: Alternator chooses a context $C_1 \in K_1$. Let $u$ be the port of the context $C_1$. This port will stay fixed for the rest of the game; all contexts produced by Alternator will have their port in the node $u$. Next, Constrainer chooses a finite prefix $D_1$ of $C_1$, which has one of its leaves in the node $u$ (i.e. fixes the port).

A subsequent round $i \in \{2, \ldots, n\}$ is played as follows. Let $D_{i-1}$ be the finite partial tree over $A \sqcup \{\square\}$ chosen by Constrainer in the previous round with a leaf in the node $u$.
- Alternator provides a context $C_i$, which extends $D_{i-1}$, belongs to $K_i$, and has its port in the node $u$. If there is no such context, the game is interrupted and Constrainer wins immediately.
- Constrainer chooses a finite prefix $D_i$ of $C_i$ and which has a leaf in the node $u$.

If Alternator manages to survive $n$ rounds then he wins. Recall that by the definition of the syntactic morphism, a tree type $h \in H_L$ is actually equal to the set of trees $\alpha_L^{-1}(h)$, similarly for a context type $v \in V_L$. Therefore, it makes sense to talk about the games $\mathcal{H}(h_1, \ldots, h_n)$, and $\mathcal{V}(v_1, \ldots, v_n)$ for sequences of types. Using these games we define the following sets of sequences of types:

**Definition 4.20.** We define two sets:
$$\mathcal{H}_L = \{(h_1, \ldots, h_n) \in (H_L)^* \mid \text{ Alternator wins } \mathcal{H}(h_1, \ldots, h_n)\},$$
$$\mathcal{V}_L = \{(v_1, \ldots, v_n) \in (V_L)^* \mid \text{ Alternator wins } \mathcal{V}(v_1, \ldots, v_n)\}.$$

A comment on notation is in order here. The sets $\mathcal{H}_L$ and $\mathcal{V}_L$ contain words, over alphabets $H_L$ and $V_L$, respectively. Usually when dealing with words, one omits the brackets and commas and writes $abc$ instead of $(a, b, c)$. When the alphabet is $V_L$ this leads to ambiguity, since the expression $vwu$ can be interpreted as a word with a single letter obtained by multiplying the three context types $v$, $w$, and $u$, or a three-letter word over the alphabet $V_L$. These two interpretations should not be confused, so we write $(v_1, \ldots, v_n)$ for $n$-letter words over the alphabet $V_L$. For the sake of uniformity, we also write $(h_1, \ldots, h_n)$ for $n$-letter words over the alphabet $H_L$, although there is no risk of ambiguity here.

It turns out that the sets of words $\mathcal{H}_L$ and $\mathcal{V}_L$ have specific structure. We say that a word $u$ is a *subword* of $w$ if $u$ can be obtained from $w$ by removing some letters.

**Fact 4.21.** Both sets $\mathcal{H}_L$ and $\mathcal{V}_L$ are closed under removing letters, i.e. if $u$ is a subword of $w$ and $w \in \mathcal{H}_L$ then also $u \in \mathcal{H}_L$ (similarly for $\mathcal{V}_L$).

*Proof.* It is clear from the definitions of the games $\mathcal{H}$ and $\mathcal{V}$. □

The order induced by the subword relation (known also as the Higman's order) has the following finiteness property.

**Lemma 4.22** (Higman's Lemma, see [Hig52]). The set of finite words $A^*$ over a finite alphabet $A$ with the subword ordering is a well-quasi order: there is no infinite antichain nor an infinite descending chain.

**Fact 4.23.** If $L \subseteq A^*$ is closed under removing letters then $L$ is regular.

*Proof.* Consider $L \subseteq A^*$ that is closed under removing letters. Then $L$ forms a downward-closed set with respect to the subword relation. Thus, Higman's Lemma implies that there exists a finite set of words $w_1, \ldots, w_N$ such that $w$ does not belong to $L$ if and only if one of $w_1, \ldots, w_N$ is a subword of $w$. Such a condition is a regular condition. □

**Corollary 4.24.** Both $\mathcal{H}_L$ and $\mathcal{V}_L$ are regular languages of finite words.

*Proof.* Both languages are closed under removing letters and therefore Fact 4.23 applies. □

The above corollary is amusing, but useless for our needs, because it does not say how to compute automata for $\mathcal{H}_L$ and $\mathcal{V}_L$ as a function of a representation of the language $L$.

**Lemma 4.25.** The following properties hold for each context $C$ and context environment $E$:
(1) $(h_1, \ldots, h_n) \in \mathcal{H}_L$ implies $(C[h_1], \ldots, C[h_n]) \in \mathcal{H}_L$.
(2) $(v_1, \ldots, v_n) \in \mathcal{V}_L$ implies $(E[v_1], \ldots, E[v_n]) \in \mathcal{H}_L$.
(3) $(v_1, \ldots, v_n), (w_1, \ldots, w_n) \in \mathcal{V}_L$ implies $(v_1 w_1, \ldots, v_n w_n) \in \mathcal{V}_L$.
(4) $(v_1, \ldots, v_n) \in \mathcal{V}_L, (h_1, \ldots, h_n) \in \mathcal{H}_L$ implies $(v_1 h_1, \ldots, v_n h_n) \in \mathcal{H}_L$.
(5) $(v_1, \ldots, v_n) \in \mathcal{V}_L$ implies $(v_1^\infty, \ldots, v_n^\infty) \in \mathcal{H}_L$.
(6) $(h_1, \ldots, h_n) \in \mathcal{H}_L$ implies $(c[\square, h_1], \ldots, c[\square, h_n]) \in \mathcal{V}_L$.
(7) $(h_1, \ldots, h_n) \in \mathcal{H}_L$ implies $(c[h_1, \square], \ldots, c[h_n, \square]) \in \mathcal{V}_L$.

*Proof.* All properties are proved by composing strategies, we prove the first one. All other properties are proved similarly. Assume that $(h_1, \ldots, h_n) \in \mathcal{H}_L$, and consider some multi-context $C$ (possibly with infinitely many ports). For all $i \le n$ let $L_i$ be the set of trees

that are Alternator's first move in some winning strategy for $\mathcal{H}(h_i, \ldots, h_n)$. Note that since $(h_1, \ldots, h_n) \in \mathcal{H}_L$, $L_i$ is non-empty for all $i$.

Lemma 4.25 follows directly from the following claim.

**Claim 4.26.** For all $i \leq n$, for all trees $t$ obtained by plugging trees of $L_i$ in the ports of $C$, Alternator has a winning strategy in $\mathcal{H}(C[h_i], \ldots, C[h_n])$ such that the tree chosen in round 1 is $t$.

*Proof.* We proceed by induction on $i$. For $i = n$ this is obvious. Assume the result holds for $i$ and we prove it for $i-1$. Let $p$ be a prefix of $t$, for all subtree $s$ plugged into a port of $C$, $p$ yields some (possibly empty) prefix $p_s$ of $s$. Since $s \in L_{i-1}$, $p_s$ can be completed into a tree $s' \in L_i$. It follows that $p$ can be completed into $t'$ obtained by plugging trees of $L_i$ in the ports of $C$. By induction hypothesis, Alternator has winning strategy in $\mathcal{H}(C[h_i], \ldots, C[h_n])$ such the tree chosen in round 1 is $t'$. Finally we conclude that Alternator has winning strategy in $\mathcal{H}(C[h_{i-1}], \ldots, C[h_n])$ such the tree chosen in round 1 is $t$. ☐

☐

**Definition 4.27.** We define the *alternation* of a finite word to be the length of the word obtained by iteratively eliminating letters that are identical to their predecessors. We say that a set of words has *unbounded alternation* if it contains words with arbitrarily large alternation. In the other case we say that a set has *bounded alternation*. A word is *alternating* if every two consecutive letters are distinct.

For example the alternation of *abccbbb* is 4 and *abcb* is the alternating word witnessing that. The notion of *alternation* gives us another characterization of the class $\mathsf{BC}(\mathbf{\Sigma}_1^0)$.

**Lemma 4.28.** For a regular language $L$ of infinite trees, the following conditions are equivalent:
(1) Alternator wins the game $\mathcal{H}^{\in, \notin}(L, n)$ for infinitely many $n$.
(2) The set $\mathcal{H}_L$ has unbounded alternation.

Clearly, if $n \leq n'$ and Constrainer wins the game $\mathcal{H}^{\in, \notin}(L, n)$ then he also wins the game $\mathcal{H}^{\in, \notin}(L, n')$. This means that in fact there are only two possibilities: either Alternator wins $\mathcal{H}^{\in, \notin}(L, n)$ for all $n$; or Constrainer wins $\mathcal{H}^{\in, \notin}(L, n)$ for all except finitely many $n$. Thus, the conditions of Lemma 4.28 are in fact equivalent to saying that Alternator wins the game $\mathcal{H}^{\in, \notin}(L, n)$ for all $n$.

*Proof.* We have to prove both implications.
$1 \Rightarrow 2$. We show that for $n \in \omega$ if Alternator wins the game $\mathcal{H}^{\in, \notin}(L, n)$, then $\mathcal{H}_L$ contains an alternating word of length $n$. Suppose that Alternator wins $\mathcal{H}^{\in, \notin}(L, n)$. Both $L$ and $L^c$ can be partitioned into tree types, see Remark 4.5. By Lemma 3.2, Alternator wins $\mathcal{H}(h_1, \ldots, h_n)$ for some sequence of types, such that $h_i$ is included in $L$ or its complement, depending on the parity of $i$. In particular, the consecutive types are different.
$2 \Rightarrow 1$. Suppose that $\mathcal{H}_L$ has unbounded alternation. By Fact 4.21 we know that $\mathcal{H}_L$ is closed under removing letters. We will now argue that there must be some $g, h \in H_L$ with $g \neq h$ such that $\mathcal{H}_L$ contains all the words

$$(g, h), (g, h, g, h), (g, h, g, h, g, h), \ldots$$

Assume contrarily and use finiteness of $H_L$: there must exist a global bound $B$ on the number of times any pair of distinct types $g \neq h$ can appear in a word in $\mathcal{H}_L$ in the alternating way as above. Consider an alternating word of length $|H_L|^2 \cdot (B+1) + 1$ in $\mathcal{H}_L$. This word contains $|H_L|^2 \cdot (B+1)$ pairs of consecutive distinct letters and therefore some pair of those must appear there at least $B+1$ times — a contradiction.

Since $g$ and $h$ are different elements of the syntactic algebra, it follows that there must be some multicontext $C$ such that the tree type $C[g]$ is contained in $L$, while the tree type $C[h]$ is disjoint with $L$. By the first item of Lemma 4.25, we can conclude that $\mathcal{H}_L$ contains all the words

$$\big(C[g], C[h]\big), \big(C[g], C[h], C[g], C[h]\big), \big(C[g], C[h], C[g], C[h], C[g], C[h]\big), \ldots$$

It follows that Alternator can alternate arbitrarily long between the language $L$ and its complement.

$\square$

**Lemma 4.29.** If $\mathcal{V}_L$ has unbounded alternation then so does $\mathcal{H}_L$.

*Proof.* Assume that $\mathcal{V}_L$ has unbounded alternation. Take $n > 0$, we will find a sequence in $\mathcal{H}_L$ of alternation at least $n$. Let $N \stackrel{\text{def}}{=} 2 \cdot n \cdot |V_L|^2$ and let $(v_1, \ldots, v_N)$ be an alternating sequence in $\mathcal{V}_L$ of length $N$ — such a sequence exists by the assumption and Fact 4.21. By Pigeon-hole Principle, there exist two types $v \neq v' \in V_L$ such that the word $(v, v')^n$ can be obtained from $(v_1, \ldots, v_N)$ by removing letters. By the definition of $V_L$ (see Definition 4.9) we know that there exists a context environment $E$ such that $E[v] \neq E[v']$. Thus, by Item 2 of Lemma 4.25 we know that the word $\big(E[v], E[v']\big)^n$ is a member of $\mathcal{H}_L$. Since this word is alternating and has length $2n$, the claim holds. $\square$

## 5. Effective characterisation of $\mathsf{BC}(\mathbf{\Sigma}_1^0)$

In this section we state the crucial result of the paper, providing an effective characterisation of the class of regular tree languages in $\mathsf{BC}(\mathbf{\Sigma}_1^0)$.

**Theorem 5.1.** For a regular language $L$ of infinite trees, the following conditions are equivalent.
(1) $L$ is a Boolean combination of open sets, i.e. $L \in \mathsf{BC}(\mathbf{\Sigma}_1^0)$.
(2) Constrainer wins the game $\mathcal{H}^{\in, \notin}(L, n)$ for all but finitely many $n$.
(3) The set $\mathcal{H}_L$ has bounded alternation.
(4) The following identities are satisfied in the algebra $(H_L, V_L)$:

$$u^\sharp u w^\sharp = u^\sharp v w^\sharp = u^\sharp w w^\sharp \qquad \text{if } (u, v, w) \in \mathcal{V}_L \text{ or } (w, v, u) \in \mathcal{V}_L \qquad (5.1)$$

$$(u_2 w_2^\sharp v)^\sharp u_1 w_1^\infty = (u_2 w_2^\sharp v)^\infty \qquad \text{if } v \in V_L \text{ and } (u_1, u_2), (w_1, w_2) \in \mathcal{V}_L \qquad (5.2)$$

We have already proved the implications $1 \Leftrightarrow 2 \Leftrightarrow 3$ respectively in Corollary 3.10 and Lemma 4.28. It remains to prove $3 \Leftrightarrow 4$. The direction $3 \Rightarrow 4$ is not hard and we prove it in the next section, whereas the direction $4 \Rightarrow 3$ forms the technical core of the paper.

**Corollary 5.2.** The problem whether a regular language $L$ belongs to $\mathsf{BC}(\mathbf{\Sigma}_1^0)$ is EXP-TIME -complete, when the language $L$ is given as a parity non-deterministic automaton $\mathcal{A}$ recognising $L$.

*Proof.* Given a representation of $L$, one can compute the algebra $(H_L, V_L)$ in EXPTime, see Fact **??**. Then, verifying the equations from condition 4 of Theorem 5.1 can be done in time polynomial in the size of the algebra (which is exponential in the number of states of the given automaton for $L$).

Hardness follows immediately from EXPTime hardness of the universality problem for non-deterministic automata over finite trees [Sei89], see [Wal02b, Theorem 4.1, page 8] for a generic reduction of problems about complexity of infinite tree languages to universality of finite tree languages. □


5.1. **Overview of the proof.** The remaining part of the proof of Theorem 5.1 consists of two implications: $3 \Rightarrow 4$ (Subsection 5.2) and $4 \Rightarrow 3$ (Subsection 5.3).

The proof of the implication $3 \Rightarrow 4$ is rather straightforward: we assume that either (5.1) or (5.2) is violated and prove that the set $\mathcal{H}_L$ has unbounded alternation. This is achieved by providing concrete strategies for Alternator in the game $\mathcal{H}_L$.

The proof of the implication $4 \Rightarrow 3$ is more demanding. First, we introduce a concept of *strategy trees* that represent specific strategies of Alternator in $\mathcal{H}$ (Subsections 5.3.1 to 5.3.3). We study properties of the set $\Sigma_L$ of those strategies and notice two distinct cases of their behaviour: Case (C1) leading to a violation of (5.1); and Case (C2) leading to a violation of (5.2). See Subsection 5.3.4 for the case distinction.

The first argument, provided in Subsection 5.4, utilises the assumption of Case (C1) to construct *strategy matrices* — finite combinatorial objects witnessing the assumptions of Case (C1). By applying Erdös–Szekeres Theorem and Ramsey Theorem for hypergraphs, we gradually simplify the structure of these matrices. Ultimately, we construct a strategy matrix that literally encodes a violation of (5.1) (see Lemma 5.38).

The second argument, given in Subsection 5.5, works under the assumption of Case (C2). This proof involves another object for representing certain strategies in $G_L$: a *strategy graph* $G_L$. The edges of that directed graph encode special types of strategies of Alternator in the game $\mathcal{V}$, see Lemma 5.40. A relatively easy observation (Lemma 5.43) shows that if the graph is *recursive* (i.e. contains certain loops) then it witnesses a violation of (5.2). Subsection 5.5.2 constructs inductively such a loop in $G_L$ based on the assumption of Case (C2).


5.2. **The implication $3 \Rightarrow 4$.** In this section we prove the implication $3 \Rightarrow 4$ of Theorem 5.1 in the contra positive, as stated below.

**Proposition 5.3.** If one of the identities (5.1) and (5.2) of Theorem 5.1 is violated then the set $\mathcal{H}_L$ has unbounded alternation.


5.2.1. *The case when* (5.1) *is violated.* The assumption that (5.1) is violated says that there are $u, v, w \in V_L$ such that

$$(u, v, w) \in \mathcal{V}_L \qquad \text{or} \qquad (w, v, u) \in \mathcal{V}_L,$$

but the three context types $u^\sharp u w^\sharp$, $u^\sharp v w^\sharp$, and $u^\sharp w w^\sharp$ are not all equal. If the three context types are not equal then the second one must be different from either the first one or the

third one. We only do the proof for the case when $(u, v, w) \in \mathcal{V}_L$ and when $u^\sharp u w^\sharp \neq u^\sharp v w^\sharp$; the other cases are entirely dual. For $n \geq 0$ and $i \in \{1, \dots, n\}$, define

$$\vec{w}_{(i,n)} \stackrel{\text{def}}{=} \big( \overbrace{u, u, \dots, u}^{2(n-i)+1}, \; v, \; \overbrace{w, w, \dots, w}^{2(i-1)} \big) \quad \in (V_L)^{2n}.$$

This word is obtained from $(u, v, w)$ by duplicating some letters, and therefore it belongs to $\mathcal{V}_L$. For a given $n$, consider the words

$$\vec{w}_{(1,n)}, \dots, \vec{w}_{(n,n)} \in \mathcal{V}_L.$$

These are $n$ words of length $2n$. Let us multiply all these words coordinate-wise, yielding a word $\vec{w}_n$, also of length $2n$, which is depicted in the following picture:

<div align="center">

letter $2i{-}1$     letter $2i$

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\vec{w}_{(1,n)} =$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $v$ |
| $\vec{w}_{(2,n)} =$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ | $v$ | $w$ | $w$ |
| $\vec{w}_{(3,n)} =$ | $u$ | $u$ | $u$ | $u$ | $u$ | $v$ | $w$ | $w$ | $w$ | $w$ |
| $\vec{w}_{(4,n)} =$ | $u$ | $u$ | $u$ | $v$ | $w$ | $w$ | $w$ | $w$ | $w$ | $w$ |
| $\vec{w}_{(5,n)} =$ | $u$ | $v$ | $w$ | $w$ | $w$ | $w$ | $w$ | $w$ | $w$ | $w$ |

$u^{n-i+1}w^{i-1}$     $u^{n-i}vw^{i-1}$

</div>

As $\vec{w}_n$ is obtained by a coordinate-wise multiplication of the rows of the above matrix, and all these rows belong to $\mathcal{V}_L$, Lemma 4.25 implies that also $\vec{w}_n \in \mathcal{V}_L$.

Recall that $\sharp$ is a number dependant on $V_L$ such that for every $z \in V_L$ we know that $z^\sharp$ is an idempotent. Choose some $k$, and take $n = k \cdot \sharp + 1$, and $i \in \{\sharp + 1, 2\sharp + 1, \dots, (k-1) \cdot \sharp + 1\}$. Consider the letters $2i - 1$ and $2i$ in the word $\vec{w}_n$, which are

$$u^{n-i+1}w^{i-1} = u^\sharp u w^\sharp \qquad u^{n-i}vw^{i-1} = u^\sharp v w^\sharp.$$

By the assumption, these letters are different, and therefore the word $\vec{w}$ has alternation at least $k$. Because $k$ was chosen arbitrarily, it follows that $\mathcal{V}_L$ has unbounded alternation. Lemma 4.29 implies that in that case also $\mathcal{H}_L$ has unbounded alternation.

5.2.2. *The case when* (5.2) *is violated.* The assumption that (5.2) is violated says that $\mathcal{V}_L$ contains pairs $(u_1, u_2)$ and $(w_1, w_2)$ such that for some $v \in V_L$,

$$e^\infty \neq e u_1 w_1^\infty \qquad \text{for } e \stackrel{\text{def}}{=} (u_2 w_2^\sharp v)^\sharp.$$

Let $h_1 = e^\infty$ and $h_2 = e u_1 w_1^\infty$, by the above assumption we know that $h_1 \neq h_2$. It turns out that a violation of Equation (5.2) has even stronger consequences than a violation of Equation (5.1), as expressed by the following claim. It speaks about the infinite variant of the game of types, which is defined analogously to the finite one, see Subsection 4.5.

**Claim 5.4.** If Equation (5.2) is violated then Alternator wins $\mathcal{H}^\infty(h_1, h_2, h_1, \dots)$.

The above claim implies in particular that for every $n$ the sequence

$$\big(h_1, h_2\big)^n$$

belongs to $\mathcal{H}_L$, and thus $\mathcal{H}_L$ has unbounded alternation.

*Proof.* Let $C_{u_1}$, $C_{w_1}$ be contexts of types $u_1$, $w_1$ that witness that $(u_1, u_2), (w_1, w_2) \in \mathcal{V}_L$ i.e. they are contexts played by Alternator in the first rounds of the respective games. Let $C_{u_2}$, $C_{w_2}$, and $C_v$ be any three contexts of types $u_2$, $w_2$, and $v$ respectively. Let $C_e \stackrel{\text{def}}{=} \left(C_{u_2} C_{w_2}^\sharp C_v\right)^\sharp$. The type of the context $C_e$ is $e$.

Our aim is to provide an explicit strategy of Alternator in the game $\mathcal{H}^\infty(h_1, h_2, h_1, \ldots)$. This will be done inductively, considering the consecutive rounds of the game. The outcome will be a set of trees $(t_i)_{i<\omega}$ that are played in the considered play of the game, with $\alpha_L(t_i) = h_1$ for odd $i$ and $\alpha_L(t_i) = h_2$ for even $i$.

The strategy of Alternator starts with the tree $t_1 = \left(C_e\right)^\infty$. We will demonstrate how it works for the first three rounds of the game, the rest is analogous.

Consider a prefix $p_1$ of the tree $t_1$ that is fixed by Constrainer. It must be the case that $p_1$ is a prefix of $\left(C_e\right)^{n_1}$ for some $n_1$. Thus, Alternator can now provide the tree

$$t_2 = \left(C_e\right)^{n_1} \cdot C_{u_1} \cdot C_{w_1}^\infty.$$

Now Constrainer chooses a prefix $p_2$ of the above tree, in fact $p_2$ is a prefix of $\left(C_e\right)^{n_1} \cdot C_{u_1} \cdot C_{w_1}^{\sharp \cdot n_2}$ for some $n_2$. Thus, by the assumptions on $C_{u_1}$ and $C_{u_2}$, Alternator is able to provide a tree of the form

$$t_2 = \left(C_e\right)^{n_1} \cdot \left(D_{u_2} \cdot D_{w_2}^{\sharp \cdot n_2} \cdot C_v\right) \cdot \left(C_e\right)^\infty,$$

where the contexts $D_{u_2}$ and $D_{w_2}$ depend on the prefix $p_2$ and have types $u_2$ and $w_2$ respectively. Now we proceed inductively as before, because any prefix $p_3$ of $t_2$ must be a prefix of $\left(C_e\right)^{n_1} \cdot \left(D_{u_2} \cdot D_{w_2}^{\sharp \cdot n_2} \cdot C_v\right) \cdot \left(C_e\right)^{n_3}$.

Notice that by the choice of $e$, the type of the context $D_{u_2} \cdot D_{w_2}^{\sharp \cdot n_2} \cdot C_v$ is $e$. Therefore, $\alpha_L(t_i) = e^\infty = h_1$ for odd $i$ and $\alpha_L(t_i) = eu_1w_1^\infty = h_2$ for even $i$. $\qquad \square$

Consider a pair of trees $t_1$ and $t_2$ of types $h_1$ and $h_2$ respectively. Since $t_1 \stackrel{\mathrm{Tr}}{\not\approx}_L t_2$, there exists a multicontext $C$ such that[4]

$$C[t_1] \in L \ \land \ C[t_2] \notin L, \tag{5.3}$$

see Definition 4.4. Now, by Item 1 of Lemma 4.12 we know that the types $\alpha_L\big(C[t_1]\big)$, $\alpha_L(C[t_2])$ (and therefore the conditions in (5.3)) do not depend on the actual choice of the trees $t_1 \in h_1$ and $t_2 \in h_2$. Thus, by using the above strategy for Alternator under the multicontext $C$, we obtain the following corollary.

**Corollary 5.5.** *If Equation (5.2) is violated then Alternator wins $\mathcal{H}^\infty(L)$.*

5.3. **The implication** $4 \Rightarrow 3$ — **case distinction.** We now move to the implication $4 \Rightarrow 3$ of Theorem 5.1, which is the most involved part of the article. To prove it, we need to introduce a crucial concept witnessing a large alternation of the set $\mathcal{H}_L$. The objects witnessing that will be called *strategy trees* and *locally optimal strategy trees*. Using these objects we will split the proof of that implication into two separate cases, see Subsection 5.3.4. We deal with these cases in Subsections 5.4 and 5.5..

---

[4]We swap $h_1$ and $h_2$ if needed.

5.3.1. *Strategy trees.* First, a *type-labelled tree* is a tree such that the nodes are labelled with tree types, i.e. it is a tree over the alphabet $H_L$.

**Definition 5.6.** If $t$ is a tree over the alphabet $A$, the *type-labelled tree induced by $t$* is the type-labelled tree $\sigma$ where the label of a node $u$ is the tree type of the subtree $t.u$, i.e. $\sigma(u) = \alpha_L(t.u)$. In particular $\sigma(\epsilon) = \alpha_L(t)$.

**Definition 5.7.** Let $\sigma$ be a type-labelled tree and let $t$ be a tree over $A$. We say that $\sigma$ is *locally consistent* with $t$ if for every node $u \in \{\text{L}, \text{R}\}^*$, whose label in $t$ is $a$, we have that $\sigma(u)$ is the type obtained by applying the letter $a$ to the pair of types $\sigma(u\text{L})$ and $\sigma(u\text{R})$, that is

$$\sigma(u) = a\big(\sigma(u\text{L}), \Box\big) \cdot \sigma(u\text{R}). \tag{5.4}$$

In other words, if we take a tree $t_1$ inside $\sigma(u\text{L})$ and a tree $t_2$ inside $\sigma(u\text{R})$ and we plug these two trees as the left and the right children of $a$, then the type of the result is $\sigma(u)$.

**Remark 5.8.** The type-labelled tree induced by $t$ is locally consistent with $t$.

**Example 5.9.** Consider the language

$$L = \{t \in \text{Tr}_A \mid t \text{ contains at least one } b\}$$

over the alphabet $A = \{a, b\}$. $H_L$ contains two tree types that (treated as sets of trees) are $L$ and $L^c$. Now consider a type-labelled tree $\sigma$ where every node is labelled with $L$. It is easy to check that $\sigma$ is locally consistent with every tree $t \in \text{Tr}_A$, even with the tree that contains only letters $a$.

**Fact 5.10.** The set of pairs

$$\big\{(t, \sigma) \in \text{Tr}_A \times \text{Tr}_{H_L} \mid \sigma \text{ is locally consistent with } t\big\}$$

is closed in the product topology.

*Proof.* As Condition (5.4) speaks about finitely many values of $\sigma$ and $t$, it corresponds to a clopen set of pairs. A type-labelled tree $\sigma$ is locally consistent with a tree $t$ if they obey the local consistency conditions from (5.4) in all the vertices. Thus, the above defined set of pairs is an intersection of a family of clopen sets. □

**Lemma 5.11.** Let $(t_n)_{n\in\mathbb{N}}$ be a sequence of trees that converges to $t_*$ and let $(\sigma_n)_{n\in\mathbb{N}}$ be a sequence of type-labelled trees that converges to $\sigma_*$. If $\sigma_n$ is locally consistent with $t_n$ for every $n$ then $\sigma_*$ is locally consistent with $t_*$.

*Proof.* Follows directly from Fact 5.10. □

Now we are ready to define strategy trees, a key concept of this paper.

**Definition 5.12.** A *strategy tree* is a tuple $\sigma = (t, \sigma_1, \ldots, \sigma_n)$ where:

(1) $t$ is a tree over $A$, called the *support* of $\sigma$.
(2) $\sigma_1$ is the type-labelled tree induced by $t$.
(3) The type-labelled trees $\sigma_2, \ldots, \sigma_n$ are locally consistent with $t$.
(4) For each node $u$ of $t$, the sequence $\big(\sigma_2(u), \ldots, \sigma_n(u)\big)$ belongs to $\mathcal{H}_L$.

Notice that the fourth condition of Definition 5.12 does not mention the type-labelled tree $\sigma_1$. By the definition, $\sigma$ can be interpreted as a single tree over the alphabet $A \times H_L^n$.

Intuitively speaking, a strategy tree represents a special kind of strategy for Alternator. In the first round, Alternator plays the support $t$ of $\sigma$. However, Alternator also declares all the types that will appear in the nodes of $t$ as the game progresses. More specifically, he

declares that for every node $u$ of $t$ and round $k \in \{2, \ldots, n\}$, he has a strategy so that for the tree played in the round $k$, the subtree in the node $u$ has type $\sigma_k(u)$.

**Alternations.** The number $n$ is called the *duration* of a strategy tree $\sigma = (t, \sigma_1, \ldots, \sigma_n)$. We define the *root sequence* of a strategy tree to be the sequence of root labels of $\sigma_1, \ldots, \sigma_n$. If the duration is $n$, the root sequence is in $H_L^n$. We define the *root alternation* of a strategy tree $\sigma$ to be the alternation of its root sequence. We define the *limit alternation* of $\sigma$ to be the maximal number $\ell$ such that infinitely many subtrees of $\sigma$ have root alternation at least $\ell$. This means that if the limit alternation of $\sigma$ is $\ell$ then there exist infinitely many nodes $u$ such that their root sequence (i.e. the root sequence of the strategy tree obtained by truncating $\sigma$ in the node $u$) has alternation at least $\ell$.

**Context zones.** A *context zone* is a set $X$ of nodes for which there exists a node $u$, called the *root* of $X$, and a node $y$, called the *port* of $X$, such that $u \prec y$ and $X$ contains the nodes that are in the subtree of $u$, but not in the subtree of $y$:

$$X = \big\{ x \in \{\mathtt{L}, \mathtt{R}\}^* \mid u \preceq x \wedge y \npreceq x \big\}.$$

We say that context zones $X_1, \ldots, X_n$ are *consecutive* if for each $i \in \{1, \ldots, n-1\}$, the port of $X_i$ is the root of $X_{i+1}$. The union of consecutive context zones is a context zone.

Consider a tree $t$, a type-labelled tree $\sigma$, and a context zone $X$. $X$ can be seen as a context inside $t$ taking into account the types of the subtrees of $t$ as declared in $\sigma$. This is achieved by defining a value $\mathsf{val}(t, \sigma, X) \in V_L$. The definition of $\mathsf{val}(t, \sigma, X)$ is inductive on the number of nodes $v$ in $X$ such that $v \preceq y$, where $y$ is the port of $X$. If there is only one such node then the port $y$ of $X$ is a child of its root $u$. Let $a$ be the label of $u$ in $t$ and $v \in X$ be the other child of $u$. We set $\mathsf{val}(t, \sigma, X)$ as $a(\square, \sigma(v))$ if $v$ is the right child and $a(\sigma(v), \square)$ if $v$ is the left child. Otherwise, let $u$ be the root of $X$, $y$ its port and $z$ the child of $u$ such that $z \preceq y$. Let $X_1$ be the context zone with root $u$ and port $z$ and $X_2$ the context zone of root $z$ and port $y$. We define

$$\mathsf{val}(t, \sigma, X) \stackrel{\text{def}}{=} \mathsf{val}(t, \sigma, X_1) \cdot \mathsf{val}(t, \sigma, X_2).$$

Figure 5 illustrates an example of a tree and a context zone with

$$\mathsf{val}(t, \sigma, X) = b(\square, h_1) \cdot c(h_2, \square) \in V_L.$$

In other words, $\mathsf{val}(t, \sigma, X)$ is the value of the context $C \stackrel{\text{def}}{=} t{\restriction}X$ when we assume that the subtrees of $t$ aside of the branch leading to the port of $C$ have types as declared by $\sigma$.

Now assume that $\sigma = (t, \sigma_1, \ldots, \sigma_n)$ is a strategy tree. Let $X$ be a context zone of $\sigma$ (we can see $\sigma$ as a single tree). For a round $i \in \{1, \ldots, n\}$, we define the type

$$\mathsf{val}(\sigma, X, i) \stackrel{\text{def}}{=} \mathsf{val}(t, \sigma_i, X) \in V_L.$$

**Fact 5.13.** Let $\sigma = (t, \sigma_1, \ldots, \sigma_n)$ be a strategy tree and $X$ be a context zone. Then

$$\big( \mathsf{val}(\sigma, X, 1), \ldots, \mathsf{val}(\sigma, X, n) \big) \in \mathcal{V}_L.$$

*Proof.* This is by construction and Items 3, 6 and 7 of Lemma 4.25.  $\square$
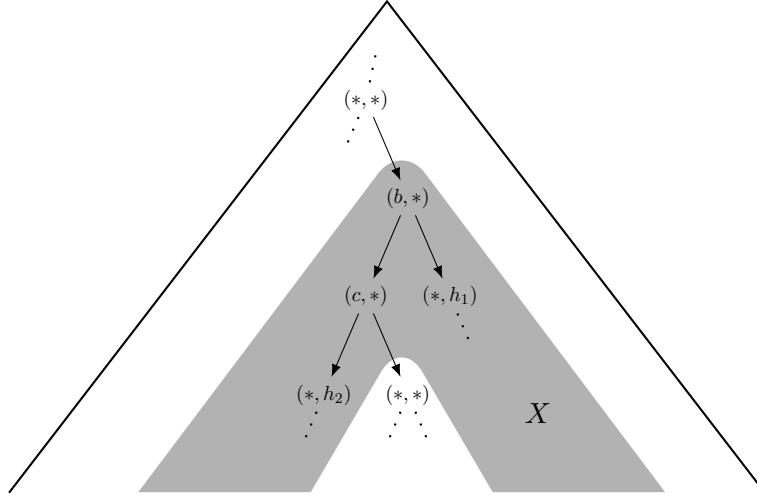
Figure 5: An illustration to the value $\mathsf{val}(t, \sigma, X)$. The grey area is the set $X$, a label $(a, h)$ of a node $u$ represents the values of $t(u) = a$ and $\sigma(u) = h$ respectively. The symbol $*$ indicates values that are irrelevant for the value of the set $X$.

5.3.2. *Existence of strategy trees.* We will now prove that whenever Alternator has a strategy in $\mathcal{H}$ then this fact is witnessed by a strategy tree. This fact is expressed by the following two lemmas. For inductive reasons these lemmas are parametrised by a finite multicontext $C$. The games $\mathcal{H}_C(h_1, \ldots, h_n)$ and $\mathcal{V}_C(h_1, \ldots, h_n)$ are defined as the games $\mathcal{H}_U(h_1, \ldots, h_n)$ and $\mathcal{V}_U(h_1, \ldots, h_n)$ respectively, where $U$ is the open set of all trees (resp. contexts) that can be obtained from $C$, i.e. $C[*]$.

**Lemma 5.14.** Let $C$ be a finite multicontext for strategy trees. Consider a sequence of valuations $\eta_1, \ldots, \eta_n$ that map ports of $C$ to $H_L$. If

$$(\eta_1(u), \ldots, \eta_n(u)) \in \mathcal{H}_L$$

for every port $u$, then Alternator wins the game

$$\mathcal{H}_C(C[\eta_1], \ldots, C[\eta_n]).$$

*Proof.* For every port $u$ of $C$, Alternator has a winning strategy for the game

$$\mathcal{H}(\eta_1(u), \ldots, \eta_n(u)).$$

We describe a winning strategy for Alternator in $\mathcal{H}_C(C[\eta_1], \ldots, C[\eta_n])$. Alternator starts with a tree obtained by plugging the initial tree from his winning strategy in the game $\mathcal{H}(\eta_1(u), \ldots, \eta_n(u))$ in every port $u$ of $C$. Then every prefix of this tree yields a prefix for each subtree plugged into a port of $C$ and Alternator can then use his strategies for the game $\mathcal{H}(\eta_1(u), \ldots, \eta_n(u))$ to answer. $\qquad\square$

**Lemma 5.15.** For a sequence $(h_1, \ldots, h_n) \in H_L^*$ and a finite multicontext $C$ the following conditions are equivalent:
(1) Alternator wins $\mathcal{H}_C(h_1, \ldots, h_n)$.
(2) There is a strategy tree whose support extends $C$, with root sequence $(h_1, \ldots, h_n)$.

*Proof.* We prove the lemma by induction on $n$. The base case when $n = 0$ or $n = 1$, is trivial. We do the induction step. Let us begin with the easier implication from 2 to 1. Suppose that

$$\sigma \equiv (t, \sigma_1, \ldots, \sigma_n)$$

is a strategy tree as in item 2. Alternator's strategy is as follows. In the first round, he plays the tree $t$, which has type $h_1$. Suppose Constrainer chooses a prefix $p$ of $t$. Let $D$ be the multicontext obtained from $p$ by labelling all the $\preceq$-minimal elements outside of $\mathrm{dom}(p)$ by $\square$. Consider the valuations $\eta_2, \ldots, \eta_n$ that map all the ports of $D$ to $H_L$ with $\eta_i(u) = \sigma_i(u)$ for any port $u$ of $D$. For every $i \in \{2, \ldots, n\}$, the root label of $\sigma_i$ is the same as $D[\eta_i]$, because $\sigma_i$ is locally consistent with $t$ and $D$ is obtained from a prefix of $t$. By Lemma 5.14, Alternator wins the game

$$\mathcal{H}_D(D[\eta_2], \ldots, D[\eta_n]).$$

This shows that for every choice of $p$, Alternator has a winning strategy in the remaining part of the game.

Now we move to the more difficult implication from 1 to 2.

Suppose that Alternator wins $\mathcal{H}_C(h_1, \ldots, h_n)$. Let $t$ be the tree of type $h_1$ that Alternator plays in the first round. This tree has prefix $C$. Also, for every finite prefix $D$ of $t$, Alternator wins $\mathcal{H}_D(h_2, \ldots, h_n)$. By the induction assumption, for every finite prefix $D$ of $t$, there is a strategy tree

$$\sigma_D = (t_D, \sigma_{D_2}, \ldots, \sigma_{D_n})$$

such that $t_D$ has prefix $D$, and the root sequence of $\sigma_D$ is $(h_2, \ldots, h_n)$.

A sequence of finite multicontexts $(D_i)_{i \in \omega}$ is said to converge to $t$ if all of the multicontexts are prefixes of $t$, and for every $j \in \omega$, only finitely many multicontexts have some port at depth at most $j$. By compactness, there is an infinite sequence of finite multicontexts $(D_i)_{i \in \omega}$ which converges to the tree $t$ and such that all of the sequences

$$(t_{D_i})_{i \in \omega} \qquad (\sigma_{D_i 2})_{i \in \omega} \qquad \ldots \qquad (\sigma_{D_i n})_{i \in \omega}$$

are convergent. Let the limits of these sequences be

$$t_* \qquad \sigma_{*2} \qquad \ldots \qquad \sigma_{*n}.$$

Because the sequence $(D_i)_{i \in \omega}$ converges to $t$, it follows that $t_* = t$. For each $D$, the type trees

$$(\sigma_{D2}, \ldots, \sigma_{Dn})$$

are locally consistent with $t_D$. Therefore, by Lemma 5.11 it follows that the limits $\sigma_{*2}, \ldots, \sigma_{*n}$ are locally consistent with $t$. Finally, define $\sigma_{*1}$ to be the unique type tree that is globally consistent with $t$. We have just proved that

$$(\sigma_{*1}, \ldots, \sigma_{*n})$$

is a strategy tree. Because root values are preserved under limits, the root value of this strategy tree is the desired $(h_1, \ldots, h_n)$. $\square$

5.3.3. *Locally optimal strategy trees.* To make the structure of a strategy tree more rigid, we will introduce the notion of a *locally optimal strategy tree.* In such a strategy tree, whenever $\sigma_i(u) \neq \sigma_{i+1}(u)$, there is some concrete reason for that fact.

**Definition 5.16.** Consider a strategy tree $(t, \sigma_1, \ldots, \sigma_n)$ and let $v \in V_L \sqcup \{1_L\}$. The strategy tree is called *locally optimal for $v$* if for every $i \in \{2, \ldots, n\}$ and for every type-labelled tree $\sigma'$, if $\sigma'$ is locally consistent with $t$ and $v \cdot \sigma'(\epsilon) = v \cdot \sigma_i(\epsilon)$, then

$$\lambda(\sigma_{i-1}, \sigma_i) \leq \lambda(\sigma_{i-1}, \sigma'),$$

where $\lambda$ is the discounted distance.

If $\sigma$ is optimal for the context type $1_L$ of the trivial context $\square$ then we just say that $\sigma$ is *locally optimal.* Let $\Sigma_L$ denote the set of all locally optimal strategy trees for $L$.

**Fact 5.17.** If $\sigma$ is locally optimal for $u \cdot v$ then $\sigma$ is locally optimal for $v$ as well.

**Lemma 5.18.** Consider a strategy tree $\sigma$ with root sequence $(h_1, \ldots, h_n)$. Let $v \in V_L \sqcup \{1_L\}$ be a context type. Then there exists a locally optimal strategy tree $\sigma'$ for $v$ with root sequence $(h'_1, \ldots, h'_n)$ such that

$$v \cdot (h_1, \ldots, h_n) = v \cdot (h'_1, \ldots, h'_n). \tag{5.5}$$

*Proof.* Take a strategy tree $\sigma = (t, \sigma_1, \ldots, \sigma_n)$. Consider the set $\Sigma_2$ of type-labelled trees $\sigma'_2$ that are:

- locally consistent with $t$,
- if $h$ (resp. $h'$) is the root value of $\sigma_2$ (resp. $\sigma'_2$) then $v \cdot h = v \cdot h'$.

Fact 5.10 implies that $\Sigma_2$ is a closed set. As a closed subset of the compact space of all trees, $\Sigma_2$ is compact. It follows that some elements of $\Sigma_2$ minimise the discounted distance with respect to $\sigma_1$. We choose such an element as the new $\sigma_2$ and we iterate this mechanism to build a locally optimal strategy tree $\bar{\sigma}$. This tree satisfies condition (5.5). $\square$

Notice that in the above construction, if $v \neq 1_L$ then the root sequence of the new strategy tree might be different than the original root sequence. However, if $v = 1_L$ then the root sequence is not modified.

**Corollary 5.19.** If $(h_1, h_2, \ldots, h_n) \in \mathcal{H}_L$ then there exists a locally optimal strategy tree $\sigma$ with root sequence $(h_1, \ldots, h_n)$.

Using the above properties, without loss of generality we can restrict our attention to locally optimal strategy trees. One of the direct consequences of working with such strategy trees is expressed by the following lemma.

**Lemma 5.20.** If $\sigma$ is a locally optimal strategy tree, $u \preceq y$ two nodes of $\sigma$ then for all $i = 1, \ldots, n$ we have

$$\sigma_i(y) \neq \sigma_{i+1}(y) \Rightarrow \sigma_i(u) \neq \sigma_{i+1}(u).$$

*Proof.* Assume that $\sigma_i(y) \neq \sigma_{i+1}(y)$ and $\sigma_i(u) = \sigma_{i+1}(u)$. We show that this contradicts local optimality. Consider the type-labelled tree $\sigma'_{i+1}$ defined as follows:

- for every $z$ such that $u \preceq z$, $\sigma'_{i+1}(z) = \sigma_i(z)$,
- for all other nodes $z$, $\sigma'_{i+1}(z) = \sigma_{i+1}(z)$.

By the construction, $\sigma'_{i+1}$ is locally consistent with $t$ (this is by the definition for all nodes $z \neq u$, and because $\sigma_i(u) = \sigma_{i+1}(u)$). Note that by the definition, for all nodes $z$:

$$\mathsf{dist}(\sigma_i(z), \sigma'_{i+1}(z)) \leq \mathsf{dist}(\sigma_i(z), \sigma_{i+1}(z))$$

Moreover, $\mathsf{dist}(\sigma_i(y), \sigma'_{i+1}(y)) = 0$ and $\mathsf{dist}(\sigma_i(y), \sigma_{i+1}(y)) = 1$. Combining all this we obtain:

- $\lambda(\sigma_i, \sigma'_{i+1}) < \lambda(\sigma_i, \sigma_{i+1})$.
- $\sigma'_{i+1}(\epsilon) = \sigma_{i+1}(\epsilon)$, as sequences in $(H_L)^n$.

This contradicts local optimality of $\sigma$. $\qquad\square$

**Corollary 5.21.** If $\sigma$ is a locally optimal strategy tree and $u \preceq y$ are two nodes of $\sigma$ then the root alternation of $\sigma.u$ is not smaller than the root alternation of $\sigma.y$.


5.3.4. *Case distinction.* We can now split the proof of the implication $4 \Rightarrow 3$ of Theorem 5.1 into two subcases, as discussed below. Then we will treat these cases separately, as expressed by Propositions 5.22 and 5.23.

Assume for the sake of contradiction that Condition 3 of Theorem 5.1 is violated, what means that $\mathcal{H}_L$ has unbounded alternation. Thus, by Corollary 5.19 we know that the root alternation of the set $\Sigma_L$ of all locally optimal strategy trees for $L$ is unbounded. Now consider the following two dual subcases:

**(C1):** $\Sigma_L$ has unbounded root alternation and there exists a subset $\Sigma' \subseteq \Sigma_L$ that has unbounded root alternation but bounded limit alternation.

**(C2):** $\Sigma_L$ has unbounded root alternation and every subset $\Sigma' \subseteq \Sigma_L$ with unbounded root alternation has also unbounded limit alternation.

The following two propositions consider these cases separately:

**Proposition 5.22.** Assuming (C1), Equation (5.1) is violated.

**Proposition 5.23.** Assuming (C2), Equation (5.2) is violated.

The proofs of these propositions are given in Subsections 5.4 and 5.5. Notice that since either Case (C1) or Case (C2) must hold, both these propositions together complete the proof of the implication $4 \Rightarrow 3$ of Theorem 5.1.

Roughly speaking, Case (C1) corresponds to the situation in which the high alternation of $\mathcal{H}_L$ is achieved by modifications in the strategy trees that are spread inside their structure. On the opposite, Case (C2) corresponds to the situation in which the high alternation occurs along some infinite branch of the considered strategy trees.


5.4. **Case (C1) — limit alternation is bounded.** In this section we assume that $\Sigma' \subseteq \Sigma_L$ is a set of locally optimal strategy trees such that the root alternation of $\Sigma'$ is unbounded but the limit alternation of $\Sigma'$ is bounded. Under that assumption we prove that Equation (5.1) is violated. We do this in two steps, using a new object called *strategy matrix* as an intermediary. Strategy matrices represent special strategies for Alternator in the game on tree types. In our first step, we show that if the root alternation of $\Sigma'$ is unbounded but the limit alternation of $\Sigma'$ is bounded then there exist special strategy matrices of arbitrarily large size. Finally we show that the existence of sufficiently large special strategy matrices violates Equation (5.1).

column $2i{-}1$     column $2i$

| | | $u_1$ $u_1$ | | | |
|---|---|---|---|---|---|
| | | $u_2$ $u_2$ | | | |
| | | $x$ $\phantom{}y$ | | | |
| | | $w_4$ $w_4$ | | | |
| | | $w_5$ $w_5$ | | | |
| | | $w_6$ $w_6$ | | | |

Figure 6: A matrix in $\mathbb{P}_6$.

**Definition 5.24.** A *strategy matrix* is a rectangular matrix with entries from $V_L$ such that every row belongs to $\mathcal{V}_L$. The *value* of a column of a strategy matrix is the value in $V_L$ obtained by multiplying the entries in that column in $V_L$ in the top-to-bottom order.

**Definition 5.25.** A strategy matrix $M$ is called *parity alternating* if for some $n \in \omega$ it has $2n$ columns and $n$ rows, and one of the following conditions holds (see Figure 6):

a) For every $i \in \{1, \dots, n\}$,
  − Columns $2i - 1$ and $2i$ have the same entries in all rows except for row $i$.
  − The values of columns $2i - 1$ and $2i$ are different.
b) Condition a) holds when the order of columns is reversed.

   If the case $a$) holds then the matrix is called *top-down* and if the case $b$) holds then it is called *bottom-up*. The set of parity alternating matrices with $n$ rows and $2n$ columns is denoted by $\mathbb{P}_n$.

   Figure 6 depicts a top-down parity alternating strategy matrix in $\mathbb{P}_6$. The picture presents columns $2i - 1$ and $2i$ for $i = 3$, the entries of these columns agree everywhere except the third row, where the values $x$ and $y$ are distinct. The differences in all the other pairs of columns are indicated by grey rectangles, their values are not written down for the sake of clarity. It is important that the definition of a parity alternating strategy matrix requires the total values of the respective columns to differ, not only the entries in grey rectangles.

5.4.1. *Constructing strategy matrices.* As we have already said, we want to prove that under our assumptions we can build arbitrary large strategy matrices: the goal that we aim now, is to prove that if $\Sigma'$ has unbounded root alternation and bounded limit alternation, then $\mathbb{P}_n$ is non-empty for every $n \in \omega$.

   Fix some arbitrary $n \in \omega$. We want to construct a strategy matrix $M \in \mathbb{P}_n$. Let $\ell$ be the maximal limit alternation among the strategy trees in $\Sigma'$. Choose a strategy tree $\sigma \in \Sigma'$ with root alternation at least $\ell \cdot 2^{n^2}$ and let $m$ be the duration of $\sigma$, i.e. $\sigma = (t, \sigma_1, \dots, \sigma_m)$.

   During the proof we will use a known combinatorial fact that we recall now.

**Theorem 5.26** (Erdös–Szekeres Theorem, see [ES35]). For given $r, s \in \mathbb{N}$, any sequence of length at least $(r{-}1)(s{-}1)+1$ contains a monotonically increasing subsequence of length $r$ or a monotonically decreasing subsequence of length $s$.

**Lemma 5.27.** There are consecutive contexts zones $X_1, \ldots, X_n$ in $\sigma$ and a sequence of rounds $i_1, \ldots, i_n \in \{1, \ldots, m-1\}$ such that the sequence of rounds is either strictly increasing or strictly decreasing, and

$$\mathsf{val}(\sigma, X_j, i_j) \neq \mathsf{val}(\sigma, X_j, i_j{+}1) \qquad \text{for every } j \in \{1, \ldots, n\}$$

*Proof.* For a round $i \in \{1, \ldots, m{-}1\}$, define:

$$\mathsf{change}_i(\sigma) = \{x \in \{\text{L}, \text{R}\}^* \mid \sigma_i(x) \neq \sigma_{i+1}(x)\}.$$

Now we prove three different claims. Once these claims are proved, we easily finish the proof of the lemma.

**Claim 1.** Let $X$ be a context zone, and let $\pi$ be an infinite path that passes through the root of $X$, but not through the port. Then

$$\pi \subseteq \mathsf{change}_i(\sigma) \quad \Longrightarrow \quad \mathsf{val}(\sigma, X, i) \neq \mathsf{val}(\sigma, X, i{+}1)$$

holds for every round $i \in \{1, \ldots, m-1\}$.

*Proof of Claim 1.* This is by local optimality of $\sigma$. Assume that $\pi \subseteq \mathsf{change}_i(\sigma)$ and $\mathsf{val}(\sigma, X, i) = \mathsf{val}(\sigma, X, i{+}1)$ for some round $i$. We construct a type-labelled tree $\sigma'_{i+1}$ that is closer to $\sigma_i$ than $\sigma_{i+1}$ regarding the discounted distance $\lambda$ and with the same root value as $\sigma_{i+1}$, contradicting local optimality.

Consider the type-labelled tree $\sigma'_{i+1}$ defined as follows:

- For nodes $x \in X$ $\sigma'_{i+1}(x) = \sigma_i(x)$. Hence, for $x \in X$

$$\mathsf{dist}(\sigma_i(x), \sigma'_{i+1}(x)) = 0 \leq \mathsf{dist}(\sigma_i(x), \sigma_{i+1}(x))$$

- For other nodes $y$ we define $\sigma'_{i+1}(y) = \sigma_{i+1}(y)$. Hence

$$\mathsf{dist}(\sigma_i(y), \sigma'_{i+1}(y)) = \mathsf{dist}(\sigma_i(y), \sigma_{i+1}(y)).$$

Because $\pi \subseteq \mathsf{change}_i(\sigma)$, there exists at least one node $x \in X$ such that s

$$\mathsf{dist}(\sigma_i(x), \sigma_{i+1}(x)) = 1.$$

It follows that:

$$\lambda(\sigma_i, \sigma'_{i+1}) < \lambda(\sigma_i, \sigma_{i+1})$$

Moreover since $\mathsf{val}(\sigma, X, i) = \mathsf{val}(\sigma, X, i{+}1)$, we have $\sigma'_{i+1}(\epsilon) = \sigma_{i+1}(\epsilon)$. This contradicts local optimality of $\sigma$. $\square$

**Claim 2.** There is a set $\Pi$ of at least $2^{n^2}$ distinct infinite paths, and a function $\mathsf{when} \colon \Pi \to \{1, \ldots, m-1\}$ such that for every $\pi \in \Pi$

$$\pi \subseteq \mathsf{change}_{\mathsf{when}(\pi)}(\sigma).$$

*Proof of Claim 2.* By consistency of a strategy tree, every node in $\mathsf{change}_j(\sigma)$ has at least one child in $\mathsf{change}_j(\sigma)$. Therefore, each of the non-empty sets $\mathsf{change}_j(\sigma)$ contains at least one infinite path. By the assumption on the root alternation there are at least $\ell \cdot 2^{n^2}$ rounds $j$ where $\mathsf{change}_j(\sigma)$ is non-empty. By the assumption on limit alternation, an infinite path can be contained in sets $\mathsf{change}_j(\sigma)$ for at most $\ell$ different values of $j$. $\square$

**Claim 3.** Let $\Pi$ be a set of $2^k$ distinct infinite paths in a binary tree. There exist consecutive context zones $X_1, \ldots, X_k$ and paths $\pi_1, \ldots, \pi_k \in \Pi$ such that for every $i \in \{1, \ldots, k\}$, the path $\pi_i$ passes through the ports of the context zones $X_1, \ldots, X_{i-1}$, but not through the port of $X_i$.

*Proof of Claim 3.* The proof is by induction on $k$. The induction base of $k = 1$ is obvious.

Now assume that the thesis holds for $k$ and consider a set $\Pi$ of $2^{k+1}$ paths. Let $u$ be the deepest node in the tree that belongs to all paths of $\Pi$ ($u$ exists since the root belongs to all paths of $\Pi$). Let $\Pi_{\mathsf{L}}$ (respectively, $\Pi_{\mathsf{R}}$) be those paths in $\Pi$ that pass through the left child of $u$ (respectively, the right child of $u$). One of the sets $\Pi_{\mathsf{L}}$ or $\Pi_{\mathsf{R}}$ must have at least half of the paths, i.e. at least $2^k$ paths. By symmetry, assume that $\Pi_{\mathsf{L}}$ has at least $2^k$ paths and let $y$ be the left child of $u$. We apply the induction hypothesis to $\Pi_{\mathsf{L}}$ and obtain paths $\pi_2, \ldots, \pi_{k+1} \in \Pi_{\mathsf{L}}$ and consecutive context zones $X_2, \ldots, X_{k+1}$ such that for every $i \in \{2, \ldots, k+1\}$, path $\pi_i$ passes through the ports of contexts $X_2, \ldots, X_{i-1}$, but not through the port of $X_i$.

We slightly modify $X_2$ by setting $y$ as its root. Note that this does not affect the properties of the paths $\pi_2, \ldots, \pi_{k+1} \in \Pi_{\mathsf{L}}$. Now, we define $X_1$ as the context zone with the root $u$ and the root of $X_2$ as the port. Let $\pi_1$ be some arbitrary path in $\Pi_{\mathsf{R}}$. By the definition, $X_1, \ldots, X_{k+1}$ are consecutive context zones. Moreover, the paths $\pi_2, \ldots, \pi_{k+1}$ are paths of $\Pi_{\mathsf{L}}$ and therefore pass through $y$, i.e. the port of $X_{\mathsf{L}}$. Finally, by the definition $\pi_1$ passes through the right child of $u$ and therefore not through the port of $X_1$. □

Now we can easily complete the proof of Lemma 5.27. Let $\Pi$ and the function when be as in Claim 2. Apply Claim 3 to $\Pi$, yielding a sequence of paths, $\tau_1, \ldots, \tau_{n^2}$, and a sequence of context zones, $Y_1, \ldots, Y_{n^2}$, such that for every $i \in \{1, \ldots, n^2\}$, the path $\tau_i$ passes through the ports of context zones $Y_1, \ldots, Y_{i-1}$, but not through the port of $Y_i$. Consider the sequence

$$\mathsf{when}(\tau_1), \ldots, \mathsf{when}(\tau_{n^2}) \in \{1, \ldots, m-1\}.$$

By Erdös–Szekeres Theorem, we can find a sequence of indexes

$$j_1 < \ldots < j_n \in \{1, \ldots, n^2\}$$

such that the sequence

$$\mathsf{when}(\tau_{j_1}), \ldots, \mathsf{when}(\tau_{j_n})$$

is either increasing or decreasing. For $i \in \{1, \ldots, n\}$, define $\pi_i$ to be $\tau_{j_i}$ and $X_i$ to be the union of the context zones

$$Y_{j_i} \cup \ldots \cup Y_{j_{i+1}-1}.$$

By the construction, we know that the path $\pi_i$ passes through the ports of the context zones $X_1, \ldots, X_{i-1}$, but not through the port of the context zone $X_i$. By Claim 1 we know that

$$\mathsf{val}\big(\sigma, X_i, \mathsf{when}(\pi_i)\big) \neq \mathsf{val}\big(\sigma, X_i, \mathsf{when}(\pi_i)+1\big).$$

Therefore, the proof of Lemma 5.27 is complete if we define $i_1, \ldots, i_n$ to be

$$\mathsf{when}(\pi_1), \ldots, \mathsf{when}(\pi_n).$$

□

**Definition 5.28.** Let $M$ be a matrix and consider a row $j$ and a column $i$ of $M$ which is not the first column. We define a new column, denoted by

$$\text{almostcopy}_i^{\neq j}(M),$$

as follows: $\text{almostcopy}_i^{\neq j}(M)$ is equal to column $i$ of $M$ in all rows except for row $j$, where it is equal to column $i - 1$ of $M$.

**Definition 5.29.** Let $\sigma$ be a strategy tree of duration $m$ and let $X_1, \ldots, X_n$ be consecutive context zones. We define a matrix with $n$ rows and $m$ columns as follows (for $i = 1, \ldots, m$ and $j = 1, \ldots, n$):

$$\text{matrix}\big(\sigma, X_1, \ldots, X_n\big)\,[i,j] \stackrel{\text{def}}{=} \text{val}(\sigma, X_j, i).$$

Note that it follows from Fact 5.13 that every row of $\text{matrix}(\sigma, X_1, \ldots, X_n)$ belongs to $\mathcal{V}_L$ and therefore it is a strategy matrix.

**Lemma 5.30.** Let $N$ be a strategy matrix defined by

$$N = \text{matrix}(\sigma, X_1, \ldots, X_n),$$

for some locally optimal strategy tree $\sigma$ and consecutive context zones $X_1, \ldots, X_n$. Let $j$ be a row and $i$ a column, which is not the first column. If columns $i$ and $i - 1$ in $N$ have different entries in row $j$ then the value of the column

$$\text{almostcopy}_i^{\neq j}(N)$$

is different from the value of column $i$ in $N$.

*Proof.* This is a consequence of local optimality of $\sigma$. The proof is the same as Claim 2 in the proof of Lemma 5.27. $\square$

Now we are ready to prove the first intermediary step: constructing a matrix in $\mathbb{P}_n$.

**Proposition 5.31.** Under our assumptions about the strategy tree $\sigma$ we can construct a matrix $M \in \mathbb{P}_n$.

*Proof.* Let $X_1, \ldots, X_n$ and $i_1, \ldots, i_n$ be as in Lemma 5.27. Consider the strategy matrix $N = \text{matrix}(\sigma, X_1, \ldots, X_n)$.

By Lemma 5.27 we know that for every $j \in \{1, \ldots, n\}$ the entries in row $j$ are different in columns $i_j$ and $i_j + 1$.

Suppose first that the sequence $i_1, \ldots, i_n$ is strictly increasing. Define a new matrix $M$, which has $n$ rows and $2n$ columns as follows.

- For $j \in \{1, \ldots, n\}$, column $2j - 1$ of $M$ is $\text{almostcopy}_{i_j+1}^{\neq j}(N)$.
- For $j \in \{1, \ldots, n\}$, column $2j$ of $M$ is column $i_j + 1$ of $N$.

Figure 7 depicts the process of constructing the matrix $M$. Gray regions in the matrix $N$ indicate pairs of distinct values in a row $j$ of the consecutive columns $i_j$ and $i_j + 1$. We assume that the sequence $i_j$ is strictly increasing (the other case leads to a bottom-up parity alternating strategy matrix).

Now we show that $M \in \mathbb{P}_n$. The dimensions of the matrix $M$ are correct: it has $n$ rows and $2n$ columns. We now show that $M$ is a strategy matrix, which means that each row belongs to $\mathcal{V}_L$. For $j \in \{1, \ldots, n\}$, let us see how row $j$ of $M$
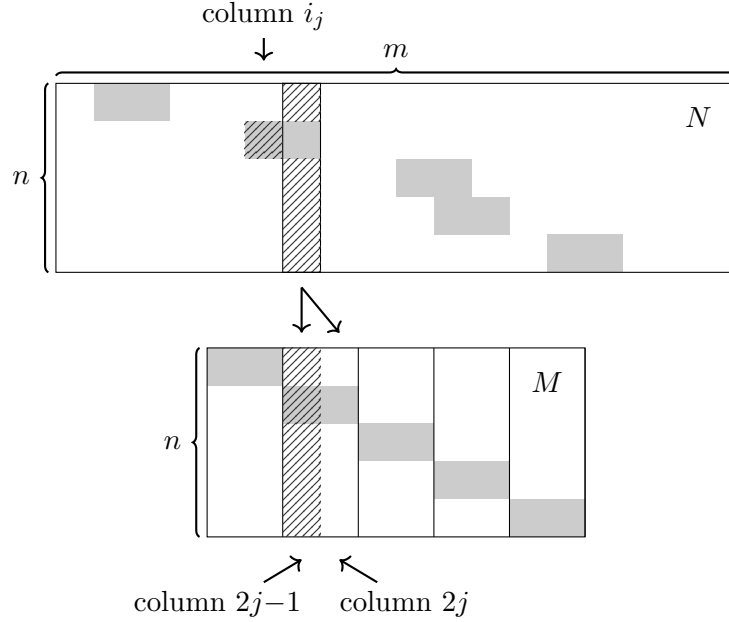
$$M[j,1], \ldots, M[j,2n]$$

Figure 7: Construction of the matrix $M$ from $N$. Column $2j - 1$ of $M$ equals almostcopy of column $i_j + 1$ of $N$ (i.e. the dashed part), while column $2j$ just equals column $i_j + 1$ of $N$.

depends on row $j$ of $N$

$$N[j, 1], \ldots, N[j, m].$$

By reading the definition of $M$, we see that the dependency is

- When $k \neq j$, then $M[j, 2k - 1] = M[j, 2k] = N[j, i_k + 1]$.
- When $k = j$, then $M[j, 2k - 1] = N[j, i_k]$ and $M[j, 2k] = N[j, i_k + 1]$.

It follows that row $j$ of $M$ is obtained from row $j$ of $N$ by eliminating some letters and duplicating some other letters. Since $\mathcal{V}_L$ is closed under eliminating and duplicating letters, and since $N$ was a strategy matrix, it follows that also $M$ is a strategy matrix.

By Lemma 5.30, for every $j \in \{1, \ldots, n\}$, the values of columns $2j - 1$ and $2j$ in $M$ are different. By the construction, columns $2j - 1$ and $2j$ in $M$ have the same entries, except for row $j$. So $M$ is parity alternating.

When the sequence $i_1, \ldots, i_n$ is strictly decreasing, the matrix $M$ is defined like for a strictly increasing sequence, except that the columns of $M$ are filled in not from left to right, but from right to left. Formally speaking:

- For $j \in \{1, \ldots, n\}$, column $2(n - j + 1) - 1$ of $M$ is almostcopy$_{i_j+1}^{\neq j}(N)$.
- For $j \in \{1, \ldots, n\}$, column $2(n - j + 1)$ of $M$ is column $i_j + 1$ of $N$.

The proof that $M$ belongs to $\mathbb{P}_n$ is the same as above. $\qquad \square$

5.4.2. *From strategy matrices to violation of* (5.1). So now we can do the second intermediary step. Let us assume that $\mathbb{P}_n$ is non-empty for any $n$. By the symmetry we can assume that for every $n$ there exists a top-down parity alternating matrix of size $n$ (the

other case is handled symmetrically): under this assumption we prove that Equation (5.1) is violated. To do that, we need to start from a parity alternating matrix of a large size, and by consecutive simplifications, construct a small matrix from which the violation can be easily extracted.

Take a strategy matrix $M$. Our aim is define a matrix obtained form $M$ by *merging* two consecutive rows $i$ and $i + 1$ of $M$. Let

$$(v_{1,1}, \ldots, v_{1,n}), \ldots, (v_{k,1}, \ldots, v_{k,n}) \in V_L^n.$$

be the rows in $M$. The merge operation removes rows

$$(v_{i,1}, \ldots, v_{i,n}) \quad \text{and} \quad (v_{(i+1),1}, \ldots, v_{(i+1),n})$$

and replaces them by the row

$$(v_{i,1} \cdot v_{(i+1),1}, \ldots, v_{i,n} \cdot v_{(i+1),n}),$$

which is the product of the two removed rows in the monoid $V_L^n$. Clearly, the result of the merging operation is also a strategy matrix with the same values of all the columns.

Now we define four *safe rules*, i.e. rules that preserve parity alternating strategy matrices.

**Lemma 5.32.** For every $M \in \mathbb{P}_n$ that is top-down and $i \in \{1, \ldots, n\}$, applying the following rules yields a parity alternating strategy matrix in $\mathbb{P}_{n-1}$:

- remove columns $2i - 1$ and $2i$ and then merge row $i$ with $i+1$;
- remove columns $2i - 1$ and $2i$ and then merge row $i-1$ with $i$;
- remove the first two columns and remove the first row;
- remove the last two columns and remove the last row.

The same holds for a bottom-up parity alternating matrices but then we count the columns in the reversed order.

*Proof.* Immediate by the definition of the rules.          □

Notice that $ux \neq uy$ implies that $x \neq y$, but $uvx \neq uvy$ does not imply that $ux \neq uy$, therefore we cannot safely remove columns $2i - 1$ and $2i$ and remove row $i$ except for $i = 1$ or $i = n$.

Consider a parity alternating strategy matrix $M \in \mathbb{P}_n$ and two indices $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, 2n\}$. We say that the entry $M[i,j]$ is *above diagonal* (resp. *below diagonal*) if:

- if $M$ is top-down then $2i$ must be smaller than $j$ (resp. $2i$ must be grater than $j + 1$),
- if $M$ is bottom-up parity alternating then $2(n - i)$ must be greater than $j - 1$ (resp. $2(n - i)$ must be smaller than $j - 2$).

The following picture depicts the regions above and below the diagonal of a top-down parity alternating matrix in $\mathbb{P}_6$.
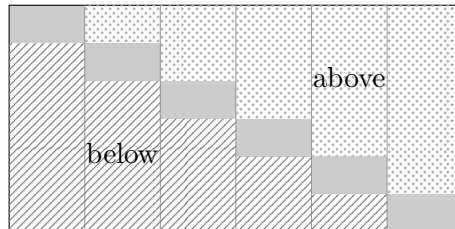
**Definition 5.33.** A parity alternating strategy matrix $M$ is upper (resp. lower) *idempotent* if there exists an idempotent $e \in V_L$ such that every entry above (resp. below) the diagonal of $M$ equals $e$. $M$ is called just *idempotent* if it is both upper and lower idempotent (possibly with two distinct idempotents $e, e' \in V_L$).

**Fact 5.34.** The safe rules preserve the fact that a given parity alternating matrix is upper (resp. lower) idempotent.

**Lemma 5.35.** For each $m \in \mathbb{N}$ there is some $n \in \mathbb{N}$ such that for any matrix in $\mathbb{P}_n$ there is a sequence of safe rules that yields a matrix $N \in \mathbb{P}_m$ that is upper (resp. lower) idempotent.

*Proof.* By the symmetry we will only deal with the upper idempotent case. The proof uses Ramsey Theorem for hypergraphs with edges of size 3. This theorem says that for every $m \in \mathbb{N}$ there exists a number $f(m)$ such that for any complete hypergraph with edges coloured over $V_L$, there exists a complete sub-hypergraph of size $m$ in which all edges share the same colour. We choose $n = f(m+1)$.

Fix a matrix $M \in \mathbb{P}_n$. Again by the symmetry we assume that $M$ is a top-down parity alternating matrix. Consider the hypergraph where the nodes are $\{1, \dots, n\}$ and an edge $\{i < j < k\}$ is coloured by the value obtained by multiplying, in the monoid $V_L$, the entries that appear in rows $i + 1, \dots, j$ of column $2k$. By the choice of $n$, we can apply Ramsey Theorem to this colouring and get a subset of size $m + 1$:

$$I = \{i_0 < i_1 < \dots < i_m\} \subseteq \{1, \dots, n\}$$

such that all the hyperedges on $I$ have the same colour, say $e \in V_L$. This colour must be an idempotent, i.e. it must satisfy $e = ee$. This situation is illustrated below, with a matrix in $\mathbb{P}_{10}$. The multiplication in $V_L$ of the entries in the regions marked by red rectangles always gives the idempotent $e$. The regions come in pairs, for columns $2i_\ell$ and $2i_\ell - 1$ but as the matrix is parity alternating, the entries in these columns agree (except row $i_\ell$).
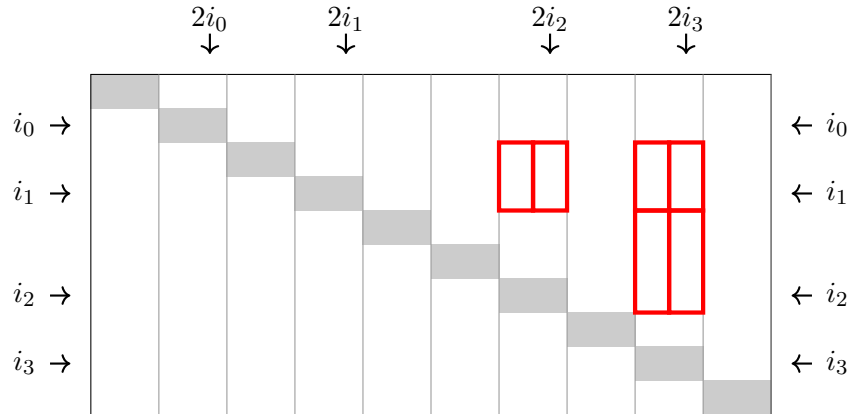


Now we apply a sequence of safe rules. First, we remove the first $i_0$ rows and first $2i_0$ columns. Then we remove the last $n - i_m$ rows and last $2(n - i_m)$ columns. We assume that the indices of rows and columns are shifted accordingly to the operations we perform; i.e. now $i_\ell \overset{\text{def}}{=} i_\ell - i_0$ for $\ell = 0, \dots, m$. After these operations, the matrix gets the following shape.
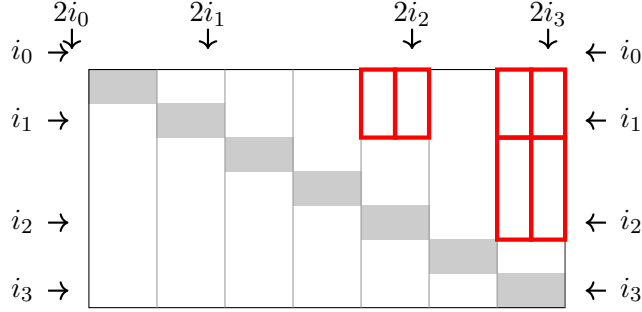
Now, for $\ell = 1, 2, \ldots, m$ we remove columns $2i_{\ell-1} + 1, \ldots, 2i_\ell - 2$ and merge rows $i_{\ell-1} + 1, \ldots, i_\ell$ into one row (with value $e$ in columns $2i_{\ell+1}, \ldots, 2i_m$). This way, we get the following matrix, which is upper idempotent.
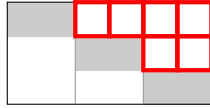


**Corollary 5.36.** For each $m \in \mathbb{N}$ there is some $n \in \mathbb{N}$ such that for any matrix in $\mathbb{P}_n$ there is a sequence of safe rules that yields a matrix $N \in \mathbb{P}_m$ that is idempotent.

*Proof.* Since the safe rules preserve the fact that a matrix is upper (resp. lower) idempotent, we can apply Lemma 5.35 twice, once to get an upper idempotent matrix and then to make it also lower idempotent. ☐

**Fact 5.37.** If $M$ is an idempotent parity alternating matrix, the operation that removes columns $2i$ and $2i - 1$ and removes row $i$ preserves the fact that the matrix is idempotent parity alternating.

*Proof.* If $i = 1$ or $i = n$ then the above operation is safe. For $1 < i < n$ we use the fact that the matrix is idempotent, so the values of columns $2i - 1$ and $2i$ depend only on the unique entry in that column which is neither above nor below the diagonal. The following picture depicts an idempotent parity alternating matrix in $\mathbb{P}_6$ where the upper idempotent is $u$ and the lower idempotent is $w$. Removing row 4 and columns 7 and 8 of that matrix preserves the fact that the matrix is idempotent parity alternating.



☐

**Lemma 5.38.** If $\mathbb{P}_n$ is non-empty for arbitrarily big $n$ then there are $u, w, x, y \in V_L$ such that $u$ and $w$ are idempotents and $\mathbb{P}_4$ contains the matrix

| $x$ | $y$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ |
|---|---|---|---|---|---|---|---|
| $w$ | $w$ | $x$ | $y$ | $u$ | $u$ | $u$ | $u$ |
| $w$ | $w$ | $w$ | $w$ | $x$ | $y$ | $u$ | $u$ |
| $w$ | $w$ | $w$ | $w$ | $w$ | $w$ | $x$ | $y$ |

*Proof.* By the hypothesis, we can apply Corollary 5.36 for $m = 4 * |V_L|^2$. Let $N$ be the resulting matrix. Each row of that matrix is of the form $w \cdots w \; \cdot x_i \cdot y_i \cdot \; u \cdots u$. Thus, there exists a pair $(x, y)$ that appears as $(x_i, y_i)$ in at least 4 rows. Using Fact 5.37 we can remove all the remaining rows (and the corresponding pairs of columns) from $N$ and the result has the above form. $\square$

We can conclude the proof of Proposition 5.22 by showing that under our assumptions Equation (5.1) is violated.

Let $M$ be the matrix described in Lemma 5.38. Let $\{u_1, \ldots, u_8\}$ be the values of all the columns in $M$. The matrix is in $\mathbb{P}_4$ so $u_3 \neq u_4$. Because $u$ and $w$ are idempotent,

$$u_3 = uxww = uxw = uuxw = u_5$$

For the same reason, $u_4 = u_6$. This is depicted in the picture below

| $x$ | $y$ | $u$ | $u$ | $u$ | $u$ | $u$ | $u$ |
|---|---|---|---|---|---|---|---|
| $w$ | $w$ | $x$ | $y$ | $u$ | $u$ | $u$ | $u$ |
| $w$ | $w$ | $w$ | $w$ | $x$ | $y$ | $u$ | $u$ |
| $w$ | $w$ | $w$ | $w$ | $w$ | $w$ | $x$ | $y$ |

$$\uparrow \; \uparrow \; \uparrow \; \uparrow$$
$$u_3 \; u_4 \; u_3 \; u_4$$

Since $u_3$ and $u_4$ are different, at least one of them is different than $uw$. Without loss of generality suppose that $u_3 \neq uw$. Because each row of the matrix belongs to $\mathcal{V}_L$ and $\mathcal{V}_L$ is closed under removing letters, it follows that

$$(w, x, u) \in \mathcal{V}_L.$$

This means that we have a violation of Equation (5.1), which requires that

$$uw = u^\sharp \cdot u \cdot w^\sharp = u^\sharp \cdot x \cdot w^\sharp = u_3.$$

This concludes the proof of Proposition 5.22.

5.5. **Case (C2) — limit alternation is unbounded.** In this section we assume that $\Sigma_L$ has unbounded root alternation and every subset $\Sigma' \subseteq \Sigma_L$ with unbounded root alternation has also unbounded limit alternation. Under that assumption we need to prove that Equation (5.2) is violated.

We start by constructing a finite directed graph $G_L$ (called the *strategy graph* of $L$) that represents very specific strategies of Alternator in the game on types. We then show that the above assumptions imply that the strategy graph needs to be *recursive*[5] — roughly speaking it contains complicated connected components. Then we finish the proof of Proposition 5.23 by showing that recursivity of the strategy graph gives rise to a violation of Equation (5.2).

---

[5]This notion, used previously in [BP12] and [FM14], has nothing to do with recursion theory, it speaks about the possibility to traverse certain edges in the graph and to go back to the beginning.

5.5.1. *Strategy graph.* Given a language $L$, the *strategy graph of $L$* (denoted $G_L$) is defined as follows. The set of nodes of $G_L$ is $\big(V_L \sqcup \{1_L\}\big) \times H_L$. There is an edge from a node $(v, h)$ to a node $(v', h')$ if there exist:

$$(u_1, u_2), (w_1, w_2) \in \mathcal{V}_L, \ z \in V_L \sqcup \{1_L\}$$

such that

$$h = v u_1 w_1^\infty \quad \text{and} \quad v' = v u_2 w_2^\sharp z.$$

Notice that the above definition does not invoke $h'$ (the value of $h'$ matters for a successive edge from $(v', h')$).

The following lemma provides a less explicit definition of edges in $G_L$, via the notion of *path-switching.*

**Definition 5.39.** Consider a tree type $h \in H_L$ and a pair of context types $v, v' \in V_L$. We say that the pair $(v, h)$ is *path-switching* into $v'$ if there exists a tree $t \in \mathrm{Tr}_A$ such that $v \cdot \alpha_L(t) = h$ and there exists an infinite path $\pi$ of $t$ such that if $D$ is a finite prefix of $t$ then $D$ can be completed into a context $C'$ with the port located on $\pi$ such that $v \cdot \alpha_L(C') \cdot z = v'$ for some $z \in V_L \sqcup \{1_L\}$.

**Lemma 5.40.** There exists an edge from $(v, h)$ to $(v', h')$ in $G_L$ if and only if $(v, h)$ is path-switching into $v'$.

*Proof.* First assume that there exists an edge from $(v, h)$ to $(v', h')$ in $G_L$ with $(u_1, u_2)$, $(w_1, w_2)$, $z$ witnessing that. Let $C_u$, $C_w$, $C_z$ be contexts of types $u_1$, $w_1$, $z$ respectively. Take $t \stackrel{\text{def}}{=} C_u \cdot C_w^\infty$. In that case $v \cdot \alpha_L(t) = h$ and let $\pi$ be the infinite path that contains all the ports of the contexts $C_u \cdot C_w^k$ for $k = 0, 1, \ldots$

Consider a finite prefix $D$ of $t$. Let $D'$ be an extension of $D$ that is also finite, contains exactly one port, and $D'$ is a prefix of $C_u \cdot C_w^{(\sharp \cdot k)}$ for some $k$.

By the definition, $D'$ can be written as

$$D = D_0 \cdot D_1 \cdot \ldots D_{\sharp \cdot k},$$

where $D_0, \ldots, D_{\sharp \cdot k}$ are finite prefixes of contexts (i.e. each of them has exactly one port); and moreover $D_0$ is a prefix of $C_u$ and all $D_1, \ldots, D_{\sharp \cdot k}$ are prefixes of $C_w$. By the assumption that $(u_1, u_2), (w_1, w_2) \in \mathcal{V}_L$ we know that we can extend all $D_0, D_1, \ldots, D_{\sharp \cdot k}$ into contexts $C_0, C_1, \ldots, C_{\sharp \cdot k}$ such that $\alpha_L(C_0) = u_2$ and $\alpha_L(C_i) = w_2$ for $i = 1, \ldots, \sharp \cdot k$. Take $C' = C_0 \cdot C_1 \cdot \ldots \cdot C_{\sharp \cdot k}$. Clearly the port of $C'$ is located on $\pi$. Notice that

$$\alpha_L(C') = u_2 \cdot w_2^{\sharp \cdot k} = u_2 \cdot w_2^\sharp.$$

Therefore, $v \cdot \alpha_L(C') \cdot z = v'$ by the assumption that there is an edge from $(v, h)$ to $(v', h')$.

Now we assume that $(v, h)$ is path-switching into $v'$ and we prove that there is an edge between $(v, h)$ and $(v', h')$ in $G_L$. This will be achieved by the Ramsey theorem. Consider a triple of nodes $x \prec y \prec \delta$ on $\pi$. Let $d = |\delta|$ be the depth of $\delta$ and let $D_d$ be the $(d+1)$-prefix of $t$ (i.e. $D_d$ is $t$ restricted to the subset $\{\mathtt{L}, \mathtt{R}\}^{\le d}$ of its domain). Let $C'_d$ be given from the assumption in Definition 5.39 for $D = D_d$.

Let $u_1, u_2, w_1, w_2$ be the $\alpha_L$-types of the following context zones:

- $u_1$ (resp. $u_2$) the type of the context zone rooted in $\epsilon$ with the port in $x$ of $t$ (resp. of $C'_d$);
- $w_1$ (resp. $w_2$) the type of the context zone rooted in $x$ with the port in $y$ of $t$ (resp. of $C'_d$).

Let $z$ be the type given by the assumption for $C'_d$.

Define $f(x, y, \delta)$ as the quintuple $(u_1, u_2, w_1, w_2, z) \in \left(V_L \sqcup \{1_L\}\right)^5$. Apply the Ramsey theorem for triples to $f$ to get an infinite set $P$ of nodes on $\pi$. We know that for all the triples from $P$ the function $f$ is constantly equal a fixed quintuple $(u_1, u_2, w_1, w_2, z)$. It is easy to see that $\alpha_L(t) = u_1 \cdot w_1^\infty$. Thus, $h = vu_1 w_1^\infty$. Similarly, the Ramsey theorem guarantees that $w_2 \cdot w_2 = w_2$ and therefore $v' = vu_2 w_2^\sharp z$. Thus, to know that there is an edge from $(v, h)$ to $(v'h,')$ in $G_L$ it is enough to prove that $(u_1, u_2)$ and $(w_1, w_2)$ are both elements of $\mathcal{V}_L$.

We will show that Alternator has a winning strategy in the game $\mathcal{V}(u_1, u_2)$ (the case of $\mathcal{V}(w_1, w_2)$ is analogous). Fix $x \prec y \in P$ and let $C_1$ be the context zone rooted in $\epsilon$ with the port in $x$ of $t$. Assume that Alternator plays $C_1$ as the first context in the game $\mathcal{V}(u_1, u_2)$. Consider a finite prefix $D$ that is played by Constrainer. Since $D$ is finite and $P$ is infinite, there exists $\delta \in P$ such that $d \overset{\text{def}}{=} |\delta|$ is greater than the depth of all the nodes in $D_1$. Let Alternator reply with $C'_d$. By the assumption on $d$, we know that $C'_d$ indeed extends $D$. By the choice of $P$ we know that $\alpha_L(C'_d) = u_2$. Thus, we have proved that Alternator wins $\mathcal{V}(u_1, u_2)$. $\qquad\square$

**Lemma 5.41.** The strategy graph $G_L$ is transitive.

*Proof.* Consider an edge between $(v, h)$ and $(v', h')$ witnessed by $(u_1, u_2)$, $(w_1, w_2)$, and $z$; and an edge between $(v', h')$ and $(v'', h'')$ witnessed by $(u'_1, u'_2)$, $(w'_1, w'_2)$, $z'$. Then $h = vu_1 w_1^\infty$ and $v'' = vu_2 w_2^\sharp\left(zu'_2(w'_2)^\sharp z'\right)$. It means that there is an edge between $(v, h)$ and $(v'', h'')$ with $(u_1, u_2), (w_1, w_2) \in \mathcal{V}_L$, and $z'' \overset{\text{def}}{=} zu'_2(w'_2)^\sharp z'$. $\qquad\square$

Recall that a strongly connected component of a directed graph $G$ is a set of nodes $X$ such that for each $x \neq x' \in X$ there exists a path in $G$ leading from $x$ to $x'$. If $G$ is transitive then this condition boils down to saying that there is a single edge from $x$ to $x'$.

**Definition 5.42** (*Recursive strategy graphs*)**.** We say that a strategy graph $G_L$ is *recursive* if there exists a strongly connected component of $G_L$ that contains two nodes $(v, h)$, $(v', h')$ with $h \neq h'$.

**Lemma 5.43.** If the strategy graph $G_L$ is recursive then Equation (5.2) is violated.

*Proof.* Assume contrarily that $G_L$ is recursive but Equation (5.2) holds. Consider a pair of nodes $(v, h)$ and $(v', h')$ with $h \neq h'$ that witness recursivity of $G_L$. By Lemma 5.41 there must exist edges in $G_L$ from $(v, h)$ to $(v', h')$ and back. Let $(u_1, u_2)$, $(w_1, w_2)$, $z$; and $(u'_1, u'_2)$, $(w'_1, w'_2)$, $z'$ witness the existence of these edges respectively.

Apply Equation (5.2) to obtain that:

$$\left(u_2(w_2)^\sharp \; zu'_2(w'_2)^\sharp z'\right)^\sharp u_1(w_1)^\infty = \left(u_2(w_2)^\sharp \; zu'_2(w'_2)^\sharp z'\right)^\infty$$

$$\left(u'_2(w'_2)^\sharp \; z'u_2(w_2)^\sharp z\right)^\sharp u'_1(w'_1)^\infty = \left(u'_2(w'_2)^\sharp \; z'u_2(w_2)^\sharp z\right)^\infty$$

Let $W = u_2(w_2)^\sharp z$ and $W' = u'_2(w'_2)^\sharp z'$. Then by the assumptions on the edges between $(v, h)$ and $(v', h')$ we get that $vW = v'$ and $v'W' = v$. Moreover, the above equations get the form

$$\left(WW'\right)^\sharp u_1(w_1)^\infty = \left(WW'\right)^\infty$$

$$\left(W'W\right)^\sharp u_2(w_2)^\infty = \left(W'W\right)^\infty$$

And therefore, by using the values of $h$, $h'$ and multiplying these equations by $v$ and $v'$ respectively we get:

$$h = v \cdot u_1(w_1)^\infty = v \cdot \left(WW'\right)^\sharp u_1(w_1)^\infty = v \cdot \left(WW'\right)^\infty$$

$$h' = v' \cdot u'_1(w'_1)^\infty = v' \cdot \left(W'W\right)^\sharp u'_1(w'_1)^\infty = v' \cdot \left(W'W\right)^\infty$$

Now since $h = v \cdot \left(WW'\right)^\infty = vW \cdot (W'W)^\infty = v' \cdot \left(W'W\right)^\infty = h'$ we obtain the contradiction as we assumed that $h \neq h'$. $\qquad\square$

5.5.2. *Constructing a path in $G_L$.* We will now use the assumption that Case (C2) holds to construct an infinite path in $G_L$ such that every two consecutive nodes on that path $(v, h)$, $(v', h')$ satisfy $h \neq h'$. Since $G_L$ is finite, such an infinite path witnesses that some strongly connected component of $G_L$ contains such two nodes and therefore $G_L$ is recursive. By Lemma 5.43 it means a violation of Equation (5.2) and the proof of Proposition 5.23 is finished.

The construction of the path will be inductive, preserving an invariant that the last node constructed on the path is *alternating*: a node $(v, h)$ in $G_L$ is *alternating* if $v \cdot \mathcal{H}_L$ contains words that begin with $h$ and have arbitrarily high alternation.

**Lemma 5.44.** $G_L$ contains at least one alternating node.

*Proof.* This is because $\mathcal{H}_L$ has unbounded alternation. Therefore, by Pigeon-hole Principle there exists some $h \in H_L$ such that $\mathcal{H}_L$ contains words that begin with $h$ that have arbitrarily high alternation. By the definition, this means that the node $(1_L, h)$ is alternating in $G_L$. $\qquad\square$

The inductive step of the construction will be based on the following lemma.

**Lemma 5.45.** If $(v, h)$ is an alternating node of $G_L$ and Case (C2) holds then there exists an edge in $G_L$ from $(v, h)$ to $(v', h')$ such that $h \neq h'$ and $(v', h')$ is also alternating.

The rest of the section is devoted to a proof of Lemma 5.45. Let $\Sigma_{(v,h)}$ be the set of all strategy trees $\sigma$ (with root sequences of the form $(h_1, \ldots, h_n)$) such that:

- $\sigma$ is locally optimal for $v$ (see Definition 5.16),
- $v \cdot h_1 = h$,
- the sequence $(vh_1, \ldots, vh_n)$ is alternating (i.e. every two consecutive values in the sequence are distinct, see Definition 4.27).

**Fact 5.46.** The set $\Sigma_{(v,h)}$ is a subset of $\Sigma_L$ that has unbounded both root and limit alternation.

*Proof.* Fact 5.17 implies that all strategy trees in $\Sigma_{(v,h)}$ belong also to $\Sigma_L$. By the fact that $(v, h)$ is alternating and by Lemma 5.18 we know that $\Sigma_{(v,h)}$ has unbounded root alternation. Therefore, Case (C2) guarantees that $\Sigma_{(v,h)}$ has also unbounded limit alternation. $\qquad\square$

Now observe that for each $g, g' \in H_L$ either $(g, g')^k \in \mathcal{H}_L$ for all $k \in \omega$, or there exists a unique $k_{g,g'} \in \omega$ such that $(g, g')^{k_{g,g'}} \in \mathcal{H}_L$ but $(g, g')^{1+k_{g,g'}} \notin \mathcal{H}_L$. Since the set $H_L$ is finite, there exists a number $K$ that is greater than all the defined numbers $k_{g,g'}$. Thus, the following fact holds.

**Fact 5.47.** If for some pair $g, g' \in H_L$ we have $(g, g')^K \in \mathcal{H}_L$ then $(g, g')^k \in \mathcal{H}_L$ for all $k \in \omega$.

Let $\ell \overset{\text{def}}{=} |H_L|^2 \cdot K$ and let $\sigma = (t, \sigma_1, \ldots, \sigma_n)$ be a strategy tree in $\Sigma_{(v,h)}$ that has limit alternation greater than $\ell$.

We will now construct a path $\pi$ in $t$ on which the high limit alternation of $\sigma$ is located. Let $Z$ be the set of nodes $z$ of $t$ such that the root alternation of $\sigma.z$ is greater than $\ell$. By Corollary 5.21 we know that $Z$ is prefix closed and by the fact that the limit alternation of $\sigma$ is greater than $\ell$ we know that $Z$ is infinite. Therefore, by König lemma we know that $Z$ contains an infinite path $\pi$.

**Lemma 5.48.** There exists an infinite set $P \subseteq \pi$ and two sequences $(h_1, \ldots, h_n) \in H_L^*$ and $(v_1, \ldots, v_n) \in V_L^*$ such that for all nodes $x \in P$:

- $\big(\sigma_1(x), \ldots, \sigma_n(x)\big) = (h_1, \ldots, h_n)$,
- $\big(\mathsf{val}(\sigma, X, 1), \ldots, \mathsf{val}(\sigma, X, n)\big) = (v_1, \ldots, v_n)$, where $X$ is the context zone with the root in $\epsilon$ and port in $x$.

Moreover, $(h_1, \ldots, h_n) \in \mathcal{H}_L$ and $(v_1, \ldots, v_n) \in \mathcal{V}_L$.

*Proof.* The choice of $P$ and the sequences $(h_1, \ldots, h_n) \in H_L^*$, $(v_1, \ldots, v_n) \in V_L^*$ is just by Pigeon-hole Principle. By the definition of a strategy tree we know that $(h_1, \ldots, h_n) \in \mathcal{H}_L$. By Fact 5.13 we know that also $(v_1, \ldots, v_n) \in \mathcal{V}_L$. $\qquad\square$

Since the alternation of $(h_1, \ldots, h_n)$ is greater than $\ell = |H_L|^2 \cdot K$, we know that there exists a pair $g \neq g' \in H_L$ and a set of indices $I \subseteq \{1, \ldots, n-1\}$ such that $|I| \geq K$ and for all $i \in I$ we have $h_i = g$ and $h_{i+1} = g'$. Fix as $i_0$ the minimal element of $I$. By closure of $\mathcal{H}_L$ under subwords, we know that $(g, g')^K \in \mathcal{H}_L$. By Fact 5.47 it implies that

$$(g, g')^k \in \mathcal{H}_L \qquad\qquad \text{for all } k \in \omega. \qquad\qquad (5.6)$$

Let $u = v_{i_0}$ and $u' = v_{i_0+1}$ where $i_0$ is the minimal element of $I$. We will finish the proof by showing the following three lemmas.

**Lemma 5.49.** The values $vu'g$ and $vu'g'$ are distinct.

*Proof.* Recall that $g = h_{i_0}$ and $g' = h_{i_0+1}$. Assume that the value $vu'g$ is equal to $vu'g'$. Consider a strategy tree $\sigma'$ that equals $\sigma$ except for the subtree $\sigma'_{i_0+1}.x$ where $x$ is the $\preceq$-minimal element of $X$. Over that subtree, let $\sigma'_{i_0+1}$ be equal to $\sigma_{i_0}$. Let $(h'_1, \ldots, h'_n)$ be the root sequence of $\sigma'$. We know that $v \cdot (h'_1, \ldots, h'_n) = (h_1, \ldots, h_n)$ and $\mathsf{dist}(\sigma_{i_0}, \sigma'_{i_0+1})$ is strictly smaller than $\mathsf{dist}(\sigma_{i_0}, \sigma_{i_0+1})$, what contradicts local minimality of $\sigma$ for $v$. $\qquad\square$

**Lemma 5.50.** The nodes $(vu', vu'g)$ and $(vu', vu'g')$ are both alternating in $G_L$.

*Proof.* By (5.6) and Fact 4.25 we know that $(vu'g, vu'g')^k \in \mathcal{H}_L$ for all $k \in \omega$. Moreover, Lemma 5.49 says that $vu'g \neq vu'g'$. $\qquad\square$

Now it remains to prove the following lemma.

**Lemma 5.51.** There exist edges in $G_L$ from $(v, h)$ to both $(vu', vu'g)$ and $(vu', vu'g')$.

*Proof.* Using Lemma 5.40 it is enough to show that $(v, h)$ is path-switching into $vu'$. Let $\sigma'$ be the strategy tree obtained from $\sigma$ by removing the first $i_0$ rounds: $\sigma' = (t, \sigma_{i_0+1}, \sigma_{i_0+2}, \ldots, \sigma_n)$ ($i_0$ is the minimal element of $I$).

Consider the tree $t$ from the strategy tree $\sigma$ and the path $\pi$. By the definition of $\sigma$ we know that $v \cdot \alpha_L(t) = h$. Let $D$ be a finite prefix of $t$. The strategy tree $\sigma'$ witnesses that $D$ can be extended into a context $C'$ with a port located on $\pi$ such that that $\alpha_L(C') = v_{i_0+1} = u'$. Therefore $v \cdot \alpha_L(C') \cdot 1_{V_L} = vu'$. this concludes the proof that $(v, h)$ is path-switching into $vu'$. $\square$

Thus, Lemma 5.49 implies that at least one of the nodes $(vu', vu'g)$ and $(vu', vu'g')$ has the second coordinate distinct than $h$, Lemma 5.50 implies that this node is alternating, and Lemma 5.51 implies that $G_L$ contains an edge from $(v, h)$ to that node. Thus, the proof of Lemma 5.45 is concluded.

## 6. Effective characterisation of $\mathbf{\Delta}_2^0$

This final section is devoted to the effective characterisation of the Borel class $\mathbf{\Delta}_2^0$ (that corresponds to the union of the first $\omega_1$ Wadge degrees). Decidability of the class $\mathbf{\Delta}_2^0$ can be actually obtained as a direct corollary of [CMS17]: in this work the authors proved that it is decidable if a regular tree language is in the Borel class $\mathbf{\Pi}_2^0$; since regular languages are closed under complement and $\mathbf{\Delta}_2^0 = \mathbf{\Pi}_2^0 \cap \mathbf{\Sigma}_2^0$, we automatically obtain that it is decidable if a regular tree language is in $\mathbf{\Delta}_2^0$. However, here we show that the algebraic approach developed in the previous sections covers the case of the Borel class $\mathbf{\Delta}_2^0$ as well. What we will prove is the following:

**Theorem 6.1.** A regular tree language $L$ belongs to the class $\mathbf{\Delta}_2^0$ if and only if its syntactic algebra satisfies Equation (5.2) from Theorem 5.1.

This theorem is the main result of [FM14]; unfortunately the proof provided there is wrong:

- Proposition 4 in [FM14] cites Lemma G.2 from [BP12] in a wrong way. The logical claim in Proposition F.2 is of the form: if there is a set $\Sigma$ of unbounded root alternation then there is a set $\Sigma'$ of unbounded limit alternation[6]. Lemma G.2 says that if Proposition F.2 is violated then the strategy graph is recursive. Logically, it takes the form: if there is a set $\Sigma$ of unbounded root alternation but no set $\Sigma'$ has unbounded limit alternation then the graph is recursive. The way Proposition 4 summarises that statement is: if there exists a set $\Sigma$ of unbounded limit alternation then the strategy graph is recursive. This statement does not follow from [BP12].
- The proof of Theorem 1 in [FM14] in the implication $(2) \Rightarrow (1)$, shows how to construct, given an infinite strategy tree $s^\infty$, a family of strategy trees of unbounded limit alternation. The first step of the proof is to construct a family $\mathcal{G}$ of strategy trees of finite duration but unbounded root alternation. Then, an invalid application of a compactness argument (to find the set $X$) shows that $\mathcal{G}$ has in fact unbounded limit alternation. Such a set $X$ doesn't need to exist, it can be the case that the limit alternation of $s$ is 2 but for each

---

[6] Unbounded limit alternation implies unbounded root alternation.

pair $k < k' \leq j$ there are infinitely many nodes $n$ in $s$ such that $\sigma_k(n) \neq \sigma_{k'}(n)$ — it is enough that the nodes for distinct values $k, k'$ lie on distinct infinite branches.

Additionally, if it was the case that every set of strategy trees of unbounded root alternation has unbounded limit alternation; Proposition F.1 would hold always, nevertheless of the assumption of Identity (2).

The proof we provide here follows the logical structure of [FM14], with the following differences:

- Instead of the wrong reference used in Proposition 4 of [FM14] we show recursivity of $G_L$ using a new Lemma 5.40 that characterises existence of edges in $G_L$.
- Instead of the statement about existence of the set $X$ from the proof of Theorem 1 in [FM14] we provide here a direct construction (Lemma 6.6) showing that a winning strategy of Alternator in $\mathcal{H}^\infty(L)$ must take a specific structure that fits the characterisation from Lemma 5.40

As observed in [FM14], the techniques used to characterise $\mathsf{BC}(\mathbf{\Sigma}_1^0)$ provide a big part of a proof of Theorem 6.1. First, Proposition 3.20 says that $L$ is $\mathbf{\Delta}_2^0$ if and only if Constrainer wins $\mathcal{H}^\infty(L)$. Corollary 5.5 says that if Equation (5.2) is violated then Alternator wins $\mathcal{H}^\infty(L)$. Thus, the "only if" part of Theorem 6.1 follows.

On the other hand, Lemma 5.43 says that if the strategy graph $G_L$ is recursive then Equation (5.2) is violated. It means that the only remaining statement to prove Theorem 6.1 is expressed by the following proposition.

**Proposition 6.2.** If Alternator wins $\mathcal{H}^\infty(L)$ then the strategy graph $G_L$ is recursive.

The rest of this section is devoted to a proof of the above proposition. During the proof we will inductively construct an infinite path $(v_n, h_n)_{n \in \omega}$ in the strategy graph such that the sequence $h_n$ alternates between $L$ and $L^c$. The invariant of our construction is expressed by the following definition, using the notions of quotients from Subsection 4.4.

**Definition 6.3.** Consider a node $(v, h)$ of the strategy graph $G_L$. We say that $(v, h)$ is *prolongable* if there exists a winning strategy $\sigma$ of Alternator in $\mathcal{H}^\infty\big(v^{-1}(L)\big)$ or in $\mathcal{H}^\infty\big(v^{-1}(L)^c\big)$ such that if $t$ is the tree played by $\sigma$ in the first round then $v \cdot \alpha_L(t) = h$.

We begin with an introduction of an object called *essential node* and a study of its properties.

6.1. **Essential nodes.** Let $(v, h)$ be a prolongable node, as witnessed by a strategy $\sigma$ and a tree $t$. By the symmetry we can assume that $\sigma$ is winning in $\mathcal{H}^\infty\big(v^{-1}(L)^c\big)$.

**Definition 6.4.** Consider a node $u \in \{\mathtt{L}, \mathtt{R}\}^d$ for $d > 0$. Let $p$ be the $d$-prefix of $t$ (i.e. $p \overset{\text{def}}{=} t{\restriction}\{\mathtt{L}, \mathtt{R}\}^{<d}$). We say that $u$ is *essential* if there exists a context $C$ of a type $w \in V_L$, such that the port of $C$ is in $u$, $C$ extends $p$, and we have $(vw)^{-1}(L) \notin \mathbf{\Delta}_2^0$.

The last condition implies that Alternator has a winning strategy in[7] $\mathcal{H}^\infty\big((vw)^{-1}(L)\big)$. Let $t'$ be a tree played by one of such winning strategies of Alternator and let $g = \alpha_L(t')$. Using the above notions we say that $u$ is *$(w, g)$-essential*.

---

[7]We have assumed that $\sigma$ is winning in $\mathcal{H}^\infty\big(v^{-1}(L)^c\big)$ and therefore we have no complement here; in the dual case $\sigma$ is winning in $\mathcal{H}^\infty\big(v^{-1}(L)\big)$ and here we consider a winning strategy in $\mathcal{H}^\infty\big((vw)^{-1}(L)^c\big)$.

**Fact 6.5.** If $C$ is a context then the function $t \mapsto C[t]$ is continuous. Therefore, if $w^{-1}(K) \notin \mathbf{\Delta}_2^0$ then also $K \notin \mathbf{\Delta}_2^0$. In particular, using (4.2) we obtain that for $v, w \in V_L \sqcup \{1_L\}$ if $(vw)^{-1}(L) \notin \mathbf{\Delta}_2^0$ then also $v^{-1}(L) \notin \mathbf{\Delta}_2^0$. It means that if $\epsilon \prec u \preceq u'$ and $u'$ is essential then also $u$ is essential.

The crucial part in the proof of Proposition 6.2 is expressed by the following lemma.

**Lemma 6.6.** If $(v, h)$ be a prolongable node, witnessed by a strategy $\sigma$ and a tree $t$, then for every $d > 0$ there exists an essential node $u$ in $t$ such that $|u| = d$.

The rest of this subsection is devoted to a proof of this lemma. Consider a number $d > 0$ and let $p$ be the $d$-prefix of $t$. We will find an essential node $u \in \{\mathtt{L}, \mathtt{R}\}^d$. Let $D$ be the multicontext obtained from $p$ by making all the nodes in $\{\mathtt{L}, \mathtt{R}\}^d$ ports; let $u_1, \ldots, u_N$ be the list of these ports ($N = 2^d$); and $U \stackrel{\text{def}}{=} D[*]$ be the basic open set defined by $p$. Finally, put

$$M \stackrel{\text{def}}{=} D^{-1}\big(v^{-1}(L)\big) \subseteq \mathrm{Tr}_A^N.$$

**Remark 6.7.** The fact whether $(t_1, \ldots, t_N) \in M$ depends only on the tuple of types $\big(\alpha_L(t_1), \ldots, \alpha_L(t_N)\big)$.

Since $(v, h)$ is prolongable and $U$ is a valid response of Constrainer to Alternator playing $t$ in $\mathcal{H}^\infty\big(v^{-1}(L)^{\mathrm{c}}\big)$, we know that Alternator has a winning strategy in $\mathcal{H}_U^\infty\big(v^{-1}(L)\big)$. This game is equivalent to the game $\mathcal{H}^\infty\big(M\big)$ played in the topological space $\mathrm{Tr}_A^N$ — the $N$-fold product of the space $\mathrm{Tr}_A$, with the induced product topology. Therefore, by Proposition 3.20 we know that $M \notin \mathbf{\Delta}_2^0$.

**Definition 6.8.** A *section* of $M \subseteq \mathrm{Tr}_A^N$ is any set of the form

$$\big\{t \in \mathrm{Tr}_A \mid (t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_N) \in M\big\}.$$

for $i \in \{1, \ldots, N\}$.

Due to Remark 6.7, taking $h_j = \alpha_L(t_j)$ for $j = 1, \ldots, N$, $j \neq i$, we can equivalently define the above section as

$$\big(h_1, \ldots, h_{i-1}, \Box, h_{i+1}, \ldots, h_N\big) \stackrel{\text{def}}{=}$$
$$\big\{t \in \mathrm{Tr}_A \mid h_1 \times \ldots \times h_{i-1} \times \{t\} \times h_{i+1} \times \ldots \times h_N \subseteq M\big\}. \tag{6.1}$$

Notice that the notation from (6.1) naturally extends to formulae with more than one $\Box$. Such sets are called *multisections*. The number of holes in a multisection is called its *dimension*, i.e. a multisection of dimension $n$ is a subset $\mathrm{Tr}_A^n$. There is one multisection of the maximal dimension $N$: $\big(\Box, \ldots, \Box\big) = M$.

**Lemma 6.9.** Each multisection can be obtained from sections by taking finite unions, finite intersections, and products.

*Proof.* The proof will be inductive on the dimension of the given multisection. For the sake of simplicity of notation we even consider multisections of dimension zero, i.e. expressions of the form $(h_1, \ldots, h_N)$ with no holes. Formally such an expression is either the empty set, or a set containing the empty tuple () (depending on whether $h_1 \times \ldots \times h_N \subseteq M$ or not).

A multisection of dimension 1 is just a section, so the claim follows. Consider a multisection of the form (this form is generic up to rearranging the coordinates):

$$N = \big(\Box, \Box, \ldots, \Box, h_{i+1}, h_{i+2}, \ldots, h_N\big).$$

We will prove that $N$ can be represented as in Lemma 6.9.

Consider a tuple $h_1, \ldots, h_{i-1}$. If $(h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N)$ is empty then let us put $S(h_1, \ldots, h_{i-1}) = \emptyset$ and otherwise let $S(h_1, \ldots, h_{i-1})$ be

$$\bigcap_{h_i \subseteq (h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N)} (\square, \ldots, \square, h_i, \ldots, h_N) \times (h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N).$$

Now consider the set $K$ defined as

$$\bigcup_{h_1, \ldots, h_{i-1}} S(h_1, \ldots, h_{i-1}).$$

By the inductive assumption the set $K$ can be obtained from sections by finite unions, finite intersections, and products. It remains to prove that $K = N$.

First consider a tuple of types $(g_1, g_2, \ldots, g_i)$ such that

$$g_1 \times \ldots \times g_i \subseteq N. \tag{6.2}$$

We will show that this product is contained in $S(g_1, \ldots, g_{i-1})$. First, by (6.2) we know that $g_i \subseteq (g_1, \ldots, g_{i-1}, \square, h_{i+1}, \ldots, h_N)$ and therefore the later set is not empty. Consider any value $h_i \subseteq (g_1, \ldots, g_{i-1}, \square, h_{i+1}, \ldots, h_N)$. By the assumption about the considered values $h_i$ we know that $g_1 \times \ldots \times g_{i-1} \subseteq (\square, \ldots, \square, h_i, \ldots, h_N)$. Therefore, $g_1 \times \ldots \times g_i$ is contained in the product $(\square, \ldots, \square, h_i, \ldots, h_N) \times (h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N)$.

Now consider the other implication: take a tuple of types $(g_1, g_2, \ldots, g_i)$ such that

$$g_1 \times \ldots \times g_i \subseteq S(h_1, \ldots, h_{i-1}), \tag{6.3}$$

for some choice of types $h_1, \ldots, h_{i-1}$. It means that there must exist a type $h_i$ contained in $(h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N)$, because the later set cannot be empty. Therefore, $g_i \subseteq (h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N)$ and it means that $g_i$ is among the values $h_i$ over which we take the intersection. Thus

$$g_1 \times \ldots \times g_i \subseteq (\square, \ldots, \square, g_i, \ldots, h_N) \times (h_1, \ldots, h_{i-1}, \square, h_{i+1}, \ldots, h_N).$$

In particular $g_1 \times \ldots \times g_{i-1} \subseteq (\square, \ldots, \square, g_i, \ldots, h_N)$ what implies that

$$g_1 \times \ldots \times g_i \subseteq N.$$

This concludes the proof that $N = K$ and thus the claim is proved. $\square$

**Corollary 6.10.** There exists a section of $M$ that is not $\mathbf{\Delta}_2^0$.

*Proof.* Assume that all the sections of $M$ are $\mathbf{\Delta}_2^0$. Since $\mathbf{\Delta}_2^0$ sets are closed under finite unions, finite intersections, and products, Lemma 6.9 implies that the multisection $(\square, \ldots, \square) = M$ is also $\mathbf{\Delta}_2^0$. A contradiction. $\square$

Let $\{t \in \mathrm{Tr}_A \mid (t_1, \ldots, t_{i-1}, t, t_{i+1}, \ldots, t_N) \in M\}$ be a section of $M$ that is not $\mathbf{\Delta}_2^0$. Put

$$C \stackrel{\text{def}}{=} D[t_1, \ldots, t_{i-1}, \square, t_{i+1}, \ldots, t_N].$$

By the assumption about the chosen section, we have $C^{-1}(v^{-1}(L)) \notin \mathbf{\Delta}_2^0$. Thus, by putting $w \stackrel{\text{def}}{=} \alpha_L(C)$ we know that $(vw)^{-1}(L) \notin \mathbf{\Delta}_2^0$ and therefore the context $C$ witnesses that $u_i$ is essential. This concludes the proof of Lemma 6.6.

6.2. **Constructing a path in $G_L$.** Lets fix a strategy $\sigma$ of Alternator in $\mathcal{H}^\infty(L)$ and let $t_0$ be the tree played by $\sigma$ in the first round. Let $v_0 = 1_L$ and $h_0 = \alpha_L(t_0)$. Directly from the definition we know that $(v_0, h_0)$ is prolongable. Therefore, to prove Proposition 6.2 it is enough to prove the following inductive lemma.

**Lemma 6.11.** If $(v, h)$ is prolongable then there exists a node $(v', h')$ in the strategy graph $G_L$ such that $h' \neq h$, $(v', h')$ is prolongable, and there is an edge from $(v, h)$ to $(v', h')$.

Lets fix a node $(v, h)$ that is prolongable, as witnessed by a strategy $\sigma$ and a tree $t$. By the symmetry we can assume that $\sigma$ is winning in $\mathcal{H}^\infty\big(v^{-1}(L)^c\big)$.

Since there are infinitely many essential nodes in $t$, and a prefix of an essential node is also essential (see Fact 6.5), there exists a path $\pi$ such that for all $\epsilon \prec u \prec \pi$ the node $u$ is essential. Notice that each of these nodes $u$ comes with at least one pair $(w_u, g_u) \in V_L \times H_L$ such that $u$ is $(w_u, g_u)$-essential, see Definition 6.4. Let $(w, g)$ be a pair that equals $(w_u, g_u)$ for infinitely many $u$. Let $v' = vw$ and $h' = vwg$. By the choice of $h$ and $h'$ we know that $h \subseteq L^c$ and $h' \subseteq L$ and therefore $h' \neq h$. Also, directly from the definition of a $(w, g)$-essential node we know that the node $(v', h')$ is prolongable.

Therefore, to conclude Lemma 6.11 it is enough check that $(v, h)$ is path-switching into $v'$ (see Lemma 5.40). Consider a finite prefix $D$ of $t$. There must exist an essential node $u \prec \pi$ such that $u \in \{\mathtt{L}, \mathtt{R}\}^d$ for some $d > 0$; $(w_u, g_u) = (w, g)$; and $D$ is a prefix of the $d$-prefix of $t$. Since $u$ is essential, $D$ can be completed into a context $C$ with the port located in $u$, such that $\alpha_L(C) = w$. Thus, $v \cdot \alpha_L(C) \cdot 1_L = v \cdot w = v'$. This fulfils the requirements of Definition 5.39 and therefore $G_L$ contains an edge from $(v, h)$ to $(v', h')$.

## 7. Conclusions

This paper utilises the syntactic algebra of a given regular language $L$ to understand the topological complexity of $L$. The main result (Theorem 5.1) says that the language is $\mathsf{BC}(\mathbf{\Sigma}_1^0)$ if and only if its syntactic algebra satisfies Identities (5.1) and (5.2).

The first equation speaks about the *branching* structure of alternations between the language and the complement. The combinatorial background of this equation is represented by the three claims from Subsection 5.4.1, which show that one can alternate using distinct infinite branches of a given tree.

The second equation is focused on alternations that continue along a single branch, see Lemma 5.40 for the explicit form of such an alternation.

As observed by Facchini and Michalewski [FM14], languages outside the class $\mathbf{\Delta}_2^0$ should admit the second kind of alternation. Although the original paper contained certain mistakes, the claim is correct (Theorem 6.1): a regular language is $\mathbf{\Delta}_2^0$ if and only if its syntactic algebra satisfies Identity (5.2).

7.1. **Limitations.** In general there is no proper algebraic framework for analysing all regular languages of infinite trees. The available algebras are either too weak [BI09a, BS13], too strong [Boj10], or not closed under homomorphic images (i.e. contain a hidden existential quantifier) [Blu11]. Therefore, effective characterisations based on algebraic approach are limited either to certain subclasses of languages as the input; or to simple classes that are being characterised.

In this paper the input is not restricted (i.e. the characterisation works for all regular languages as the input); but the characterised classes are low in the Borel hierarchy. It is

not surprising, as the structure of the considered algebras is quite weak, as expressed by the following remark.

**Remark 7.1.** For every non-trivial language $L_0 \subseteq \mathrm{Tr}_A$ the syntactic algebra of the language

$$\{t \in \mathrm{Tr}_A \mid \forall u\; \exists v \succeq u.\; t{\restriction}v \in L_0\}$$

is the same, with both $H$ and $V$ being two-element sets.

Notice that if $L_0 = \{t \in \mathrm{Tr}_{\{a,b\}} \mid t(\epsilon) = a\}$ then the above language belongs to $\mathbf{\Pi}^0_2$, while for $L_0 = \big\{t \in \mathrm{Tr}_{\{a,b,c\}} \mid$ if $t(\epsilon) = c$ then $t$ contains a branch with infinitely many $a\big\}$ the above language is non-Borel ($\mathbf{\Sigma}^1_1$-complete). This shows that the structure of the considered algebras is not strong enough to distinguish the complexity of languages right above the class $\mathbf{\Delta}^0_2$. To climb higher in the Borel hierarchy one should consider some class of algebras with richer structure; unfortunately there is no natural candidate for such a class at the moment. The known characterisations of classes higher in the hierarchy[8] [SW16, CMS17] are based directly on games instead of using algebras.

7.2. **Comparison with original papers.** Comparing to the original work [BP12], this work provides more detailed and polished proofs, more pictures, and certain minor glitches corrected. Moreover, the newly added Lemma 5.40 provides an explicit characterisation of the edges in the strategy graph $G_L$.

As discussed in Section 6, the logical structure of [FM14] is not sound. In this work we repair the gap by providing a direct proof of existence of certain edges in the strategy graph $G_L$. For that, we introduce a new concept of an *essential node* (see Definition 6.4) and a combinatorial lemma about sections of matrices, see Lemma 6.9.

7.3. **Further work.** There are still natural classes of languages within $\mathbf{\Delta}^0_2$ that await characterisations.

**Equations for finite levels.** In this work we remark that the finite levels of the difference hierarchy can be characterised using sentences of MSO. It remains open whether these finite levels correspond to equations (or their ordered variants) in the syntactic algebra of the language.

**Bounds for infinite levels.** In [DM07] it has been proved that if $L_1$ and $L_2$ are regular tree languages with Wadge rank $\alpha_1$ and $\alpha_2$ respectively, then we can build regular languages $L_1 \oplus L_2$ and $L_1 \otimes \omega$ with Wadge rank $\alpha_1 + \alpha_2$ and $\alpha_1 \times \omega$ respectively. Hence, every Wadge degree with Wadge rank below $\omega^\omega$ is inhabited by regular languages. In particular, there are examples of regular tree languages in $\mathbf{\Delta}^0_2 \setminus \mathrm{BC}(\mathbf{\Sigma}^0_1)$. However, as the Wadge hierarchy over $\mathbf{\Sigma}^0_1$ has length $\omega_1$ and there are only countably many regular languages, there must exist a bound on the levels of the Wadge hierarchy occupied by regular languages. The ordinal $\omega^\omega$ is a natural candidate for the upper bound, as all the examples of the regular languages inside $\mathbf{\Delta}^0_2$ exhaust exactly the first $\omega^\omega$ levels of the Wadge hierarchy.

---

[8]There are other characterisations where the input is restricted to languages with limited use of non-determinism, see [Mur08b, FMM16].

**Characterisations of transfinite levels.** Not only we don't know how many transfinite levels of the difference hierarchy are occupied, but there is no effective characterisation of any transfinite level known. Thus, one can ask for instance how to verify if the Wadge degree $\eta$ of a regular tree language verifies $\omega \leq \eta < \omega \cdot 2$.

**Higher in the hierarchy.** The operations of the algebras used in this paper seem to be suited exactly to the classes up to $\mathbf{\Delta}_2^0$. However, one might imagine enriching their algebraic structure just a bit in such a way to cover one more level of the Borel hierarchy. This motivates the following problem.

**Problem 7.2.** Is there a natural algebraic structure extending the algebras given in this paper which allow an equational characterisation of the Borel class $\mathbf{\Delta}_3^0$.

7.4. **Acknowledgements.** The authors would like to express their gratitude to Henryk Michalewski for a number of insightful discussions on the subject. Additionally, the authors than anonymous referees for their helpful comments and suggestions.

## References

[AC13]   Alessandro Andretta and Riccardo Camerlo. The descriptive set theory of the Lebesgue density theorem. *Advances in Mathematics*, 234:1–42, 2013.

[BI09a]   Mikołaj Bojańczyk and Tomasz Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *CONCUR*, pages 131–145, 2009.

[BI09b]   Mikołaj Bojańczyk and Tomasz Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *CONCUR*, pages 131–145, 2009.

[Blu11]   Achim Blumensath. Recognisability for algebras of infinite trees. *Theor. Comput. Sci.*, 412(29):3463–3486, 2011.

[Boj04]   Mikołaj Bojańczyk. A bounding quantifier. In *CSL*, pages 41–55, 2004.

[Boj10]   Mikołaj Bojańczyk. Algebra for trees. A draft version of a chapter that will appear in the AutomathA handbook, 2010.

[Boj15a]   Mikołaj Bojańczyk. Recognisable languages over monads. In *DLT*, pages 1–13, 2015.

[Boj15b]   Mikolaj Boja'nczyk. Recognisable languages over monads. *CoRR*, abs/1502.04898, 2015.

[BP12]   Mikołaj Bojańczyk and Thomas Place. Regular languages of infinite trees that are Boolean combinations of open sets. In *ICALP*, pages 104–115, 2012.

[BS13]   Marcin Bilkowski and Michał Skrzypczak. Unambiguity and uniformization problems on infinite trees. In *CSL*, volume 23 of *LIPIcs*, pages 81–100, 2013.

[BW08]   Mikołaj Bojańczyk and Igor Walukiewicz. Forest algebras. In *Logic and Automata*, pages 107–132, 2008.

[CKLV13]   Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, pages 215–230, 2013.

[CL08]   Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *ICALP (2)*, pages 398–409, 2008.

[CMS17]   Filippo Cavallari, Henryk Michalewski, and Michał Skrzypczak. A characterisation of Pi^0_2 regular tree languages. In *MFCS*, pages 56:1–56:14, 2017.

[DM07]   Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. *Fundamentals of computation theory*, 2007.

[Eng89]   Ryszard Engelking. *General topology*. Sigma series in pure mathematics. Heldermann Verlag, 1989.

[ES35]   Paul Erdös and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.

[FM14]   Alessandro Facchini and Henryk Michalewski. Deciding the Borel complexity of regular tree languages. In *CiE 2014*, pages 163–172, 2014.

[FMM16]   Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Trans. Comput. Log.*, 17(4):24:1–24:38, 2016.

[GS53]   David Gale and Frank M. Stewart. Infinite games with perfect information. In *Contributions to the theory of games*, volume 2 of *Annals of Mathematics Studies, no. 28*, pages 245–266. Princeton University Press, 1953.

[GTW02]   Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[Hig52]   Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, s3-2(1):326–336, 1952.

[Idz12]   Tomasz Idziaszek. *Algebraic methods in the theory of infinite trees*. PhD thesis, University of Warsaw, 2012. unpublished.

[ISB16]   Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory Comput. Syst.*, 58(4):614–663, 2016.

[Kec95]   Alexander Kechris. *Classical descriptive set theory*. Springer-Verlag, New York, 1995.

[KW02]   Ralf Küsters and Thomas Wilke. Deciding the first level of the mu-calculus alternation hierarchy. In *FST TCS 2002*, volume 2556 of *LNCS*, pages 241–252, 2002.

[Mar75]   Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[MRS14]   Luca Motto Ros and Philipp Schlicht. Lipschitz and uniformly continuous reducibilities on ultrametric polish spaces. *V. Brattka, H. Diener, and D. Spreen (Eds.), Logic, Computation, Hierarchies, Ontos Mathematical Logic*, 4:213–258, 2014.

[MRSS15]   Luca Motto Ros, Philipp Schlicht, and Victor Selivanov. Wadge-like reducibilities on arbitrary quasi-polish spaces. *Mathematical Structures in Computer Science*, 25:1705–1754, 2015.

[Mur08a]   Filip Murlak. *Effective topological hierarchies of recognizable tree languages*. PhD thesis, University of Warsaw, 2008.

[Mur08b]   Filip Murlak. The Wadge hierarchy of deterministic tree languages. *Logical Methods in Computer Science*, 4(4), 2008.

[NW03]   Damian Niwiński and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

[NW05]   Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput. Sci.*, 123:195–208, 2005.

[PP04]   Dominique Perrin and Jean-Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.

[PZ15]   Thomas Place and Marc Zeitoun. The tale of the quantifier alternation hierarchy of first-order logic over words. *SIGLOG News*, 2015.

[Rab69]   Michael Oser Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. of the American Math. Soc.*, 141:1–35, 1969.

[Sch65]   Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[Sei89]   Helmut Seidl. Deciding equivalence of finite tree automata. In *STACS*, pages 480–492, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[Sku93]   Jerzy Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

[SW16]   Michał Skrzypczak and Igor Walukiewicz. Deciding the topological complexity of Büchi languages. In *ICALP (2)*, pages 99:1–99:13, 2016.

[Tho96]   Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

[Wag79]   Klaus Wagner. On $\omega$-regular sets. *Information and Control*, 43(2):123–177, 1979.

[Wal02a]   Igor Walukiewicz. Deciding low levels of tree-automata hierarchy. In *Workshop on Logic, Language, Information and Computation*, volume 67 of *Electronic Notes in Theoretical Computer Science*, pages 61–75, 2002.

[Wal02b]   Igor Walukiewicz. Deciding low levels of tree-automata hierarchy. *Electr. Notes Theor. Comput. Sci.*, 67:61–75, 2002.

[Wil93]   Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *Int. J. Alg. Comput.*, 3:447–489, 1993.