EXPLORING THE TOPOLOGICAL ENTROPY OF FORMAL LANGUAGES

FLORIAN STARKE

ABSTRACT. We introduce the notions of *topological entropy* of a formal language and of a topological automaton. We show that the entropy function is surjective and bound the entropy of languages accepted by deterministic ε-free push-down automata with an arbitrary amount of stacks.

1. INTRODUCTION

A well established notion to measure the complexity of a dynamical system is *topological entropy*. It measures how chaotic or random a dynamical system is. A topological automaton contains a dynamical system. Using this we can define the complexity of a language to be the topological entropy of the minimal topological automaton accepting it.

Steinberg introduced the notion of a topolocial automaton in 2013 [4]. Then in 2016, Schneider and Borchmann used this notion to define the topological entropy of a formal language. They gave a characterization of the topological entropy of a formal language in terms of Myhill-Nerode congruence classes and determined the entropy of some example languages [3].

In this article we solve previously open problems in this field, and expand the variety of example languages. In particular, we show that the entropy function is surjective. We show that every language accepted by a deterministic ε-free counter automaton with an arbitrary amount of counters has zero entropy, which generalizes the fact that all regular languages have zero entropy. Furthermore, we give a finite upper bound for the entropy of languages accepted by deterministic ε-free push-down automata with an arbitrary amount of stacks, which shows that every deterministic ε-free context-free language has finite entropy. We determine the entropy of the Dyck languages, the deterministic palindrome language, and of some other new example languages. Among them is also a deterministic context-free language with infinite entropy.

This article is structured as follows. First we introduce the notion of a topological automaton and its topological entropy. We will have a brief discussion on the effect different encodings have on the entropy of a language in Section 3.1. This will motivate Section 3.2, where we will show that for every possible entropy there is also a language with that entropy. In Section 4 we will take a glimpse on the connection between topological entropy and complexity theory by calculating the entropy of **SAT** and looking at the effect padding has on the entropy. In Section 5 we will bound the entropy of languages accepted by certain kinds of counter automata and push-down automata. There we will also explore the connection between topological

2010 *Mathematics Subject Classification.* 68Q45.
Key words and phrases. Complexity of formal languages, Automata, Topological entropy.

entropy and the Chomsky hierarchy. In the end we will give an outline for future work.

2. Preliminaries

In this section we will give a brief introduction to the Myhill-Nerode congruence relation, topological automata, and their topological entropy. We will use the results from Schneider and Borchmann [3] to give a compact definition of the topological entropy of a formal language.

2.1. Myhill-Nerode Congruence Relation.

The Myhill-Nerode congruence relation is a very basic concept of formal languages, but as it is essential to this article we will give a short recapitulation. Let L be a formal language over some alphabet Σ. The *Myhill-Nerode right-congruence relation of* L, denoted by $\Theta(L)$, is

$$\Theta(L) = \{(u, v) \mid \forall w \in \Sigma^*.\ uw \in L \iff vw \in L\}.$$

For two words u and v we say that w *witnesses* $(u, v) \notin \Theta(L)$ if $uw \in L \not\Longleftrightarrow vw \in L$. For a word u, we call all words w with $uw \in L$ *positive witnesses of* u. Denote the set of all positive witnesses of u by P_u. Note that $[u] = [v]$ if and only if $P_u = P_v$.

If a language L has only finitely many congruence classes, then the minimal finite automaton accepting it can be constructed using $\Theta(L)$: take the set $\Sigma^*/\Theta(L)$ as states, where the final states are the congruence classes contained in L, $[\varepsilon]$ as initial state, and $([w], a) \mapsto [wa]$ as transition function.

2.2. Topological Automata and Topological Entropy.

Every deterministic finite automaton $\mathcal{A}$ contains a dynamical system, where the monoid Σ^* acts on the states of $\mathcal{A}$. In the following definition we generalize this idea to automata with infinitely many states.

Definition 2.1. A *topological automaton* is a 5 tuple $\mathcal{A} = (X, \Sigma, \delta, x_0, F)$ where

- X is a compact Hausdorff space (the *states*),
- Σ is an alphabet (the *input alphabet*),
- $\delta\colon X \times \Sigma^* \to X$ is a continuous action of Σ^* on X (the *transition function*),
- x_0 is an element from X (the *initial state*), and
- F is a clopen, i.e., closed and open, subset of X (the *final states*).

The language accepted by $\mathcal{A}$ is

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \delta(x_0, w) \in F\}.$$

We can define the topological entropy of $\mathcal{A}$ as the entropy of the underlying dynamical system but we do not want to introduce all the necessary notions and therefore we will use an equivalent definition.

Let $L = L(\mathcal{A})$, and $E \subseteq \Sigma^*$. We define the following two equivalence relations

$$\Theta_E(L) = \{(u, v) \mid \forall w \in E.\ uw \in L \iff vw \in L\} \text{ and}$$
$$\Lambda_E(\mathcal{A}) = \{(x, y) \mid \forall w \in E.\ \delta(x, w) \in F \iff \delta(y, w) \in F\}.$$

The relation $\Theta_E(L)$ is an approximation of the Myhill Nerode congruence relation of L in the sense that it allows only words from E as witnesses. Hence if we choose $E = \Sigma^*$, then $\Theta_{\Sigma^*}(L)$ is the Myhill Nerode congruence relation of L. The second relation $\Lambda_E(\mathcal{A})$ is the counterpart of $\Theta_E(L)$ for the states of $\mathcal{A}$.

For $n \in \mathbb{N}$ we denote

$$\Theta_{\Sigma^{(n)}}(L) \text{ by } \Theta_n(L) \text{ and } \Lambda_{\Sigma^{(n)}}(\mathcal{A}) \text{ by } \Lambda_n(\mathcal{A}).$$

We will write $P_u^{(n)}$ (and P_u^n) for the positive witnesses of u with length at most n (with length exactly n). Schneider and Borchmann showed (Lemma 3.6 from [3]) that if $\mathcal{A}$ is a topological automaton, where the reachable states are dense in the set of all states. Then the *topological entropy* $\eta(\mathcal{A})$ of $\mathcal{A}$ satisfies:

$$\eta(\mathcal{A}) = \limsup_{n\to\infty} \frac{\log_2(\operatorname{ind} \Lambda_n(\mathcal{A}))}{n}.$$

Note that since $\Sigma^{(n)}$ is finite the index of $\Lambda_n(L)$ is also finite. Since from every topological automaton we can obtain a topological automaton accepting the same language, where the reachable states are dense in the set of all states, we will use the above identity as a definition. Analogously, we define the *topological entropy* of a language:

$$\eta(L) = \limsup_{n\to\infty} \frac{\log_2(\operatorname{ind} \Theta_n(L))}{n}.$$

Furthermore, Schneider and Borchmann give us the following connection between the entropy of an automaton and the entropy of its language.

Theorem 2.2 (Theorem 3.10 from [3])**.** *Let L be a language. Then*

$$\eta(L) = \min\{\eta(\mathcal{A}) \mid \mathcal{A} \text{ is a topological automaton with } L(\mathcal{A}) = L\}.$$

In particular, for every language L there also exists a topological automaton $\mathcal{A}$ such that $\eta(L) = \eta(\mathcal{A})$.

3. Encodings and Surjectivity

In this section we will first discuss how encodings effect the entropy of a language. In Section 3.2 we will solve the previously open problem of whether the entropy function is surjective by constructing a language for every possible entropy. Finally, we will discuss the entropy of languages over unary alphabets.

3.1. Encoding.

We motivate this subsection with an example.

Example 3.1. An example considered in [3] is the *Dyck language with k sorts of parenthesis*, which consists of all balanced strings over $\{(_1,)_1, \ldots, (_k,)_k\}$. More generally, let Γ be an alphabet and $\overline{\Gamma} = \{\overline{a} \mid a \in \Gamma\}$. Then $\overline{}\colon \Gamma \to \overline{\Gamma}$ is a bijection. Now the *Dyck language over* Γ, denoted by $\mathbf{Dyck}_\Gamma$, is the set of all words u such that successively replacing $a\overline{a}$ in u by ε results in ε.

Later in Lemma 6.3 we will show that $\eta(\mathbf{Dyck}_\Gamma) = \log_2 |\Gamma|$. This shows that there are Dyck languages with arbitrarily high entropy. But what happens with the entropy if we encode all Dyck languages over a two element alphabet? In this section we will give upper and lower bounds for the entropy of an encoded language. Then we will apply these results to the Dyck languages, to show that their entropy is bounded if we encode them over a fixed alphabet.

Definition 3.2. An encoding of Σ over Γ is a mapping $\operatorname{enc}\colon \Sigma \to \Gamma^+$ with the prefix property, i.e., there is no word in the image of enc which is a prefix of another word in the image.

The prefix property is necessary to ensure the invertability of the encoding.

Lemma 3.3. *Let L be a language over Σ and* $\mathrm{enc}\colon \Sigma \to \Gamma^+$ *an encoding of Σ over Γ. Then* $\frac{\eta(L)}{k_1} \leq \eta(\mathrm{enc}(L)) \leq \frac{\eta(L)}{k_2}$*, where* $k_1 = \max\{|u| \mid u \in \mathrm{im}(\mathrm{enc})\}$ *and* $k_2 = \min\{|u| \mid u \in \mathrm{im}(\mathrm{enc})\}$.

Proof. Note that ε is not in the image of enc, and as a consequence k_2 is at least 1. We show the following two inequalities:

$$\operatorname{ind}\Theta_{n\cdot k_1}(\mathrm{enc}(L)) \geq \operatorname{ind}\Theta_n(L) \tag{$*$}$$

$$\operatorname{ind}\Theta_{n\cdot k_2}(\mathrm{enc}(L)) \leq |\mathit{Pre}| \cdot \operatorname{ind}\Theta_n(L) + 1 \tag{$**$}$$

where *Pre* contains all real prefixes of words in the image of enc.

For the first inequality consider the map

$$[u] \mapsto \{[\mathrm{enc}(u')] \mid u' \in [u]\}.$$

Note that injectivity of this map does not suffice to show $(*)$. We need to show that the images of two different classes $[u_1]$ and $[u_2]$ are disjoint. Let $[\mathrm{enc}(u_1')]$ and $[\mathrm{enc}(u_2')]$ be elements from sets corresponding to $[u_1]$ and $[u_2]$, respectively. Because $u_1' \in [u_1]$ and $u_2' \in [u_2]$ there is a w that witnesses $(u_1', u_2') \notin \Theta_n(L)$ and $|\mathrm{enc}(w)| \leq k_1 \cdot |w| \leq k_1 \cdot n$. Hence $\mathrm{enc}(w)$ is a witness for $(\mathrm{enc}(u_1'), \mathrm{enc}(u_2')) \notin \Theta_{n\cdot k_1}(\mathrm{enc}(L))$.

For $(**)$ we will show that the following map is almost surjective

$$\Sigma^*/\Theta_n(L) \times \mathit{Pre} \to \Gamma^*/\Theta_{n\cdot k_2}(\mathrm{enc}(L))$$
$$([u], v) \mapsto [\mathrm{enc}(u) \cdot v].$$

Note that all words $u \in \Gamma^*$ with $P_u = \emptyset$ lie in the same class. Let $[u]$ be a class of $\Theta_{n\cdot k_2}(\mathrm{enc}(L))$ such that u has at least one positive witness w. Then uw is the encoding of some word in Σ^*. Therefore there are $u' \in \Sigma^*$ and $v \in \mathit{Pre}$ such that $\mathrm{enc}(u')v = u$. As a consequence, $([u'], v)$ lies in the preimage of $[u]$. From this $(**)$ follows.

Using these inequalities we can now infer

$$k_1 \cdot \eta(\mathrm{enc}(L)) = k_1 \cdot \limsup_{n\to\infty} \frac{\log_2 \operatorname{ind}\Theta_{n\cdot k_1}(\mathrm{enc}(L))}{n \cdot k_1} \overset{(*)}{\geq} \limsup_{n\to\infty} \frac{\log_2(\operatorname{ind}\Theta_n(L))}{n} = \eta(L).$$

The other direction follows similarly

$$k_2 \cdot \eta(\mathrm{enc}(L)) \overset{(**)}{\leq} \limsup_{n\to\infty} \frac{\log_2(|\Gamma^{(k_2-1)}| \cdot \operatorname{ind}\Theta_n(L) + 1)}{n} = \eta(L).$$

This concludes the proof. ☐

Note that if a language has infinite entropy, then every encoding of this language also has infinite entropy. On the other hand if $\eta(L)$ is zero, then the entropy of $\mathrm{enc}(L)$ will also be zero. If all encoded letters have the same length we obtain the following corollary.

Corollary 3.4. *Let L be a language over Σ, $k \geq 1$, and* $\mathrm{enc}\colon \Sigma \to \Gamma^k$ *an encoding of Σ over Γ. Then* $\eta(\mathrm{enc}(L)) = \frac{\eta(L)}{k}$.

We can encode every language over Σ over a two element alphabet Γ with an encoding $\mathrm{enc}\colon \Sigma \to \Gamma^k$ for some $k \in \mathbb{N}$. Note that the minimal possible value of k for which we can define such an encoding is $\lceil \log_2 |\Sigma| \rceil$. Hence we call the encoding enc *efficient* if $k = \lceil \log_2 |\Sigma| \rceil$.

Corollary 3.5. *If L is a language over Σ and* $\mathrm{enc}\colon \Sigma \to \Gamma^k$ *efficiently encodes Σ over a two letter alphabet Γ, then* $\eta(\mathrm{enc}(L)) = \frac{\eta(L)}{\lceil \log_2 |\Sigma| \rceil}$.

Now we can answer our initial question: What happens if we encode the Dyck languages over a two element alphabet?

Corollary 3.6. *Let* $\mathrm{enc}\colon \Sigma \to \Delta^k$ *be an efficient encoding of* $\Sigma = \Gamma \cup \overline{\Gamma}$ *over a two letter alphabet Δ. Then*

$$\eta(\mathrm{enc}(\boldsymbol{Dyck}_\Gamma)) = \frac{\log_2 |\Gamma|}{\lceil \log_2 |\Gamma| \rceil + 1}.$$

For example, if we encode $\mathbf{Dyck}_{\{(_1,(_2\}}$ over $\{0,1\}$ with

$$(_1 \mapsto 00 \qquad)_1 \mapsto 10$$
$$(_2 \mapsto 01 \qquad)_2 \mapsto 11$$

then the encoded language has entropy $\frac{1}{2}$.

This corollary implies that any encoded Dyck language has entropy less than one. As this argument also affects the other example presented in [3], namely the palindrome languages, we now lack examples for languages over a two element alphabet with an entropy in $(2, \infty)$. We will remedy this in the next section by constructing a language for any given entropy.

3.2. **Every Entropy has its Language.** In this subsection we will show that the entropy function $\eta\colon \mathcal{P}(\Sigma^*) \to [0, \infty]$ is surjective if Σ contains at least two elements. For now we fix the alphabet $\Sigma = \{0, 1\}$.

Let us call a sequence $(k_n)_{n\in\mathbb{N}}$ of natural numbers *suitable* if it is monotone increasing, $k_0 = 1$, and $k_n \leq 2 \cdot k_{n-1}$. Note that in this case $k_n \leq 2^n$. Our goal is to construct a language L such that $\mathrm{ind}\,\Theta_n(L)$ is about 2^{k_n}. The construction generalizes an idea from Schneider and Borchmann (Example 4.12 in [3]). We define for all $n \in \mathbb{N}$

$$\begin{aligned} \varphi_n\colon \Sigma^{2^n} &\to \mathcal{P}(\{1, \dots, k_n\}) \\ a_1 \dots a_{2^n} &\mapsto \{i \in \{1, \dots, k_n\} \mid a_i = 1\}. \end{aligned}$$

Note that $|\operatorname{im} \varphi_n| = 2^{k_n}$. For $n \in \mathbb{N}$ we define a function $f_n\colon \Sigma^n \to \{1, \dots, k_n\}$ recursively. Let us fix $f_0(\varepsilon) = 1$. For $n \in \mathbb{N}$ define

$$f_{n+1}(0u) = f_n(u) \qquad f_{n+1}(1u) = \begin{cases} f_n(u) & \text{if } f_n(u) + k_n > k_{n+1} \\ f_n(u) + k_n & \text{if } f_n(u) + k_n \leq k_{n+1}. \end{cases}$$

We construct a language L in the following way:

$$L = \left\{ uv \;\middle|\; |v| = 2^{|u|}, f_{|u|}(v) \in \varphi_{|u|}(u) \right\}.$$

The idea of f_n is that if $k_{n+1} < 2 \cdot k_n$, then we do not need all possible new words to distinguish all elements in $\operatorname{im} \varphi_{n+1}$. This is because the number of words doubles since every word u is split into $0u$ and $1u$. As a consequence, depending on the way we look at it, f_n fuses some of these words together, or only splits as much words as needed.

Now it is time to take a closer look at the properties of φ_n and f_n. Clearly, φ_n and f_n are surjective. More interesting are the following two properties:

(P1) the following are equivalent for all $n \in \mathbb{N}$ and all $u, v \in \Sigma^n$

(a) $f_n(u) = f_n(v)$
(b) $f_{n+1}(au) = f_{n+1}(av)$ for all $a \in \Sigma$
(c) $f_{n+1}(au) = f_{n+1}(av)$ for some $a \in \Sigma$

(P2) the following are equivalent for all $n, k \in \mathbb{N}$, $u_1, u_2 \in \Sigma^k$ with $k \leq 2^n$:
(a) $\varphi_n(u_1 v) = \varphi_n(u_2 v)$ for some $v \in \Sigma^{2^n - k}$
(b) $\varphi_n(u_1 v) = \varphi_n(u_2 v)$ for all $v \in \Sigma^{2^n - k}$.

Let us check that these properties hold. For (P1) consider first (a) $\Rightarrow$ (b). We either have

$$\begin{aligned} f_{n+1}(au) &= f_n(u) &&= f_n(v) &&= f_{n+1}(av), \text{ or} \\ f_{n+1}(au) &= f_n(u) + k_n &&= f_n(v) + k_n &&= f_{n+1}(av) \end{aligned}$$

for all $a \in \Sigma$. The implication from (b) to (c) is trivial. For the last implication consider the case $a = 0$, then $f_{n+1}(0u) = f_{n+1}(0v)$ implies $f_n(u) = f_n(v)$ by definition. If $f_{n+1}(1u) = f_{n+1}(1v)$, then either $f_{n+1}(1u) + k_n > k_{n+1}$ and $f_{n+1}(1u) = f_n(u) = f_n(v) = f_{n+1}(1v)$ or $f_{n+1}(1u) + k_n \leq k_{n+1}$, which again implies $f_n(u) = f_n(v)$.

The property (P2) is clear from the definition of φ_n. If we assume an additional property on k_n, then we are able to bound $\operatorname{ind} \Theta_n(L)$ for large n.

Lemma 3.7. *If k_n grows sufficiently fast, i.e., there is an $N \in \mathbb{N}$ such that $n \cdot k_n \leq k_{2^n}$ for all $n \geq N$, then*

$$2^{k_n} \leq \operatorname{ind} \Theta_n(L) \leq 3 \cdot (n+1)^2 \cdot 2^{k_n} \text{ for all } n \geq 2^N.$$

Proof. Let $n \in \mathbb{N}$ with $n \geq 2^N$. First we will determine the number of classes of $\Theta_n(L)$ generated by words of length 2^n. For $u, v \in \Sigma^{2^n}$, if $\varphi_n(u) \neq \varphi_n(v)$, then fix some element $k \in \varphi_n(u) \triangle \varphi_n(v)$. Since f_n is surjective there is a $w \in \Sigma^n$ such that $f_n(w) = k$, and the word w witnesses the fact that u and v are not in the same class. Vice versa, if $\varphi_n(u) = \varphi_n(v)$, then $(u, v) \in \Theta_n(L)$. Now the lower bound immediately follows from

$$\operatorname{ind} \Theta_n(L) \geq \left| \{[u] \mid u \in \Sigma^{2^n}\} \right| = |\operatorname{im} \varphi_n| = 2^{k_n}.$$

For the upper bound there is much more work to do. We will look at the different types of words and bound the number of equivalence classes generated by words of each type separately.

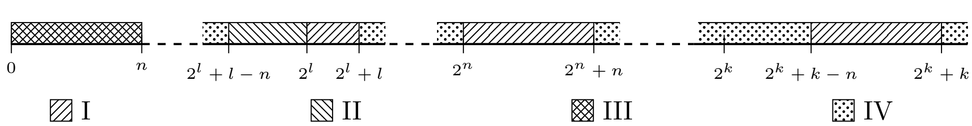

Figure 1. The four different types of words

- Words of type I are of the form uv where $u \in \Sigma^{2^k}$ and $v \in \Sigma^{(k)}$ for some $k \geq \log_2 n$. All words $v' \in \Sigma^{(n)}$ for which $uvv' \in L$ have to be of length $k - |v|$. Recall $uvv' \in L$ iff $f_k(vv') \in \varphi_k(u)$. Hence the witnesses can only be used to determine $\varphi_k(u)$. Note that the decomposition into u and v is unique.

- Every word u of type II is in Σ^{2^l-k} for some l and k with $\log_2 n \leq l \leq n$ and $k+l \leq n$. All words $w \in \Sigma^{(n)}$ with $uw \in L$ are of the form $u'v$ with $u' \in \Sigma^k$ and $v \in \Sigma^l$. The difference to words of type I is that now $\varphi_k(uu')$ depends on the choice of the witness. Note that this could potentially lead to a lot of new equivalence classes.
- The words of type III are all words with length at most $n-1$. Some of these words u are short enough such that there can be positive witnesses of u in $\Sigma^{(n)}$ with different lengths.
- Finally, for all words u of type IV we have $uw \notin L$ for all $w \in \Sigma^{(n)}$. Hence these words are all in the same equivalence class.

We denote number of equivalence classes the words of type I generate, i.e., $|\{[u] \mid u \text{ is of type I}\}|$, by B_{I}. Analogously we define B_{II}, B_{III}, and B_{IV}. Since any word in Σ^* is of one of the four types we have that

$$\operatorname{ind}\Theta_n(L) \leq B_{\mathrm{I}} + B_{\mathrm{II}} + B_{\mathrm{III}} + B_{\mathrm{IV}}.$$

We have already noted that $B_{\mathrm{IV}} = 1$. Before we determine an upper bound for B_{I}, B_{II}, and B_{III} we will look at sets of the form $\Sigma^{2^n} \cdot \{0\}^k$ and determine an upper bound for the size of $\{[u] \mid u \in \Sigma^{2^n} \cdot 0^k\}$ for $k \in \{0, \ldots, n\}$.

Let $u_1, u_2 \in \Sigma^{2^n} \cdot 0^k$. Then $u_1 = u_1'v$, $u_2 = u_2'v$ for some $u_1', u_2' \in \Sigma^{2^n}$ and $v = 0^k$. If $(u_1, u_2) \notin \Theta_n(L)$, then there is some $w \in \Sigma^{n-k}$ which bares witness to this fact. Now vw witnesses $(u_1', u_2') \notin \Theta_n(L)$. Hence the words in $\Sigma^{2^n} \cdot 0^k$ give rise to at most as many classes as the words in Σ^{2^n}, which, as we have already shown, decompose into 2^{k_n} classes. Therefore we conclude

$$(*) \qquad \left|\{[u] \mid u \in \Sigma^{2^n} \cdot 0^k\}\right| \leq 2^{k_n} \text{ for all } n \in \mathbb{N},\, k \in \{0, \ldots, n\}.$$

To find an upper bound for B_{I} we will show that every word of type I is already in the same class as some word in $\Sigma^{2^n} \cdot \{0\}^{(n)}$. Let uv be a word of type I with $u \in \Sigma^{2^k}$ and $v \in \Sigma^{(k)}$. We have already noticed that then $P_{uv}^{(n)} \subseteq \Sigma^{k-|v|}$. Define $v' = 0^{n-k+|v|}$. Since $n - k + |v| + k - |v| = n$ and φ_n is surjective there is an u' such that $\varphi_n(u') = f_n(v' P_{uv}^{(n)})$. Note that $u'v' \in \Sigma^{2^n} \cdot \{0\}^{(n)}$. Our next goal is to show that uv and $u'v'$ are in the same equivalence class, i.e.,

$$w \in P_{uv}^{(n)} \iff f_n(v'w) \in \varphi_n(u') \overset{\text{Def.}}{\iff} w \in P_{u'v'}^{(n)}.$$

The $\Rightarrow$ direction is straightforward. If $w \in P_{uv}^{(n)}$, then $v'w \in v'P_{uv}^{(n)}$ and therefore $f_n(v'w) \in f_n(v'P_{uv}^{(n)}) = \varphi_n(u')$. For the $\Leftarrow$ direction take some word w with $f_n(v'w) \in \varphi_n(u')$. By choice of u' there is a $w' \in P_{uv}^{(n)}$ such that $f_n(v'w) = f_n(v'w')$. Then

$$\begin{aligned} &(|v'| \text{ times (P1)}) && f_n(v'w) = f_n(v'w') \iff f_{k-|v|}(w) = f_{k-|v|}(w') \\ &(|v| \text{ times (P1)}) && \iff f_k(vw) = f_k(vw'). \end{aligned}$$

Because $w' \in P_{uv}^{(n)}$ implies $f_k(vw) = f_k(vw') \in \varphi_k(u)$ we conclude $w \in P_{uv}^{(n)}$. Thus uv is in the same class as $u'v'$.

Consequently, we can finally give an upper bound for B_{I}:

$$B_{\mathrm{I}} \leq \left|\{[w] \mid w \in \Sigma^{2^n} \cdot \{0\}^{(n)}\}\right| \leq \sum_{k=0}^{n} \left|\{[w] \mid w \in \Sigma^{2^n} \cdot 0^k\}\right| \overset{(*)}{\leq} (n+1) \cdot 2^{k_n}.$$

For B_{II} we will consider all words of type II with the same length $2^l - k$ for some $l, k \in \mathbb{N}$ with $l+k \leq n$. For fixed l and k we will bound the size of $\{P_u^{(n)} \mid u \in \Sigma^{2^l-k}\}$ by 2^{k_l}. To do this let $u' \in \Sigma^k$. We show that

$$\begin{aligned} \{P_u^{(n)} \mid u \in \Sigma^{2^l-k}\} &\to \operatorname{im} \varphi_l \\ P_u^{(n)} &\mapsto \varphi_l(uu') \end{aligned}$$

is a well defined injective map. Firstly, we tackle the problem of well definedness. Let $u_1, u_2 \in \Sigma^{2^l-k}$ such that $P_{u_1}^{(n)} = P_{u_2}^{(n)}$. Then

$$f_l(v) \in \varphi_l(u_1u') \iff u'v \in P_{u_1}^{(n)} \iff u'v \in P_{u_2}^{(n)} \iff f_l(v) \in \varphi_l(u_2u')$$

and $\varphi_l(u_1u') = \varphi_l(u_2u')$. For injectivity take two words $u_1, u_2 \in \Sigma^{2^l-k}$ with $\varphi_l(u_1u') = \varphi_l(u_2u')$. Let $w \in P_{u_1}^{(n)}$. Then decompose w into $u''v'$ with $u'' \in \Sigma^k$ and $v' \in \Sigma^l$. Now, by definition, $v' \in \varphi_l(u_1u'')$. Furthermore, from (P2) we can deduce $\varphi_l(u_1u'') = \varphi_l(u_2u'')$, and thus $v' \in \varphi_l(u_2u'')$. Because of this $w \in P_{u_2}^{(n)}$ and $P_{u_1}^{(n)} \subseteq P_{u_2}^{(n)}$. By exchanging the roles of u_1 and u_2 equality of $P_{u_1}^{(n)}$ and $P_{u_2}^{(n)}$ follows. Therefore

$$\left|\{[u] \mid u \in \Sigma^{2^l-k}\}\right| = \left|\{P_u^{(n)} \mid u \in \Sigma^{2^l-k}\}\right| \leq |\operatorname{im} \varphi_l| = 2^{k_l} \leq 2^{k_n}.$$

Recall that $l + k \leq n$. Consequently,

$$B_{\mathrm{II}} \leq (n+1)^2 \cdot 2^{k_n}.$$

To bound B_{III} we use the same method we used for B_{II}. Fix a $k < n$ and consider all classes generated by words in Σ^k. Firstly, note that for an $u \in \Sigma^k$ there can be words of different lengths in $P_u^{(n)}$, but we still know that $P_u^{(n)} \subseteq \bigcup_{l \leq m} \Sigma^{2^l+l-k}$ where $m = \max\{l \in \mathbb{N} \mid 2^l + l - k \leq n\}$ and $\Sigma^{-i} = \emptyset$. Furthermore, $m \leq \lceil \log_2 n \rceil$, because

$$2^{\lceil \log_2 n \rceil + 1} + \lceil \log_2 n \rceil + 1 - k \geq 2 \cdot n + 1 - (n-1) > n.$$

But $P_u^{2^l+l-k}$ contains by definition only witnesses of the same length, hence we can apply the same argument as before to obtain $|\{P_u^{2^l+l-k} \mid u \in \Sigma^k\}| \leq |\operatorname{im} \varphi_l| \leq 2^{k_l}$. Since $k_0 = 1$ we deduce

$$\left|\{P_u^{(n)} \mid u \in \Sigma^k\}\right| \leq \prod_{l \leq m} \left|\{P_u^{2^l+l-k} \mid u \in \Sigma^k\}\right| \leq \prod_{l \leq m} 2^{k_l} \leq 2 \cdot \prod_{l=1}^{m} 2^{k_l} \leq 2 \cdot 2^{m \cdot k_m}.$$

We know that $m \leq \log_2 n$, also $n \geq 2^N$ implies $\log_2 n \geq N$. Furthermore, if we assume for the sake of readability that $\log_2 n$ is a natural number, then by assumtion that k_n grows sufficiently fast we have

$$m \cdot k_m \leq \log_2 n \cdot k_{\log_2 n} \leq k_n.$$

As there are n possible values for k we obtain $B_{\mathrm{III}} \leq 2 \cdot n \cdot 2^{k_n}$.

Now we can finally give an upper bound for $\operatorname{ind} \Theta_n(L)$:

$$\begin{aligned} \operatorname{ind} \Theta_n(L) &\leq B_{\mathrm{I}} + B_{\mathrm{II}} + B_{\mathrm{III}} + B_{\mathrm{IV}} \\ &\leq (n+1) \cdot 2^{k_n} + (n+1)^2 \cdot 2^{k_n} + 2 \cdot n \cdot 2^{k_n} + 1 \\ &\leq 3 \cdot (n+1)^2 \cdot 2^{k_n}. \end{aligned}$$

This finishes the proof. $\square$

Note that this lemma can be applied to any surjective functions φ_n and f_n fulfilling the properties (P1) and (P2), not just the φ_n and f_n we defined. Now we can, using this lemma, show the main result from this section.

Theorem 3.8. *For any alphabet Σ with at least two letters the entropy function*

$$\begin{aligned} \eta\colon \mathcal{P}(\Sigma^*) &\to [0,\infty] \\ L &\mapsto \eta(L) \end{aligned}$$

is surjective.

Proof. For 0 and ∞ we have already seen that there are languages with that entropy. Thus let x be a positive real number. We would like to use the sequence $k_n = \lceil n \cdot x \rceil$, but it is not suitable, because $k_0 = 0$. Furthermore, k_1 could be larger than 2. Because of this we define $k_n = \max\{\min\{\lceil n \cdot x \rceil, 2^n\}, 1\}$. Now the sequence $(k_n)_{n\in\mathbb{N}}$ is suitable. Since Σ has at least two letters we can define f_n, φ_n, and L as above. Clearly, there is an $N_1 \in \mathbb{N}$ such that $k_n = \lceil n \cdot x \rceil$ for all $n \geq N_1$. Furthermore, there is an $N_2 \in \mathbb{N}$ such that for all $n \geq \max\{N_1, N_2\}$

$$n \cdot k_n = n \cdot \lceil n \cdot x \rceil \leq n \cdot (n \cdot x + 1) = n \cdot (n + \frac{1}{x}) \cdot x \leq 2^n \cdot x \leq k_{2^n}.$$

Hence we apply Lemma 3.7 with $N = \max\{N_1, N_2\}$ to obtain

$$2^{k_n} \leq \operatorname{ind} \Theta_n(L) \leq 3 \cdot (n+1)^2 \cdot 2^{k_n}$$

for all $n \geq 2^N$. Now we easily compute

$$\eta(L) \geq \limsup_{n\to\infty} \frac{\log_2(2^{k_n})}{n} = \limsup_{n\to\infty} \frac{\lceil n \cdot x \rceil}{n} = x$$

and

$$\eta(L) \leq \limsup_{n\to\infty} \frac{\log_2(3 \cdot (n+1)^2 \cdot 2^{k_n})}{n} = x.$$

Therefore, $\eta(L) = x$, and we conclude that η is surjective. □

Naturally the question arises whether same holds for unary alphabets. In the next subsection we will address this question.

3.3. **Unary Languages.** For the remainder of this section let $\Sigma = \{a\}$ be a unary alphabet. We shall write n instead of a^n. Then we can view Σ^* as $\mathbb{N}$ and a unary language L is just a subset of $\mathbb{N}$. We will show that the entropy of a language over a unary alphabet can be bounded by one and we will show that this upper bound is tight. Note that there are undeciable unary languages, for example we can encoded the halting problem using an enumeration $M_1, M_2, \ldots$ of all Turing machines and define $L = \{i \in \mathbb{N} \mid M_i \text{ halts on input } \varepsilon\}$. This makes the following result a bit surprising.

Theorem 3.9. *Let L be a unary language. Then $\eta(L) \leq 1$.*

Proof. Let $L \subseteq \mathbb{N}$ be a unary language. Clearly, $\operatorname{ind} \Theta_n(L) \leq 2^{n+1}$ and therefore

$$\eta(L) = \limsup_{n\to\infty} \frac{\log_2(\operatorname{ind} \Theta_n(L))}{n} \leq \limsup_{n\to\infty} \frac{\log_2(2^{n+1})}{n} = 1.$$

This finishes the proof. □

Using a construction similar to the one in the previous subsection we can show that this bound is tight.

Example 3.10. For any $n \in \mathbb{N}$ let $\varphi_n \colon \{0, \dots, 2^{n+1} - 1\} \to \mathcal{P}(\{0, \dots, n\})$ be a bijection. Note that if we know that a number is of the form $2^n + k$ and $k < 2^n$, then we can uniquely determine k and n from that number. With this in mind we define:

$$\mathbf{UInf} = \{2^{n+2^{n+m}} + k \mid n \in \mathbb{N}, m \leq 2^{n+1} - 1, k \in \varphi_n(m)\}.$$

Observe that from any number of the form $2^{n+2^{n+m}} + k$ uniquely determines the numbers n, m, and k. To compute the entropy of **UInf** let $n \in \mathbb{N}$. Firstly, we denote $2^{n+2^{n+m}}$ by $w_{n,m}$. Now let us consider the set $\{w_{n,m} \mid m \leq 2^{n+1} - 1\}$. If $w_{n,m} \neq w_{n,m'}$, then $m \neq m'$ and since φ_n is injective we have that there is a $k \in \varphi_n(m) \triangle \varphi_n(m')$. This k witnesses that $(w_{n,m}, w_{n,m'}) \notin \Theta_n(\mathbf{UInf})$. Hence

$$\operatorname{ind} \Theta_n(\mathbf{UInf}) \geq |\{w_{n,m} \mid m \leq 2^{n+1} - 1\}| = 2^{n+1}.$$

Together with Theorem 3.9 we can conclude that $\eta(\mathbf{UInf}) = 1$.

We have just seen that the entropy function is surjective over every alphabet with at least two elements. Hence we conjecture that the following holds.

Conjecture 3.11. *For any unary alphabet Σ we have that the entropy function*

$$\begin{aligned} \eta \colon \mathcal{P}(\Sigma^*) &\to [0,1] \\ L &\mapsto \eta(L) \end{aligned}$$

is surjective.

Unfortunately, we were not able to prove this conjecture. In the next section we will look at the entropy of decision problems and use an observation there to strengthen the surjectivity result for at least two element alphabets.

4. Entropy of Decision Problems

In this chapter we will connect topological entropy with decision problems. First we will compute the entropy of the **NP**-complete problem **SAT**. Then we will use padding to show that the entropy of any language can be reduced to zero. In particular this shows that there are undecidable languages with zero entropy.

To be able to compute the entropy of **SAT**, we need to define a suitable encoding. We use the alphabet $\{(,), \wedge, \vee, \neg, 0, 1\}$. To encode a formula φ we replace every variable x_n by $\operatorname{bin}(n)$, the binary representation of n. We denote the encoded formula by $\langle \varphi \rangle$. For example $\langle x_1 \wedge x_2 \rangle = 1 \wedge 10$. Now we can define

$$\mathbf{SAT} = \{\langle \varphi \rangle \mid \varphi \text{ is satisfiable}\}.$$

Lemma 4.1. *The language **SAT** has infinite entropy.*

Proof. Consider the set

$$\{\langle L_1 \wedge \cdots \wedge L_{2^n} \rangle \mid L_1 \in \{x_1, \neg x_1\}, \dots, L_{2^n} \in \{x_{2^n}, \neg x_{2^n}\}\}.$$

Take two words $w_1 = \langle \varphi_1 \rangle$ and $w_2 = \langle \varphi_2 \rangle$ from this set. If $w_1 \neq w_2$, then there is some $k \in \{1, \dots, 2^n\}$ such that the k^{th} literal of φ_1 and φ_2 differ. Without loss of generality assume that the k^{th} literal of φ_1 is x_k and the k^{th} literal of φ_2 is $\neg x_k$. Then $\varphi_1 \wedge x_k$ is satisfiable and $\varphi_2 \wedge x_k$ is not. Note that $|\wedge \operatorname{bin}(k)| \leq 1 + \log_2 2^n =$

$1+n$. Therefore $\wedge\,\mathrm{bin}(k)$ witnesses $(w_1, w_2) \notin \Theta_{n+1}(\mathbf{SAT})$. Since the set contains 2^{2^n} words we can now show that infinity is a lower bound for the entropy of **SAT**

$$\eta(\mathbf{SAT}) \geq \limsup_{n\to\infty} \frac{\log_2(2^{2^n})}{n+1} = \infty.$$

This concludes the proof. □

Next we will discuss the effect padding has on the complexity of a language. In complexity theory padding can be used to decrease the complexity of a language. What happens for topological entropy? For a language L over Σ define

$$\mathrm{PAD}(L) = \{uv \mid u \in L, v \in \Sigma^{2^{|u|}}\}.$$

Note that if L is in **EXPTIME**, then $\mathrm{PAD}(L)$ is in **P**. So $\mathrm{PAD}(L)$ is much easier than L, and this decrease in complexity is also reflected in the topological entropy of $\mathrm{PAD}(L)$.

Theorem 4.2. *Let L be a language over Σ with $|\Sigma| \geq 2$. Then*

$$\eta(\mathrm{PAD}(L)) = 0.$$

Proof. Note that every word in $\mathrm{PAD}(L)$ has a length of the form $2^k + k$. Consider the set $\{P_u^{(n)} \mid |u| \geq \log_2 n\}$. We will show that every $P_u^{(n)}$ is either the empty set or Σ^k for some $k \in \{0, 1 \ldots, n\}$. Let $u \in \Sigma^*$ with $|u| \geq \log_2 n$. For every $k \geq \log_2 n$ we have:

$$(2^{k+1} + (k+1)) - (2^k + k) = 2 \cdot 2^k - 2^k + 1 = 2^k + 1 \geq n + 1.$$

Hence the lengths of words in $\mathrm{PAD}(L)$ are so far apart that $P_u^{(n)} \subseteq \Sigma^k$ for some $k \in \{0, 1 \ldots, n\}$. If $P_u^{(n)}$ is not empty, then there is some $w \in P_u^{(n)}$ and $uw \in \mathrm{PAD}(L)$. We know that uw is of the form $u'v'$ for some $u' \in L$ and $v' \in \Sigma^*$ with $|v'| = 2^{|u'|}$. Since $2^{|u|} \geq n \geq |w|$ we have that w is a postfix of v'.

By definition $u'v'' \in \mathrm{PAD}(L)$ for all v'' with $|v''| = 2^{|u'|}$. As a consequence, $uv'' \in \mathrm{PAD}(L)$ for all $v'' \in \Sigma^k$ and therefore $P_u^{(n)} = \Sigma^k$. Hence we can bound the number of classes in $\Theta_n(\mathrm{PAD}(L))$ by

$$\begin{aligned} \operatorname{ind} \Theta_n(\mathrm{PAD}(L)) &\leq \left|\{P_u^{(n)} \mid |u| < \log_2 n\}\right| + \left|\{P_u^{(n)} \mid |u| \geq \log_2 n\}\right| \\ &\leq \left|\Sigma^{(\log_2 n)}\right| + n + 2. \end{aligned}$$

Now we can determine the entropy

$$\eta(\mathrm{PAD}(L)) \leq \limsup_{n\to\infty} \frac{\log_2(|\Sigma^{\log_2 n}| + n + 2)}{n} = 0.$$

This finishes the proof. □

Note that this works for any language L, even if it is undecidable, and since L can be reconstructed from $\mathrm{PAD}(L)$ we know that $\mathrm{PAD}(L)$ is also undecidable.

Corollary 4.3. *There are undecidable languages with zero entropy.*

This is a rather strange result, because the barrier of undecidability cannot be breached in classical complexity theory. Undecidable languages are always complicated.

We will use this result to show that not only is the entropy function surjective, there are even uncountably many languages for every entropy.

Corollary 4.4. *Over an at least two element alphabet Σ there are uncountably many languages with zero entropy.*

Proof. There are uncountably many languages over Σ and as mentioned before PAD is injective and every language in its image has zero entropy. □

The following lemma will help us to construct uncountably many languages for every entropy, not just for zero.

Lemma 4.5. *Let $L_1, L_2 \subseteq \Sigma^*$ be nonempty languages and $\#$ a new symbol not in Σ. Then*

$$\eta(L_1\#L_2) = \max\{\eta(L_1), \eta(L_2)\}.$$

Proof. Let $v' \in L_2$ with $|v'|$ minimal. We show that for $n \geq |v'|$

$$\operatorname{ind}\Theta_n(L_1\#L_2) = \operatorname{ind}\Theta_{n-|v'|-1}(L_1) + \operatorname{ind}\Theta_n(L_2) + k,$$

for k either 1 or 0. Consider the map

$$\begin{aligned}(\Sigma^*\#\Sigma^*)/\Theta_n(L_1\#L_2) &\to \Sigma^*/\Theta_{n-|v'|-1}(L_1) \sqcup \Sigma^*/\Theta_n(L_2)\\ [u]^n &\mapsto [u]_1^{n-|v'|-1}\\ [u\#v]^n &\mapsto [v]_2^n,\end{aligned}$$

where the sets in the image are potentially renamed to make the classes for L_1 and L_2 disjoint. It is left as an exercise to the reader to show that this map is well defined. It is clearly surjective and since v' is of minimal length and $[u\#v]^n = [u'\#v]^n$ also injective. Note that any word in $(\Sigma \cup \{\#\})^* \setminus (\Sigma^*\#\Sigma^*)$ has no positive witnesses and is in the same class as $\#\#$. This shows the above equality. Therefore

$$\begin{aligned}\eta(L_1\#L_2) &= \limsup_{n\to\infty} \frac{\operatorname{ind}\Theta_{n-|v'|-1}(L_1) + \operatorname{ind}\Theta_n(L_2)}{n}\\ &= \max\left(\limsup_{n\to\infty} \frac{\operatorname{ind}\Theta_n(L_1)}{n}, \limsup_{n\to\infty} \frac{\operatorname{ind}\Theta_n(L_2)}{n}\right) = \max\{\eta(L_1), \eta(L_2)\}\end{aligned}$$

as desired.

□

We can use this to strengthen Theorem 3.8.

Corollary 4.6. *For any alphabet Σ with at least two letters and any $x \in [0, \infty]$ the set*

$$\{L \subseteq \Sigma^* \mid \eta(L) = x\}$$

is uncountable.

Proof. Assume that Σ has just two elements. By Corollary 4.4 the claim holds for $x = 0$. Let $x \in (0, \infty]$. By Theorem 3.8 there exists a language L with entropy $2x$. Now, by Lemma 4.5, every language in the uncountable set $\{L_0\#L \mid \eta(L_0) = 0\}$ has entropy $2x$. Note that we have introduced a new symbol, hence we encode every language from the set over Σ and obtain by 3.5 an uncountable set of languages with entropy x.

□

5. Topological Entropy and Automata

Now we come to the second part of this article. Here we introduce k-stack pushdown automaton (k-stack PDA) and k-counter automata. We show that the entropy of a language that is accepted by a k-stack PDA can be bounded in terms of the number of stack symbols. As a corollary we obtain that all languages recognized by k-counter automata have zero entropy, proving an open conjecture from Schneider and Borchmann [3].

Before we start with the formal definition of k-stack PDAs, we fix some functions and conventions. We will write $\boldsymbol{v}$ for the tuple $(v_1, \dots, v_k)$ from the set $\Gamma_1^* \times \cdots \times \Gamma_k^*$ and $\boldsymbol{\varepsilon}$ for the tuple containing only ε in each entry. For a word $u = a_1 \dots a_n \in \Sigma^n$ and $I \subseteq \mathbb{N}$ let $\pi_I(u)$ be the *projection of u onto I*:

$$\pi_I(u) = a_{i_1} \dots a_{i_k} \text{ where } i_1 < \cdots < i_k \text{ and } \{i_1, \dots, i_k\} = I \cap \{1, \dots, n\}.$$

Let $\pi_n = \pi_{\{1,\dots,n\}}$, head $= \pi_{\{1\}}$, and tail $= \pi_{\{2,3,\dots\}}$. Dually we define the functions $\mathrm{bottom}(u) = \pi_{\{|u|\}}(u)$ and $\mathrm{front}(u) = \pi_{|u|-1}(u)$. For two words $u, v \in \Sigma^*$ where v is a postfix of u define $u - v$ as u_1, where $u = u_1 v$. For tuples $\boldsymbol{v}$ the functions π_I, head, tail, and $-$ are applied componentwise.

Definition 5.1. A k-stack PDA is a tuple $\mathcal{A} = (Q, \Sigma, \Gamma_1, \dots, \Gamma_k, \delta, q_0, F)$ where

- Q is a finite set (of *states*),
- Σ is an alphabet (the *input alphabet*),
- $\Gamma_1, \dots, \Gamma_k$ are alphabets (the *stack alphabets*),
- $\delta \subseteq \Gamma_1^{(1)} \times \cdots \times \Gamma_k^{(1)} \times Q \times (\Sigma \cup \{\varepsilon\}) \times Q \times \Gamma_1^* \times \cdots \times \Gamma_k^*$ (the *transition relation*),
- q_0 is from Q (the *initial state*), and
- F is a subset of Q (the *set of final states*).

A configuration of $\mathcal{A}$ is a tuple $(q, \boldsymbol{v}, w)$ with the current state q, the values stored in the stacks $\boldsymbol{v}$, and the remaining input w. For $a \in \Sigma^{(1)}$ we can make the transition

$$(q, \boldsymbol{v}, aw) \vdash (p, \boldsymbol{u}, w) \text{ if } (\mathrm{head}(\boldsymbol{v}), q, a, p, \boldsymbol{u} - \mathrm{tail}(\boldsymbol{v})) \in \delta.$$

Beware that a transition of the form $(\boldsymbol{\varepsilon}, p, a, q, \boldsymbol{u})$ can only be used if all stacks are empty. Also the symbol on top of a stack represented by v is the leftmost symbol of v. Hence the stacks grow to the left. The language accepted by $\mathcal{A}$ is defined as

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \exists p \in F.\ (q_0, \boldsymbol{\varepsilon}, w) \vdash^* (p, \boldsymbol{v}, \varepsilon)\}.$$

We call $\mathcal{A}$ *deterministic* if for every configuration K there is at most one configuration K' such that $K \vdash K'$, *total* if for every configuration K there is at least one configuration K' such that $K \vdash K'$, and *ε-free* if there is no transition of the form $(\boldsymbol{v}, q, \varepsilon, p, \boldsymbol{u})$ in δ.

In the following we will always assume that our automata are total. It is clear that a 0-stack PDA is just an ordinary finite automaton. We define a *k-counter automaton* to be a k-stack PDA where all stack alphabets are unary. The height of a stack can be seen as the value of a counter. Note that a counter automaton can only test whether a counter is zero or not. While counter automata do not seem to be that powerful, we have the following surprising result.

Theorem 5.2 (Theorem 7.9 from [1]). *A deterministic 2-counter automaton can simulate an arbitrary Turing machine.*

Because of this and since topological automata are inherently deterministic we will restrict ourselfs to deterministic ε-free k-stack PDAs for now.

Our next goal is to obtain an upper bound on the entropy of such an automaton. The idea is the following: First, translate the PDA $\mathcal{A}$ into a topological automaton $\mathcal{T}_{\mathcal{A}}$. Second, we observe that witnesses of length n can only see the leftmost n symbols on each stack. Third, we use this observation to find an upper bound for $\operatorname{ind}\Lambda_n(\mathcal{T}_{\mathcal{A}})$ and therefore also for $\eta(L(\mathcal{A}))$.

Fix a deterministic ε-free k-stack PDA $\mathcal{A} = (Q, \Sigma, \Gamma_1, \ldots, \Gamma_k, \delta, q_0, F)$. A naive choice for the states of $\mathcal{T}_{\mathcal{A}}$ would be $Q \times \Gamma_1^* \times \cdots \times \Gamma_k^*$. But recall that the states of a topological automaton are a compact Hausdorff space. Therefore, we adjust this idea a little bit and define for an alphabet Γ the set $\Gamma^\infty = \Gamma^* \cup \Gamma^{\mathbb{N}}$, where $\Gamma^{\mathbb{N}}$ denotes the set of all (right) infinite words over Γ. We equip Γ^∞ with the topology defined by the following basis of open sets

$$\{\{u\} \mid u \in \Gamma^*\} \cup \{u\Gamma^\infty \mid u \in \Gamma^*\},$$

i.e., the open sets are exactly the sets that can be expressed as unions of sets from the basis. The space Γ^∞ is compact.

Definition 5.3. The topological automaton $\mathcal{T}_{\mathcal{A}}$ is defined as

$$(Q \times \Gamma_1^\infty \times \cdots \times \Gamma_k^\infty, \Sigma, \alpha, (q_0, \boldsymbol{\varepsilon}), F \times \Gamma_1^\infty \times \cdots \times \Gamma_k^\infty),$$

where $\alpha((q, \boldsymbol{v}), w) = (p, \boldsymbol{u})$ if $(q, \boldsymbol{v}, w) \vdash^* (p, \boldsymbol{u}, \varepsilon)$. Here $\vdash^*$ is extended to infinite words in the obvious way. We equip $Q \times \Gamma_1^\infty \times \cdots \times \Gamma_k^\infty$ with the product topology, where the topology on Γ_i^∞ is the one defined above and the topology on Q is the discrete topology.

Clearly, we have that

$$L(\mathcal{T}_{\mathcal{A}}) = L(\mathcal{A}).$$

Now comes the second step of the proof.

Lemma 5.4. *For two states $(q, \boldsymbol{u})$ and $(p, \boldsymbol{v})$ of $\mathcal{T}_{\mathcal{A}}$. If $q = p$ and $\pi_n(\boldsymbol{u}) = \pi_n(\boldsymbol{v})$, then $((q, \boldsymbol{u}), (p, \boldsymbol{v})) \in \Lambda_n(\mathcal{T}_{\mathcal{A}})$.*

Proof. Let $F' = F \times \Gamma_1^\infty \times \cdots \times \Gamma_k^\infty$. Note that the lemma is equivalent to showing that for any $w \in \Sigma^{(n)}$ we have that $\alpha((q, \boldsymbol{u}), w) \in F'$ iff $\alpha((p, \boldsymbol{v}), w) \in F'$.

The proof is by induction on n.

If $n = 0$. Then $w = \varepsilon$ and the statement is trivially true.

If $n > 0$. Then either $w = \varepsilon$ and the statement is again trivial or $w = aw'$ for some $a \in \Sigma$ and $w' \in \Sigma^{(n-1)}$. Consider $(q', \boldsymbol{u}') = \alpha((q, \boldsymbol{u}), a)$ and $(p', \boldsymbol{v}') = \alpha((p, \boldsymbol{v}), a)$. Since $n > 0$ we have that $\operatorname{head}(\boldsymbol{u}) = \operatorname{head}(\boldsymbol{v})$. Furthermore, $p = q$ and therefore both states have to use the same transition. Hence $q' = p'$ and $\pi_{n-1}(\boldsymbol{u}') = \pi_{n-1}(\boldsymbol{v}')$. Applying the induction hypothesis yields

$$\alpha((q, \boldsymbol{u}), w) = \alpha((q', \boldsymbol{u}'), w') \in F' \text{ iff } \alpha((p, \boldsymbol{v}), w) = \alpha((p', \boldsymbol{v}'), w') \in F',$$

as desired. □

Theorem 5.5. *Let $\mathcal{A} = (Q, \Sigma, \Gamma_1, \ldots, \Gamma_k, \delta, q_0, F)$ be a deterministic ε-free k-stack PDA. Then*

$$\eta(L(\mathcal{A})) \leq \log_2 |\Gamma_1| + \cdots + \log_2 |\Gamma_k|.$$

Proof. From Lemma 5.4 we know that every equivalence class in $\Lambda_n(\mathcal{T}_\mathcal{A})$ can be represented by a state of the form $(q, \boldsymbol{u})$, where $q \in Q$ and $\boldsymbol{u} \in \Gamma_1^{(n)} \times \cdots \times \Gamma_k^{(n)}$. Therefore $\operatorname{ind} \Lambda_n(\mathcal{T}_\mathcal{A}) \leq |Q| \cdot |\Gamma_1^{(n)}| \cdot \ldots \cdot |\Gamma_k^{(n)}|$. Consequently,

$$\eta(L(\mathcal{A})) \leq \eta(\mathcal{T}_\mathcal{A}) = \limsup_{n\to\infty} \frac{\log_2(\operatorname{ind} \Lambda_n(\mathcal{T}_\mathcal{A}))}{n} \leq \log_2 |\Gamma_1| + \cdots + \log_2 |\Gamma_k|.$$

This concludes the proof. □

We will see in Example 6.6 that this upper bound can be reached using deterministic palindrome languages. From this theorem the following conjecture from Schneider and Borchmann follows.

Corollary 5.6. *Let L be a language accepted by a deterministic ε-free k-counter automaton. Then*

$$\eta(L) = 0.$$

Furthermore, we are now able to obtain a lower bound on the number of symbols in the stack alphabet.

Corollary 5.7. *Let $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ be a deterministic ε-free 1-stack PDA accepting the language L. Then $|\Gamma| \geq 2^{\eta(L)}$.*

Now we discuss what happens if we drop some of the restrictions of determinisity and ε-freeness. In the following example we see that in most cases the entropy can no longer be bounded.

Example 5.8. Define the language

$$\mathbf{T}_\infty = \{a^{n_1}\#\ldots\#a^{n_k}b^l a^m \mid m, k, l, n_1, \ldots, n_k \in \mathbb{N}, l \leq k, n_l = m\}.$$

To determine the entropy of $\mathbf{T}_\infty$ consider the set $\{a^{n_1}\#\ldots\#a^{n_k} \mid k, n_1, \ldots, n_k \leq n\}$. If two words u and u' from this set differ in the i^{th} block, i.e., $n_i \neq n'_i$. Then $b^i a^{n_i}$ is a word of length at most $2n$ that witnesses $(u, u') \notin \Theta_{2n}(\mathbf{T}_\infty)$. Therefore $\operatorname{ind} \Theta_{2n}(\mathbf{T}_\infty) \geq n^n$ and

$$\eta(\mathbf{T}_\infty) \geq \limsup_{n\to\infty} \frac{\log_2(n^n)}{2n} = \infty.$$

Clearly, there is a nondeterministic ε-free 1-stack PDA, a deterministic 1-stack PDA, and a nondeterministic ε-free 2-counter automaton accepting $\mathbf{T}_\infty$.

Corollary 5.9. *All deterministic ε-free context-free languages have finite entropy. But deterministic context-free languages can have infinity entropy.*

The only thing left to study are 1-counter automata. The language $\mathbf{T}_\infty$ does not seem to be recognized by a 1-counter automaton. But the following modified version can be recognized by a nondeterministic one.

$$\mathbf{T'}_k = \left\{ a_1^{n_1^1}\#\ldots\#a_1^{n_{l_1}^1} \ldots a_k^{n_1^k}\#\ldots\#a_k^{n_{l_k}^k} b a_j^m \;\middle|\; \exists l.\ n_l^j = m \right\}$$

Consider the set

$$\left\{ a_1^{n_1^1}\#\ldots\#a_1^{n_{l_1}^1} \ldots a_k^{n_1^k}\#\ldots\#a_k^{n_{l_k}^k} \;\middle|\; n_1^i < n_2^i < \cdots < n_{l_i}^i \leq n \text{ for all } 1 \leq i \leq k \right\}.$$

Any two words in this set can be separated by a witness of length at most n. Consequently, $\operatorname{ind}\Theta_n(\mathbf{T'}_k) \geq 2^{nk}$ and

$$\eta(\mathbf{T'}_k) \geq k.$$

Although the entropy of $\mathbf{T'}_k$ is unfortunately not infinite we have at least found a family of languages with unbounded entropy. But beware we have increased the alphabet to obtain a higher entropy. The alphabet of $\mathbf{T'}_k$ contains $k+2$ symbols. But if we encode $\mathbf{T'}_k$ over a two element alphabet we obtain, by Corollary 3.4,

$$\eta(\operatorname{enc}(\mathbf{T'}_k)) \geq \frac{k}{\lceil \log_2(k+2) \rceil}.$$

This gives us a family with unbounded entropy over a fixed alphabet.

Corollary 5.10. *Let Σ be an at least two element alphabet. The entropy of languages over Σ accepted by nondeterministic 1-counter automata can not be uniformly bounded.*

For deterministic counter automata that have at least two counters allowing ε-transitions makes the automata model Turing complete. We will show that if there is only one counter than ε-transitions might increase the computational expressibility but not the entropy.

For the computational expressibility consider the language

$$L = \{a^{n_1} i_1 a^{m_1} \# \dots \# a^{n_k} i_k a^{m_k} \mid k \geq 1, n_1, \dots, n_k, m_1, \dots, m_k \in \mathbb{N}, i_1, \dots, i_k \in \{0,1\},\\ i_l = 1 \text{ implies } n_l = m_l \text{ for all } l \leq k\}.$$

There is a deterministic 1-counter automaton accepting L but the ε-transitions seem to be absolutely necessary.

To determine the entropy let us fix a deterministic 1-counter automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. And define $\mathcal{T}_{\mathcal{A}}$ as $((Q \cup \{p_\infty\}) \times \mathbb{N}^\infty, \Sigma, \alpha, (q_0, 0), F \times \mathbb{N}^\infty)$, where $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$, p_∞ is a new state, and α is defined in the following way:

$$\alpha((q,c),w) = \begin{cases} (p,d) & \text{if } (q,c,w) \vdash^* (p,d,\varepsilon) \text{ and not } (p,d,\varepsilon) \vdash (p',d',\varepsilon) \\ (p_\infty, \infty) & \text{otherwise} \end{cases}$$

Note that if $\mathcal{A}$ has no ε-transitions then $\mathcal{T}_{\mathcal{A}}$ is the topological automation we constructed before. To compute the entropy of $\mathcal{T}_{\mathcal{A}}$ we first need to understand what transitions we added. Fix $q \in Q$ and $a \in \Sigma$. For every $c \in \mathbb{N}$ consider the unique maximal sequence

$$(q,c,a) \vdash (q_1, c_1, w_1) \vdash (q_2, c_2, w_2) \vdash \dots$$

and define the function $c \mapsto (p_c, m_c)$, where $\alpha((q,c),a) = (p_c, m_c)$.

Now there are two cases to consider. If there is a C such that for all i we have $c_i > 0$. Then for all $c \geq C$ we have $m := m_C - C = m_c - c$ and $p_c = p_C$ and α fulfils the following equation

$$\alpha((q,c),a) = \begin{cases} (p_c, m_c) & \text{if } c < C \\ (p_C, c+m) & \text{if } c \geq C \end{cases}$$

Note that m can be negative.

In the other case we have for all c that there is an i with $c_i = 0$. For larger c the sequences become arbitrarily long and since there are only finitely many states there must be $j > i$ such that $q_i = q_j$, $w_i = w_j$, and $c_j < c_i$. Define $k = c_i - c_j$. Observe that the value of $\alpha((q, c), a)$ depends only on $c \bmod k$ and therefore

$$\alpha((q, c), a) = (p_{c \bmod k}, m_{c \bmod k}).$$

Let $\boldsymbol{C}$ be the maximum of all C, $\boldsymbol{M} \geq 1$ minimal such that no m is smaller than $-\boldsymbol{M}$, and K the set containing all k's. Now we can deduce the analogue of Lemma 5.4.

Lemma 5.11. *If $c, c' > \boldsymbol{M} \cdot n \geq \boldsymbol{C}$, and $c \equiv c' \pmod{k}$ for all $k \in K$. Then $((q, c), (q, c')) \in \Lambda_n(\mathcal{T}_\mathcal{A})$ for all $q \in Q \cup \{p_\infty\}$.*

Proof. We will proof the lemma by induction on n.

If $n = 0$. Then the only witness to consider is ε and the claim is trivial.

If $n > 0$. Then consider a witness aw of length at most n. If (q, a) gives a transition as in the first case, then let C and m be the parameters as above. Since $c, c' \geq \boldsymbol{C} \geq C$ we have

$$\alpha((q, c), a) = (p_C, c + m) \qquad \text{and} \qquad \alpha((q, c'), a) = (p_C, c' + m).$$

By assumption $m \geq -M$, hence $c + m, c' + m \geq \boldsymbol{M} \cdot (n - 1)$. Also $c + m \equiv c' + m \pmod{k}$ for all $k \in K$. Therefore the claim holds by induction hypotheses.

If the transition is as in the second case with parameter k. Then, by assumption,

$$\alpha((q, c), a) = (p_{c \bmod k}, m_{c \bmod k}) = (p_{c' \bmod k}, m_{c' \bmod k}) = \alpha((q, c'), a)$$

and the claim holds trivially. □

Using this lemma we can deduce the desired theorem.

Theorem 5.12. *Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ be a deterministic 1-counter automaton with ε-transitions. Then*

$$\eta(L(\mathcal{A})) = 0.$$

Proof. Note that for every c there is a $c' \leq \prod_{k \in K} k$ such that $c \equiv c' \pmod{k}$ for all $k \in K$. Therefore Lemma 5.11 implies that every class in $\Lambda_n(\mathcal{T}_\mathcal{A})$ contains a state where the counter value is at most $\max\{\boldsymbol{M} \cdot n, \prod_{k \in K} k\}$. Hence for large n

$$\operatorname{ind} \Lambda_n(\mathcal{T}_\mathcal{A}) \leq (|Q| + 1) \cdot \boldsymbol{M} \cdot n$$

and $\eta(L(\mathcal{A})) = 0$. □

In Table 1 we summarize the results from this section.

	ε-transitions		ε-free	
	nondet.	deterministic	nondet.	deterministic
1-counter automaton	∞ ($\uparrow \mathbf{T'}_n$)	0	∞ ($\uparrow \mathbf{T'}_n$)	0
1-stack PDA	∞ ($\mathbf{T}_\infty$)	∞ ($\mathbf{T}_\infty$)	∞ ($\mathbf{T}_\infty$)	$\log_2 \lvert\Gamma\rvert$
k-counter automaton	$\underline{\infty}$	$\underline{\infty}$	∞ ($\mathbf{T}_\infty$)	0
k-stack PDA	$\underline{\infty}$	$\underline{\infty}$	∞ ($\mathbf{T}_\infty$)	$\sum_{i=1}^{k} \log_2 \lvert\Gamma_i\rvert$

TABLE 1. Upper bounds for the entropy of languages accepted by certain kinds of automata. Here k is at least 2. The $\underline{\infty}$ indicates that the model is Turing complete.

We have shown that most of the upper bounds presented in this table can be reached. But we suspect that this is not the case for all bounds.

Conjecture 5.13. *All languages accepted by 1-counter automata have finite entropy.*

This concludes our investigation of push-down automata.

6. Entropy of Example Languages

The purpose of this section is to give entropies of selected example languages. Some of these examples were already discussed in [3], the results from Examples 6.2, 6.4, and 6.7 are new and have to our knowledge not been discussed before.

Example 6.1. Let L be a regular language. Then $\Theta(L)$ is finite. Hence

$$\eta(L) = \limsup_{n\to\infty} \frac{\log_2 \operatorname{ind} \Theta_n(L)}{n} \leq \limsup_{n\to\infty} \frac{\log_2 \operatorname{ind} \Theta(L)}{n} = 0.$$

All regular languages have zero entropy, which goes with the intuition that regular languages are simple.

Example 6.2. An example considered in [3] is the *Dyck language with k sorts of parenthesis*, which consists of all balanced strings over $\{(_1,)_1, \ldots, (_k,)_k\}$. More generally, let Γ be an alphabet and $\overline{\Gamma} = \{\overline{a} \mid a \in \Gamma\}$. Then $\overline{}\colon \Gamma \to \overline{\Gamma}$ is a bijection. Now the *Dyck language over* Γ, denoted by $\mathbf{Dyck}_\Gamma$, is the set of all words u such that successively replacing $a\overline{a}$ in u by ε results in ε.

Lemma 6.3. *For all alphabets Γ we have $\eta(\boldsymbol{Dyck}_\Gamma) = \log_2 |\Gamma|$.*

Proof. Let $a \in \Gamma$. Observe that the set $\Gamma^{(n)} \cup \{\overline{a}\}$ contains exactly one representative of each class of $\Theta_n(\mathbf{Dyck}_\Gamma)$. Therefore $\operatorname{ind} \Theta_n(\mathbf{Dyck}_\Gamma) = |\Gamma^{(n)}| + 1$ and we can compute

$$\eta(\mathbf{Dyck}_\Gamma) = \limsup_{n\to\infty} \frac{\log_2(|\Gamma^{(n)}| + 1)}{n} = \log_2 |\Gamma|.$$

This finishes the proof. □

Another example discussed in [3] is the palindrome language. For an alphabet Σ define the *palindrome language over* Σ to be

$$\mathbf{Pali}_\Sigma = \{uu^R \mid u \in \Sigma^*\}.$$

Schneider and Borchmann showed that $\log_2 |\Sigma| \leq \eta(\mathbf{Pali}_\Sigma) \leq \log_2 |\Sigma| + 1$.

Example 6.4. We will consider here the *deterministic palindrome language.* Assume that # is not in Σ and define

$$\mathbf{DPali}_\Sigma = \{u\#u^R \mid u \in \Sigma^*\}.$$

The entropy can be computed similar to that of the Dyck languages. Consider a class in $\Theta_n(\mathbf{DPali}_\Sigma)$ with at least one positive witness. Let w be the positive witness of minimal length. Then either $w = \#w'$ or $w \in \Sigma^*$. In the first case the class is represented by w'^R and in the second case it is represented by $w^R\#$. Therefore

$$\eta(\mathbf{DPali}_\Sigma) = \limsup_{n\to\infty} \frac{\log_2 2 \cdot |\Sigma^{(n)}|}{n} = \log_2 |\Sigma|.$$

Both $\mathbf{Dyck}_\Gamma$ or $\mathbf{DPali}_\Gamma$ are typical examples for languages recognized by PDAs. We can apply Corollary 5.7 to obtain the following result.

Corollary 6.5. *Every deterministic ε-free 1-stack PDA recognizing $\boldsymbol{Dyck}_\Gamma$ or $\boldsymbol{DPali}_\Gamma$ has at least $|\Gamma|$ many stack symbols.*

Example 6.6. We construct a product palindrome language. Let $\Gamma_1, \ldots, \Gamma_k$ be alphabets and consider the language $\mathbf{DPali}_{\Gamma_1 \times \cdots \times \Gamma_k}$. As we have just seen the entropy of this language is

$$\eta(\mathbf{DPali}_{\Gamma_1 \times \cdots \times \Gamma_k}) = \log_2 |\Gamma_1| + \cdots + \log_2 |\Gamma_k|.$$

Obviously, there is a deterministic ε-free k-stack PDA with stack alphabets $\Gamma_1, \ldots, \Gamma_k$ accepting $\mathbf{DPali}_{\Gamma_1 \times \cdots \times \Gamma_k}$. Therefore the upper bound given in Theorem 5.5 can be reached.

Finally, we will discuss the mathematically very interesting language of all prime numbers.

Example 6.7. Let $\mathbf{Prime} = \{p \mid p \text{ is prime}\}$ be the unary encoding of all prime numbers. For $m > 2$ there cannot be two consecutive numbers in P_m since the only k for which k and $k+1$ is prime is 2. Because of this, P_m contains only even numbers or only odd numbers if m is odd or even, respectively. Consequently, $\operatorname{ind}\Theta_n(\mathbf{Prime}) \leq 2 \cdot 2^{\lceil n/2 \rceil} + 3$ and $\eta(\mathbf{Prime}) \leq \frac{1}{2}$. We can make a similar argument for the prime 3. For $m > 3$ there cannot be a k such that $k, k+2, k+4 \in P_m$ because one of these numbers is divisible by 3. Thus $\operatorname{ind}\Theta_n(\mathbf{Prime}) \leq 2 \cdot 3 \cdot 2^{\lceil \frac{n}{2} \cdot \frac{2}{3} \rceil} + 4$ and $\eta(\mathbf{Prime}) \leq \frac{1}{3}$. These observations lead to the following definition:

Definition 6.8. A set $A \subseteq \{0, \ldots, n-1\}$ *represents a plausible sequence of primes of length* n if for all primes $p \leq n$

$$A \bmod p \neq \{0, \ldots, p-1\}.$$

The set A *represents an occurring sequence of primes of length* n if there is a $k \in \mathbb{N}$ such that for all $l \in \{0, \ldots, n-1\}$

$$k + l \text{ is prime} \iff l \in A.$$

We denote the number of plausible sequences of length n by s_n.

Note that for example the set $\{0, 1\}$ occurs for $k = 2$, but is not plausible since $\{0, 1\} \bmod 2 = \{0, 1\}$. This can happen because the idea behind the plausible sequences is that every number in the sequence dividable by p is not a prime, which holds for all numbers except for p itself. But if we say that the starting point k of the sequence is greater than the length n of the sequence, than p cannot occur in $k + \{0, \ldots, n-1\}$. Hence if A represents an occurring sequence of length n, which occurs for some $k > n$, then A also represents a plausible sequence.

With this observation we can bound $\operatorname{ind}\Theta_n(\mathbf{Prime})$ by

$$|\{[k] \mid k \leq n+1\}| + |\{[k] \mid k > n+1\}| \leq n + 2 + s_{n+1}.$$

Lemma 6.9. *Let $p_1, \ldots, p_k$ be the first k primes and $n = n' \cdot \prod_{i \leq k} p_i$. We have that*

$$s_n \leq \left(\prod_{i \leq k} p_i \right) \cdot 2^{n \cdot \prod_{i \leq k} \frac{p_i - 1}{p_i}}.$$

Proof. For every $i \leq k$ fix an $\ell_i \in \{0, \ldots, p_i - 1\}$. Now we bound the number of plausible sequences A of length n with $\ell_i \notin (A \bmod p_i)$ for all i. Clearly,

$$A \subseteq \bigcap_{i \leq k} \left(\{0, \ldots, n-1\} \setminus \{m \cdot p_i + \ell_i \mid m < \frac{n}{p_i}\} \right).$$

Furthermore,

$$\left| \{0, \ldots, n-1\} \setminus \{m \cdot p_i + \ell_i \mid m < \frac{n}{p_i}\} \right| = n - \frac{n}{p_i}$$

and

$$\begin{aligned} \left| \bigcap_{i \leq k} \left(\{0, \ldots, n-1\} \setminus \{m \cdot p_i + \ell_i \mid m < \frac{n}{p_i}\} \right) \right| &= \sum_{m, j_1, \ldots, j_m \leq k} (-1)^m \frac{n}{p_{j_1} \cdot \ldots \cdot p_{j_m}} \\ &= n \cdot \prod_{i \leq k} \frac{p_i - 1}{p_i}. \end{aligned}$$

Together with the fact that there are $\prod_{i \leq k} p_i$ possible choices for $\ell_1, \ldots, \ell_k$ this proves the claim. $\square$

Using this lemma we can compute the entropy.

Theorem 6.10. *The entropy $\eta($**Prime**$)$ is zero.*

Proof. From the previous observation and Lemma 6.9 we can deduce that

$$\eta(\mathbf{Prime}) \leq \prod_{i \leq k} \frac{p_i - 1}{p_i}$$

for all k. An old result from Mertens [2] states

$$\prod_{i \leq k} \frac{p_i - 1}{p_i} \in \mathcal{O}\left(\frac{1}{\log_2(k)} \right). \text{ Therefore } \lim_{k \to \infty} \prod_{i \leq k} \frac{p_i - 1}{p_i} = 0$$

and we conclude that $\eta(\mathbf{Prime}) = 0$. $\square$

Even though not relevant for the entropy of **Prime**, the following observations are simply too beautiful not to be mentioned. We conjecture the following.

Conjecture 6.11. *Every plausible sequence of primes occurs at least once.*

We have computationally verified this conjecture for sequences of length up to 29. What makes this conjecture interesting is the following lemma.

Lemma 6.12. *The following statements are equivalent:*

(1) Every plausible sequence of primes occurs at least once.
(2) Every plausible sequence of primes occurs infinitely often.

Proof. Assume the sequence represented by $A \subseteq \{0, \ldots, n\}$ occurs only k times. Then let $N \in \mathbb{N}$ such that there are more than k plausible sequences represented by $A_1, \ldots, A_l$ of length N with the initial sequence as prefix, i.e., $A_i \cap \{0, \ldots, n\} = A$. Hence at least one of these A_i does not represent an occurring sequence, a contradiction. $\square$

Note that the twin prime conjecture can be formulated as: the sequence $\{0, 2\}$ occurs infinitely often. As a consequence, Conjecture 6.11 is a generalization of the twin prime conjecture and the Green-Tao Theorem.

7. Conclusion

In this article we introduced the notion of a topological automaton from Steinberg [4]. We defined the topological entropy of a topological automata and of a formal language.

We further investigated the notion of topological entropy of formal languages and its suitability as a measure of the complexity of formal languages. We were able to calculate the entropy of the Dyck languages, an previously open problem, and provided many other new examples.

We modified an example from [3] to show that the entropy function is surjective for every Σ with $|\Sigma| \geq 2$. Whether this is also the case for unary alphabets remains an open problem (Conjecture 3.11).

Our second main result concerns a conjecture from Schneider and Borchmann [3]. They suspected that all languages accepted by a one-way finite automaton equipped with a fixed number of counters and an acceptance condition that does only require to check local conditions have zero entropy. We showed that this conjecture holds if we assume the automaton to be deterministic and ε-free and we were even able to generalize this result to deterministic ε-free push-down automata. We showed that the entropy of a language accepted by such an automaton is bounded in terms of the sizes of the stack alphabets of the automaton. This result proves that all deterministic ε-free context-free languages have finite entropy. An open problem from this section is whether all languages accepted by nondeterministic 1-counter automata with ε-transitions have finite entropy (Conjecture 5.13).

On the other hand, we also saw that the definition of entropy is not very robust, since we can use padding to decrease the entropy of any language to zero. Consequently, there are also undecidable languages with zero entropy. It is also counterintuitive that the entropy of a language is not the same as the entropy of the reversed language. Hence we suggest to define something like the entropy of the *core* of a language with the following properties:

- the entropy of the core of a language is at least as large as the entropy of the language,
- padding a language does not influence the entropy of the core of this language,
- reversing a language does not influence the entropy of the core of the language,
- the entropy of the core of the languages we used to show surjectivity should be infinite, and
- encoding the language should not change the entropy of the core of the language.

We propose to define the entropy of the core of a language L in the following way:

$$\eta_{\mathrm{core}}(L) = \sup\{\eta(L') \mid L' \in \mathrm{core}(L)\},$$

where $\mathrm{core}(L)$ should contain at least L, $L^R = \{w^R \mid w \in L\}$, and every L' such that there is an encoding enc with $\mathrm{enc}(L') = L$. But a suitable definition of $\mathrm{core}(L)$ remains to be found.

References

[1] John E. Hopcroft, Jeffrey D. Ullman. "Introduction to Automata Theory, Languages, and Computation". Addison-Wesley Publishing Company, Inc, 1979, pp. 171–172.

[2] Franz Mertens. "Ein Beitrag zur analytischen Zahlentheorie". In: *Journal für die reine und angewandte Mathematik*, 1874, pp. 46-62.

[3] Friedrich M. Schneider, Daniel Borchmann. "Topological Entropy of Formal Languages". In: *Semigroup Forum* (2017), pp. 556–581.

[4] Benjamin Steinberg. "Topological Dynamics and Reconstruction of Languages". In: *CoRP* abs/1306.1468 (2013).