A Structural Property of Regular Frequency Computations

Holger Austinat, Volker Diekert, Ulrich Hertrampf

Universität Stuttgart
Institut für Informatik
Breitwiesenstr. 20–22

Abstract. The notion of an (m, n)–computation was already introduced in 1960 by Rose and further investigated by Trakhtenbrot in 1963. It has been extended to finite automata by Kinber in 1976 and he has shown an analogue of Trakhtenbrot's result: The class of languages (m, n)-recognizable by deterministic finite automata is equal to the class of regular languages if and only if $2m > n$. Furthermore, for a unary alphabet, the class of (m, n)-recognizable languages coincides with the class of regular languages for all m and n.
In this paper, we will present the first structural property of (m, n)-recognizable languages which is valid for all $1 \leq m \leq n$ and for all alphabets. Kinber's result for unary alphabets becomes a corollary. This property is also used to show that certain non-unary languages are not (m, n)-regular and that the class of all (m, n)-recognizable languages is not closed under the reversal operation. However, this class forms a Boolean algebra.

1 Introduction

The notion of frequency computations was introduced in 1960 by Rose [12]: An (m, n)-computation of a function $f : \Sigma^* \to \mathbb{N}$ is given by a deterministic Turing machine M which on n pairwise different inputs produces n output values where at least m of the n values are in accordance with f. In such a case we say that the Turing machine computes f with frequency $\geq m/n$. It was asked whether f is recursive if m is close to n [11, p. 393]; and this question was answered positively by Trakhtenbrot. He showed that an (m, n)-computable function f is recursive if $2m > n$. Furthermore he showed that there are uncountably many functions being (m, n)-computable whenever $0 < 2m \leq n$. In particular, there are non-recursive functions of this type [13]. Later Degtev, Kummer and Stephan showed that for $2m \leq n$ and $2m' \leq n'$, the classes of (m, n)- and (m', n')-computable functions differ whenever $m \neq m'$ or $n \neq n'$ [5, 9]. The exact inclusions, however, are still unknown (except for some special cases).

A *time bounded* version of frequency computations has also been considered. For example, one may require that the Turing machine which performs the frequency computations works in polynomial time. In this case, the inclusion problem for frequency classes bears a one-to-one correspondence to so-called

(m, n)-*admissible* sets, which can be handled by finite combinatorics [3, Theorem 7.3]. Hinrichs and Wechsung showed that $(m+1, n+1)$P is a proper subset of (m, n)P whenever $m < 2^{n-m}$ [7], where (m, n)P denotes the class of all sets whose characteristic functions are (m, n)-computable in polynomial time.

The notion of frequency computations has also been extended to deterministic finite automata, which leads to *regular frequency computations* and to the notion of (m, n)-*recognizable language*. Formal languages are viewed as characteristic functions and then Trakhtenbrot's result for functions carries over to regular frequency computations: The class of (m, n)-recognizable languages equals the class of regular languages if and only if $2m > n$. If $2m \leq n$, then there are uncountably many subsets of Σ^* that are (m, n)–recognizable as soon as $|\Sigma| \geq 2$. However, when restricted to a one-letter alphabet, then all (m, n)-recognizable languages are regular for $1 \leq m \leq n$. These results about regular frequency computations are due to Kinber [8], see also [1].

In this paper, we will present structural properties of (m, n)–recognizable languages which hold for all alphabets and all $1 \leq m \leq n$. A priori, since there are uncountably many (m, n)–recognizable languages, it is not clear to show for any specific language that it is not $(1, n)$–recognizable for some n. Here we derive a structural property of (m, n)–recognizable languages which can be viewed as weak analogue of a pumping lemma. This property shows immediately that there is no m, n such that a language like $\{a^i b^i \mid i \geq 0\}$ becomes (m, n)-recognizable. More general, no inherently context-free language is (m, n)-recognizable. It also shows that the class of (m, n)-recognizable languages is not closed under the reversal operation which reads words from right to left. Another corollary is the previously known result that all (m, n)-recognizable languages over a unary alphabet are regular.

Finally we show that the union of the classes of (m, n)-recognizable languages forms a Boolean algebra.

2 Notions and Examples

By Σ we denote a finite alphabet and ε denotes the empty word. The characteristic function of a language $L \subseteq \Sigma^*$ is denoted by $\chi_L : \Sigma^* \to \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$ is the set of Boolean values; it is defined as $\chi_L(w) = 1$ if $w \in L$ and $\chi_L(w) = 0$ otherwise.

For a word the set of prefixes is denoted by $\mathrm{Pref}(w)$ and its set of suffixes is denoted by $\mathrm{Suff}(w)$. The notation is extended to languages $L \subseteq \Sigma^*$ by

$$\mathrm{Pref}(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^* : uv \in L\}.$$

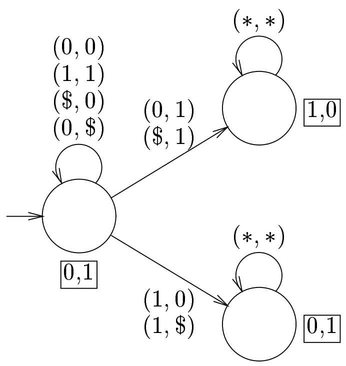
The notion of a deterministic finite automaton will be used in the following way where we allow n input tapes (read synchronously) and an output is specified by the type of a state. Formally, an n–DFA is a tuple $A = (Q, \Sigma, \$, \delta, q_0, \tau, n)$, where Q is a finite set of states with initial state q_0, the set Σ is a finite alphabet and \$ is a new symbol, $\$ \notin \Sigma$, the mapping $\delta : Q \times (\Sigma \cup \{\$\})^n \to Q$ is the transition function, the mapping $\tau : Q \to \mathbb{B}^n$ specifies the type of a state, and n is the number of components, $1 \leq n$.

We describe the behavior of such an automaton. For an input vector $u = (u_1, \ldots, u_n) \in (\Sigma^*)^n$, define $|u| = \max\{|u_i| \mid 1 \leq i \leq n\}$, and $q \cdot u = \delta(q, (u_1\$^{\ell_1}, \ldots, u_n\$^{\ell_n}))$, where $\delta : Q \times ((\Sigma \cup \{\$\})^n)^* \to Q$ is the natural extension of δ and $\ell_i = |u| - |u_i|$ for $1 \leq i \leq n$. The output of the automaton is then defined to be the type $\tau(q_0 \cdot u)$.

Let $1 \leq m \leq n$. A language $L \subseteq \Sigma^*$ is (m, n)-recognized by an n-DFA A if and only if for each n-tuple $u = (u_1, \ldots, u_n) \in (\Sigma^*)^n$ of pairwise distinct words, the n-tuples $\tau(q_0 \cdot u)$ and $(\chi_L(u_1), \ldots, \chi_L(u_n))$ coincide on at least m components. A language L is called *(m, n)-recognizable*, if there exists an n-DFA A that (m, n)-recognizes L. Note that the same n-DFA may accept many different languages, in general. The class of all (m, n)-recognizable languages is denoted by (m, n)REG. The union over all $1 \leq m \leq n$ of all classes (m, n)REG is denoted by FREQ-REG. Note that (m, n)REG and FREQ-REG are closed under complementation for trivial reasons (only the output types have to be inverted).

Remark 1. In Kinber's work [8] the input words are right-aligned, being padded with \$ symbols from the left. It turns out that the classes are incomparable, see Corollary 4 below. However, the following can be shown for every $1 \leq m \leq n$: A language L is (m, n)-recognizable in Kinber's model if and only if its reversal language $\tilde{L}$ is (m, n)-recognizable in our model. Thus, by symmetry there is essentially no difference between both models.

Example 1. Let $\Sigma = \{0, 1\}$, $x \in \mathbb{R}$ be an arbitrary real number from the half open interval $[0, 1)$, and $bin(x)$ be the infinite binary expansion of x (padded with zeros if necessary). Then the language $L_x := \{w \in \Sigma^* \mid 0.w < bin(x)\}$ is $(1, 2)$-recognizable. A 2-DFA recognizing L_x (simultaneously for all $x \in [0, 1)$) is shown below; the boxed pairs are the types (=outputs), and a $*$ stands for an arbitrary input. The example is from [1].



Example 2. Kinber [8] gave the following example: Let $\alpha \in \Sigma^\omega$ be an infinite word. Then $K_\alpha = \mathrm{Pref}(\alpha)$ is $(1, 2)$-recognizable.[1] Again there is one automaton which works for all α. We just have to test whether one input is the prefix of the other (in which case we output $(1, 0)$ for the shorter and longer input word, respectively). If the two inputs are incomparable, then we output $(0, 0)$. In the sequel, the languages K_α will be referred to as Kinber languages.

Note that there are uncountably many languages L_x and K_α; thus there are languages in $(1, 2)$REG that are not recursively enumerable. The following example shows that there are languages in $(1, 2)$REG which are context-sensitive but not context-free.

Example 3. Consider the Kinber language K defined by the infinite word

$$\alpha = 0101101110 \cdots 1^i 01^{i+1} 0 \cdots ,$$

i.e., $K = K_\alpha = \mathrm{Pref}(\alpha)$. Then K is in $(1, 2)$REG by Example 2, and it is context-sensitive but not context-free. In fact, the language K does not contain any infinite context-free subset, so it is a so-called inherently context-sensitive language. (Compare this with the statement in Corollary 3 below.)

Remark 2. Stephan and Lange have observed that $(1, 2)$REG contains non-regular context-free languages (personal communication). In fact, they have used the same Kinber language K as in Example 3, but they considered its complement $L = \{0, 1\}^* \setminus K$ instead. From our remark about complementation, we have $L \in (1, 2)$REG, too. The language L is context-free without being deterministic context-free. It is a language which is inherently ambiguous.

To see that L is context-free, we represent it as the union of five languages: $L_1 = \{1v \mid v \in \{0, 1\}^*\}$, $L_2 = \{011v \mid v \in \{0, 1\}^*\}$, $L_3 = \{u00v \mid u, v \in \{0, 1\}^*\}$, $L_4 = \{u01^i01^j0v \mid u, v \in \{0, 1\}^* \wedge j \neq i + 1\}$, $L_5 = \{u01^i01^j \mid u \in \{0, 1\}^* \wedge j > i + 1\}$. It should be clear that $L = L_1 \cup \ldots \cup L_5$, and that in all five cases, corresponding context-free grammars can easily be constructed.

3 A Structural Property of (m, n)REG–Classes

For a language $L \subseteq \Sigma^*$ and a number $r \in \mathbb{N}$ we define

$$L_r = \{xyz \in L \mid xy^*z \subseteq L,\ 1 \leq |y|, \text{and } |yz| \leq r\}.$$

The language L_r is a union of infinite regular subsets of L having a particularly simple structure: the pumping part is not longer than r characters and is near the end of the string.

Note that if we have $xy^*z \subseteq L$ such that $1 \leq |y|$ and $|yz| \leq r$, then we have in fact $xy^+z \subseteq L_s$ for all $r \leq s$. In particular, if L contains some infinite regular subset R, then L_r contains an infinite regular subset of R for almost all r. Now we are ready to state our main theorem.

[1] Kinber actually has to consider the reversal of K_α, since in his model the input words are right-aligned; cf. our remark above.

Theorem 1. *Let $L \in (1,n)\mathrm{REG}$. Then there exists an $r \in \mathbb{N}$ such that the prefix closed language $\mathrm{Pref}(L \setminus L_r)$ contains no infinite regular subset.*

Before giving the proof for Theorem 1, we will first state and prove some corollaries.

Corollary 1 ([8]). *Let $|\Sigma| = 1$, $L \subseteq \Sigma^*$, and $L \in (1,n)\mathrm{REG}$. Then L is regular.*

Proof. Let $\Sigma = \{a\}$ be the unary alphabet. From Theorem 1 we know that for some $r \in \mathbb{N}$ the set $\mathrm{Pref}(L \setminus L_r)$ does not contain any infinite regular subset, but, since $|\Sigma| = 1$ and $\mathrm{Pref}(L \setminus L_r)$ is prefix closed, this means that $\mathrm{Pref}(L \setminus L_r)$ is finite. Thus, $L \setminus L_r$ is finite as well. The set L_r is regular over unary alphabets for all r since it can be written as a finite union of sets of the form $a^t(a^s)^*$ for suitable t and s. Therefore, $L = (L \setminus L_r) \cup L_r$ is regular. □

Corollary 2. *Let $L \in (1,n)\mathrm{REG}$. If $\mathrm{Pref}(L)$ contains an infinite regular subset, then L contains an infinite regular subset, too.*

Proof. Assume by contradiction that $\mathrm{Pref}(L)$ contains an infinite regular subset, but L does not. From the latter it follows that $L_r = \emptyset$ for all $r \geq 0$, hence $\mathrm{Pref}(L) = \mathrm{Pref}(L \setminus L_r)$. From Theorem 1 we know that $\mathrm{Pref}(L)$ does not contain any infinite regular subset—a contradiction to our assumption about $\mathrm{Pref}(L)$. □

Using this corollary, we can easily prove that there is no n such that the language $L = \{a^i b^i \mid i \geq 0\}$ is $(1,n)$-recognizable. This follows from the fact that $a^* \subseteq \mathrm{Pref}(L)$, but there is no infinite regular subset of L. In fact we can generalize this example. A language $L \subseteq \Sigma^*$ is called *inherently context-free*, if L is infinite and context-free, but it does not contain any infinite regular subset.

Corollary 3. *Let $L \subseteq \Sigma^*$ be an inherently context-free language. Then we have $L \notin$ FREQ-REG.*

Proof. The Pumping Lemma for context-free languages shows that $\mathrm{Pref}(L)$ contains an infinite regular subset (just take the prefixes ending after the first pumping position). Corollary 2 then tells us that L itself must contain an infinite regular subset, contradicting our assumption that L is inherently context-free. □

By Remark 2 the class FREQ-REG contains context-free languages. But it is open whether there are deterministic (or unambiguous) context-free languages which are $(1,n)$-recognizable without being regular.

Corollary 2 has another interesting consequence. It shows that the class FREQ-REG is not closed under reversal. (The reversal operation reads words from right to left, see e.g. [4]) The non-closure property holds in a strong form:

Corollary 4. *There are uncountably many $(1,2)$–recognizable languages K such that the reversal language $\tilde{K}$ is not in FREQ-REG.*

Proof. We shall exhibit only one specific example of a $(1, 2)$–recognizable languages K such that $\tilde{K}$ is not in FREQ-REG. It is left to the reader to see that the example can be modified in order to produce uncountably many of them (over the alphabet $\{0, 1\}$).

Consider again the infinite word

$$\alpha = 0101101110 \cdots 1^i 01^{i+1} 0 \cdots$$

and let $K = \mathrm{Pref}(\alpha)$. By Example 2 we know that K is $(1, 2)$–recognizable. An easy inspection shows that K has no infinite regular subsets. Hence the reversal language $\tilde{K}$ has no infinite regular subsets as well. On the other hand, 1^* is an infinite regular subset of the prefix closure $\mathrm{Pref}(\tilde{K})$. Corollary 2 tells us that $\tilde{K}$ is not in FREQ-REG. $\square$

Corollary 2 fails to show that a language like $L = \{ww \mid w \in \Sigma^*\}$ is not $(1, n)$-recognizable, since L contains infinite regular subsets. However, we can use the following:

Corollary 5. *Let $L \in (1, n)\mathrm{REG}$. Then there exists an $r \geq 1$ such that for all $u, v, w \in \Sigma^*$ the following implication holds:*

$$|uv^*w \cap L| = \infty \quad \Longrightarrow \quad |uv^*w \cap L_r| = \infty.$$

Proof. If this corollary did not hold, then for every $r \geq 1$ there were words $u, v, w \in \Sigma^*$ with $|v| > 0$ and infinitely many i such that $uv^iw \in L \setminus L_r$. But this implies $uv^* \subseteq \mathrm{Pref}(L \setminus L_r)$, which means that $\mathrm{Pref}(L \setminus L_r)$ contains an infinite regular subset for all $r \geq 1$, contradicting Theorem 1. $\square$

We can now show that for all n the language $L = \{ww \mid w \in \{a, b\}^*\}$ is not in $(1, n)\mathrm{REG}$. Consider the following set $(a^rb)^*$ whose intersection with L is infinite, because $(a^rba^rb)^* \subseteq L$. But $(a^rb)^* \cap L_r$ is empty, therefore is is certainly not infinite.

Proof (of Theorem 1). Let $L \in (1, n)\mathrm{REG}$, and let n be minimal, i.e., for all $n' < n$ we have $L \notin (1, n')\mathrm{REG}$. Let A be an automaton $(1, n)$–recognizing L and let Q be its set of states. We show the assertion of the theorem with $r = |Q|$.

First, we consider the case $n = 1$. Then L is regular, and from the Pumping Lemma we know that every word $w \in L$ with $|w| \geq r$ has a decomposition $w = xyz$ with $1 \leq |y|$ and $1 \leq |yz| \leq r$ such that $xy^*z \subseteq L$. Therefore the set $L \setminus L_r$ is finite, and so is $\mathrm{Pref}(L \setminus L_r)$.

Hence, from now on we let $n > 1$. If $\mathrm{Pref}(L \setminus L_r)$ does not contain any infinite regular subset, we are done. Hence we assume the contrary, $\mathrm{Pref}(L \setminus L_r)$ does contain an infinite regular subset. Then $\mathrm{Pref}(L \setminus L_r)$ contains a regular subset of the form uv^* with $|v| \geq 1$. By replacing v by a large enough power like $v^{r!}$, we may assume that for all $q \in Q$ the following holds:

$$q \cdot (\varepsilon, \ldots, \varepsilon, v) = q \cdot (\varepsilon, \ldots, \varepsilon, v^2). \tag{1}$$

Let $S = \mathrm{Suff}(uv)$ be the finite set of suffixes of uv. Using this set, we will construct an $(n-1)$–DFA $A' = (Q', \Sigma, \$, \delta', q'_0, \tau', n-1)$. The new automaton will be a simulation of the old one in the following sense: On an input of $n-1$ words $(u_1, \ldots, u_{n-1})$, the automaton A' is led to a state that resembles the state A is led into on input $(u_1, \ldots, u_{n-1}, uv^\omega)$ (where the n-th input is cut to the length of the longest u_i).

- The set of states is $Q' := \{\langle q, s\rangle \mid q \in Q, s \in S\}$.
- The initial state is $q'_0 := \langle q_0, uv\rangle$.
- For the transition function, let $c_1, \ldots, c_{n-1} \in (\Sigma \cup \{\$\})$, and define

$$\delta'(\langle q, s\rangle, (c_1, \ldots, c_{n-1})) := \langle \delta(q, (c_1, \ldots, c_{n-1}, a)), s'\rangle$$

  where the letter $a \in \Sigma$ and the suffix $s' \in S$ are uniquely defined by either $s = as'$ or $s = \varepsilon$ and $as' = v$.
- To specify the type function τ', we first define types τ_i for all $i \geq 1$: We choose $w_i \in \Sigma^*$ such that $uv^i w_i \in L \setminus L_r$ and $|w_i| > r$. Then we let

$$\tau_i(\langle q, s\rangle) := \tau(q \cdot (\varepsilon, \ldots, \varepsilon, svw_i)).$$

  Since there are only 2^n different possibilities for the value $\tau_i(\langle q, s\rangle)$, at least one of them has to appear infinitely often—among these types, we will choose one arbitrarily and we call it the *repeated type* $\rho(\langle q, s\rangle)$). We define the type τ' by restriction to the first $n-1$ components.

$$\tau'(\langle q, s\rangle) := \rho(\langle q, s\rangle)|_{1,\ldots,n-1}.$$

  (The vertical bar means restriction to the components $1, \ldots, n-1$.)

The $(n-1)$–DFA A' defined above cannot realize any $(1, n-1)$–recognition of L, since n was chosen to be minimal. Hence there are words $u_1, \ldots, u_{n-1} \in \Sigma^*$ such that the Boolean vectors $\tau'(q'_0 \cdot (u_1, \ldots, u_{n-1}))$ and $(\chi_L(u_1), \ldots, \chi_L(u_{n-1}))$ differ in all $n-1$ components.

Now let $m = \max\{|u_i| \mid 1 \leq i \leq n-1\}$ and let α_m be the prefix of uv^ω of length m. Then we have:

$$\begin{aligned}
&\tau'(\langle q_0, uv\rangle \cdot (u_1, \ldots, u_{n-1})) \\
&\quad = \tau'(\langle q, s\rangle) \qquad \text{(where } q \in Q,\ s \in S, \text{ and } \alpha_m s = uv^k \text{ for some } k \geq 1) \\
&\quad = \tau(q \cdot (\varepsilon, \ldots, \varepsilon, svw_i))|_{1,\ldots,n-1} \qquad \text{(for some } i \geq k+1) \\
&\quad = \tau(q_0 \cdot (u_1, \ldots, u_{n-1}, \alpha_m svw_i))|_{1,\ldots,n-1} \\
&\quad = \tau(q_0 \cdot (u_1, \ldots, u_{n-1}, uv^{k+1} w_i))|_{1,\ldots,n-1} \\
&\quad = \tau(q_0 \cdot (u_1, \ldots, u_{n-1}, uv^i w_i))|_{1,\ldots,n-1} \qquad \text{(because of Eqn. 1)} \\
&\quad = (b_1, \ldots, b_{n-1}, b_n)|_{1,\ldots,n-1} \qquad \text{(for some values } b_i \in \mathbb{B})
\end{aligned}$$

Since $b_i \neq \chi_L(u_i)$ for all $i \in \{1, \ldots, n-1\}$, within the original automaton A, the last output b_n must denote the correct answer, i.e., $b_n = \chi_L(uv^i w_i)$. But w_i was chosen such that $uv^i w_i \in L$, hence $b_n = 1$.

Additionally, our choice of w_i guarantees its length to be greater than r. Thus for some decomposition of uv^iw_i into xyz with $1 \leq |y|$ and $|yz| \leq r$, the omission or multiple repetition of y (i.e., the pumping of y) leads to the same state in Q since $m \leq |x|$. This means we always have $b_n = 1$ in the repeated type which is correct, since all answers $b_1, \ldots, b_{n-1}$ remain wrong, hence $xy^*z \subseteq L$.

By definition, the word xyz is contained in L_r, but we have chosen w_i such that $xyz = uv^iw_i \in L \setminus L_r$. This is a contradiction, hence $\mathrm{Pref}(L \setminus L_r)$ cannot contain any infinite regular subset.

□

4 Closure under Union

It is well-known that the class Ω of all (m,n)-computable sets $(1 \leq m \leq n)$ forms a Boolean algebra [10, Thm. 2.3]. Since the proof is essentially a purely combinatorial argument, it can be translated to regular frequency computations. We give a direct proof here, using Ramsey's Theorem.

Theorem 2. *The class FREQ-REG is a Boolean algebra, i.e., it is closed under complementation and finite union.*

Proof. It is enough to show that for two languages $L, K \in (1,n)\mathrm{REG}$, their union $L \cup K \in (1,N)\mathrm{REG}$ for some $N \geq 1$. The value of N depends on some suitable Ramsey number and will be rather huge. For the moment we let $N \geq n$ to be any number.

We assume that L and K are $(1,n)$-recognizable via some automata A and B, respectively. The N-DFA we are going to construct works on an input $(u_1, \ldots, u_N)$. We view $\{1, \ldots, N\}$ as the set of vertices of a complete hypergraph with edge set $\binom{N}{n}$, where $\binom{N}{n}$ is the family of subsets of $\{1, \ldots, N\}$ having cardinality n. Each hyperedge $I \in \binom{N}{n}$ is colored by some vector $(c,d) \in \mathbb{B}^n \times \mathbb{B}^n$, and the color depends on the input $(u_1, \ldots, u_N)$ in the following way. Let $I = \{i_1, \ldots, i_n\}$, $i_1 < \ldots < i_n$. Then c is the output vector of A when reading $(u_{i_1}, \ldots, u_{i_n})$ as input, and d is defined analogously via B.

We have 4^n colors and having read $(u_1, \ldots, u_N)$, we can store the colored hypergraph in some finite control, i.e., we remember the graph in some state. Using Ramsey's Theorem (see e.g. [6]) we can choose N large enough such that the existence of some monochromatic subgraph G of size $4n-2$ is guaranteed. This means $G \subseteq \{1, \ldots, N\}$, $|G| = 4n-2$, and each $I \in \binom{N}{n}$ with $I \subseteq G$ has the same color (c,d). The finite control knows G since it knows the entire hypergraph.

Let $k = 4n-2$ and $G = \{i_1, \ldots, i_k\}$ with $1 \leq i_1 < \ldots < i_k \leq N$. We now produce an output vector $(b_1, \ldots, b_N) \in \mathbb{B}^N$ as follows:

$$b_\ell = \begin{cases} 1 & \text{if } \ell = i_{2j} \text{ and } 1 < 2j \leq k \\ 0 & \text{otherwise} \end{cases}$$

We claim that at least one output bit b_ℓ for $\ell \in G$ is correct. In particular, we don't care for the values b_ℓ with $\ell \notin G$. For simplification of notation, from now on we assume $G = \{1, \ldots, k\}$.

By contradiction assume that all b_ℓ were wrong ($1 \le \ell \le k$). Then, in particular, $u_1, u_3, \ldots, u_{4n-3} \in L \cup K$ and $u_2, u_4, \ldots, u_{4n-2} \notin L \cup K$. For the first set of inputs, at least half of them belong to L or at least half of them belong to K. Thus, without loss of generality, we can assume $u_1, u_3, \ldots, u_{2n-1} \in L$. The second set of inputs neither belongs to L nor to K, thus in particular we have $u_2, u_4, \ldots, u_{2n} \notin L$.

Remember that the color of each hyperedge in G is some fixed (c, d). Let $c = (c_1, \ldots, c_n)$, then we find a hyperedge $I = \{i_1, \ldots, i_n\}$, $1 \le i_1 < \cdots < i_n \le k$ such that $b_{i_\ell} = c_\ell$ for all $1 \le \ell \le n$. This is clear, simply because $(b_1, \ldots, b_k) = (0, 1, 0, 1, \ldots, 0, 1)$. However, at least one value c_ℓ is correct. Assume that $c_\ell = 1$ is correct for some $1 \le \ell \le n$. Then $b_{i_\ell} = 1$ and i_ℓ is even. But then $u_{i_\ell} \notin L$ which is a contradiction. If $c_\ell = 0$ is correct, then i_ℓ is odd (and $1 \le i_\ell \le 2n-1$), and we obtain a contradiction by $u_{i_\ell} \in L$. □

Remark 3. The construction above is not optimal in the sense that there is an overestimation in N. For example, assume that there is a single hyperedge which is colored by (c, d) with $c = (1, \ldots, 1)$. Then we can simply output 1 everywhere and at least one bit is correct. So it is not necessary to consider all 4^n possible colors. This also has an influence on the choice of k in the proof above. However, these are minor details which will not lead to a substantially smaller value of N. On the other hand, Tantau observed that, transferring techniques from Beigel [2], one can show that the union of two languages from $(1,n)$REG is contained in $(1,(2n)^2)$REG (personal communication). We will not give the details of his proof, which would require the introduction of several new notions. Moreover, our construction satisfies another nice property. Having read $(u_1, \ldots, u_N)$ we can point to $2n$ components, and among these $2n$ components we know that at least one output bit is correct.

For the concrete examples of Kinber languages or interval languages, direct constructions yield much better results. These two examples are presented below.

Example 4. As a first example, consider the union of k different Kinber languages as defined in Example 2, i.e., let $\alpha_1, \ldots, \alpha_k \in \{0,1\}^\omega$ and consider the language

$$K = K_{\alpha_1} \cup \ldots \cup K_{\alpha_k}.$$

This language can be $(1, k+1)$-recognized by the following $(k+1)$-DFA A. Let $u_1, \ldots, u_{k+1}$ be an arbitrary input. If all inputs are pairwise incomparable with respect to the prefix relation, the automaton will output 0^{k+1}—this is correct since for any two inputs u_i and u_j $(i \neq j)$, it can never be that $u_i \in K_{\alpha_\ell}$ and $u_j \in K_{\alpha_\ell}$ at the same time; hence, at most k of the $k+1$ inputs can belong to the language K.

If one of the inputs is a prefix of another, i.e., $u_i < u_j$ for some $i \neq j$, than u_i and u_j are answered with 1 and 0, respectively. It is easy to check that at least one of these answers is correct.

Thus K can be $(1, k+1)$-recognized.

Example 5. As our second example, we consider so-called interval languages (which are obtained by union and complementation of languages L_x as specified in Example 1). Let $I = [0, 1)$ be a half-open interval and $0 < x_1 < \ldots < x_k < 1$ be $k \geq 1$ real numbers within this interval (additionally, we define $x_{k+1} = 1$). Then the interval language $L_{x_1,\ldots,x_k}$ is defined as

$$L_{x_1,\ldots,x_k} = \bigcup_{i=1}^{\lceil \frac{k}{2} \rceil} \{w \in \{0,1\}^* \mid x_{2i-1} \leq 0.w < x_{2i}\}$$

We show that each such interval language $L_{x_1,\ldots,x_k}$ is $(1, k+1)$-recognizable.

We are given $k+1$ inputs. The automaton $(1, k+1)$-recognizing $L_{x_1,\ldots,x_k}$ can determine the relative order of the inputs, and assign the output vector $(0, 1, 0, 1, \ldots, 0, 1(, 0))$ (to the ordered input). We call this automaton A, and have to show the correctness of its output.

As the interval I is divided into $k+1$ subintervals and we have $k+1$ inputs, there are two possibilities:

1. All inputs are in different intervals. Then $0 \leq u_1 < x_1 \leq u_2 < x_2 \leq \ldots \leq u_{k+1} < x_{k+1} = 1$, and *all* inputs are decided correctly by the automaton.
2. Two inputs are in the same interval, especially there is a j such that u_j and u_{j+1} are in the same interval. Then the automaton decides exactly one of them correctly.

Acknowledgements

We would like to thank Klaus–Jörn Lange, Frank Stephan and Till Tantau for helpful discussions and various comments on the first version of this paper.

References

1. Holger Austinat, Volker Diekert, Ulrich Hertrampf, and Holger Petersen. Regular frequency computations. In *Proceedings of the RIMS Symposium on Algebraic Systems, Formal Languages and Computation*, Kyoto, Japan, 2000. To appear.
2. Richard Beigel. *Query-Limited Reducibilities.* PhD thesis, Stanford University, January 1995.
3. Richard Beigel, Martin Kummer, and Frank Stephan. Quantifying the amount of verboseness. *Information and Computation*, 118:73–90, 1995.
4. Jean Berstel. *Transductions and context-free languages.* Teubner Studienbücher, Stuttgart, 1979.
5. Alexandr N. Degtev. On (m, n)–computable sets. In *Algebraic Systems*, pages 88–99. Ivanova Gos. University, 1981. (In Russian.)
6. Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey Theory.* Wiley Interscience Series in Discrete Mathematics. Wiley, New York, 1980.
7. Maren Hinrichs and Gerd Wechsung. Time bounded frequency computations. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity*, pages 185–192, Ulm, Germany, 1997.

8. Efim B. Kinber. Frequency computations in finite automata. *Kibernetika*, 2:7–15, 1976. (In Russian; English translation in *Cybernetics*, 12:179–187, 1976.)
9. Martin Kummer and Frank Stephan. The power of frequency computation. In *Proceedings of the Tenth International Congress on Fundamentals of Computation Theory–FCT 1995*, LNCS 969, pages 323–332, 1995.
10. Martin Kummer and Frank Stephan. Recursion theoretic properties of frequency computation and bounded queries. *Information and Computation*, (120):59–77, 1995. Extended Abstract appeared in Proceedings of the Third Kurt-Gödel-Colloquium, Springer, LNCS 713, pages 243–254, 1993.
11. Robert McNaughton. The theory of automata, a survey. *Advances in Computers*, 2:379–421, 1961.
12. Gene F. Rose. An extended notion of computability. In *Proceedings of the International Congress for Logic, Methodology, and Philosophy of Science*, Stanford, California, 1960.
13. Boris A. Trakhtenbrot. On the frequency computation of functions. *Algebra i Logika*, 2:25–32, 1963. (In Russian.)