Regular Approximations of Non-Regular Languages*

Brendan Cordy and Kai Salomaa
School of Computing, Queen's University
Kingston, Ontario K7L 3N6, Canada
{brendan,ksalomaa}@cs.queensu.ca

Abstract

We approximate context-free, or more general, languages using finite automata. The degree of approximation is measured, roughly speaking, by counting the number of incorrect answers an automaton gives on inputs of length m and then looking how the values behave for large m. More restrictive variants are obtained by requiring that the automaton never accepts words outside the language (bottom approximation) or that it accepts all words in the language (top approximation). A further distinction is whether a given (context-free) language has a regular approximation which is optimal under the measure of approximation degree or an approximation which is arbitrarily close to optimal.

1 Introduction

Different ways of measuring levels of reliability of finite automata have been considered in [7, 8]. By allowing a finite automaton to give incorrect answers on some inputs we may obtain significant savings in the state-complexity of a language. Here, instead of comparing state-complexity of representations with different reliability, we consider various ways to give approximate representations of non-regular languages using finite automata. Strong non-approximability results for certain non-regular languages have been previously obtained in [5].

For the purpose of evaluating how well a language is approximated by another language, we need a way to measure the similarity of two languages. The existence of "good" approximations for non-regular languages naturally depends very much on the measures used to compare languages. We present examples for different types of measurements where good regular approximations exist or do not exist, respectively. We use variants of the measures considered in [5, 7, 8]. The measures, roughly speaking, count the number of inputs of a given length for which a finite automaton gives an incorrect answer, and then look how this value behaves for long inputs in the limit. We make some modifications to the earlier definitions, mainly, in order to avoid trivial good approximations. This issue did not occur in [5, 7, 8], perhaps, because the work there was concerned on the one hand with negative approximability results, that is results about the non-existence of approximations, and on the other hand with worst-case results for the descriptional

*Research supported by the Natural Sciences and Engineering Research Council of Canada.

complexity of approximations having different reliability. Thus, the work in [5, 7, 8] was for the main part not dealing with positive results on the existence of approximations.

Other types of regular approximations of languages have been considered e.g. in [1, 2, 3, 4, 11, 12, 13]. The automaticity descriptional complexity measure [13] and the model of cover automata [3] count the number of states of finite automata that recognize approximations of the language to be represented. Automaticity is a descriptional complexity measure for arbitrary languages whereas cover automata are used as an implementing method to reduce the size of automata that represent finite languages. These models differ from our approach in that the approximations are required to be correct for all words up to length n (with variable n) and do not consider the number of incorrect answers on words longer than n. The paper [4] investigates minimal covers and the approach is quite different from the measures considered here, as well. The metric used in [1] is somewhat similar to the measure we consider here, however, in [1] the metrics are required to have an additivity property with respect to catenation. This property guarantees that the metric preserves regularity of languages which means it would not be useful for our current purpose. The work [11, 12] investigates lower and upper rough set approximations that converge to a given language L. It turns out that in most cases the lower and upper approximations of context-free languages are regular.

Work along more practical lines on regular approximations of context-free languages can be found in [9, 10].

2 Degree of approximation

In the following Σ denotes a finite alphabet and Σ^* is the set of all strings over Σ. The length of a string w is the number of occurrences of symbols of Σ in w. The reversal (or mirror-image) of $w \in \Sigma^*$ is denoted w^R. For $L \subseteq \Sigma^*$ and $m \geq 0$, we denote by $|L|_m$ the number of strings of length m in L. The symmetric difference of sets A and B is $A \triangle B$. The cardinality of a finite set is $|A|$.

A deterministic finite automaton (DFA) is a tuple $M = (\Sigma, Q, q_0, Q_F, \delta)$ where Σ is the input alphabet, Q is the finite set of states, $q_0 \in Q$ is the start state, $Q_F \subseteq Q$ is the set of accepting states, and $\delta : Q \times \Sigma \longrightarrow Q$ is the state transition function. The function δ is extended in the natural way to a function $Q \times \Sigma^* \longrightarrow Q$ and the language recognized by the DFA M is $L(M) = \{w \in \Sigma^* \mid \delta(q_0, w) \in Q_F\}$. Note that a DFA as defined above is complete, i.e., $\delta(q, \sigma)$ is defined for all $q \in Q$, $\sigma \in \Sigma$. The deterministic finite automata recognize exactly all the regular languages. For all unexplained notions in language theory we refer the reader e.g. to [6, 14].

For integers $m, n \geq 0$, we define the "non-zero minimum" of m and n as

$$\min_1(m, n) = \max(\min(m, n), 1).$$

When a finite automaton A is used as an approximation of a language L, the value $\mid L(A) \triangle L \mid_m$ gives the number of strings of length m for which A gives an incorrect answer. More generally, if R and L are two languages over Σ, we define the *degree of approximation of L by R* as the quantity

$$App(R, L) = \limsup_{m \to \infty} \frac{\mid R \triangle L \mid_m}{\min_1(\mid L \mid_m, \mid \Sigma^* - L \mid_m)} \tag{1}$$

It can be noted that the degree-of-approximation relation is not symmetric. In our terminology, a degree-of-approximation value (close to) zero means a "good approximation".

The definition of $App(R, L)$ does not require that R is regular, however, in the following we are mainly concerned with cases where the language used as an approximation is regular.

The relation (1) resembles the reliability measure of [8] or the approximation measure used in [5]. These measures use the number of all words of length m, $\mid \Sigma^* \mid_m$ (or equivalently $\mid \Sigma \mid^m$), as the denominator. The main reason for introducing the definition (1) is that if we would use

$$\limsup_{m \to \infty} \frac{\mid R \triangle L \mid_m}{\mid \Sigma^* \mid_m}$$

as the right side of (1) all very "dense" languages would always have Σ^* as a good approximation and, similarly, all "sparse" languages could be approximated by $\emptyset$. Above by a dense (respectively, sparse) language we mean a language L such that the limit of $\mid \Sigma^* - L \mid_m$ (respectively, $\mid L \mid_m$) divided by $|\Sigma|^m$ approaches zero when m approaches infinity.

In fact, the degree of approximation can be thought of as a measure of how well some language R approximates L relative to the best trivial approximation ($\emptyset$ or Σ^*). If we were to use the maximum of $\mid L \mid_m$ and $\mid \Sigma^* - L \mid_m$ in the denominator in place of the minimum, this could be viewed as comparing an approximation relative to the worse of the two trivial approximations, ($\emptyset$ or Σ^*).

In (1) it is possible that $\min(\mid L \mid_m, \mid \Sigma^* - L \mid_m)$ in the denominator becomes zero. In particular, if L contains no words of length m (or all words of length m) where m ranges over infinitely many non-periodic values, this could cause the approximation degree to become infinite for any regular language. This could happen even if L is always very sparse, in which case $\emptyset$ would intuitively be a "reasonably good" approximation of L. For the above reason we use the "non-zero minimum" $\min_1(\cdot, \cdot)$ in the denominator.

First we consider the question of what values the approximation degree can have. The below example gives a construction showing that $App(R, L)$ with R regular and L context-free can have any rational value between 0 and 1.

Example 2.1 Let $\Sigma = \{0, 1, b, c\}$. For any integers $0 \leq i \leq n$ we construct a regular language R and a context-free language L over Σ such that $App(R, L) = \frac{i}{n}$.

Let $f : \{1, \ldots, n\} \longrightarrow \{0, 1\}^r$ be a bijective mapping where $r = \lceil \log n \rceil$. We define

$$L = \bigcup_{i=1}^{n} f(i)\{b^j c^k \mid j \neq k, \quad j, k \geq 0\}$$

All strings in L begin with a sequence of r symbols 0 and 1, followed by a string of b's and c's.

Let $0 \leq k \leq n$ and define

$$R_k = \bigcup_{i=1}^{k} f(i) b^* c^*$$

Let $u = r + 2m$, $m \geq 1$. All strings of L having length u begin with a sequence of r symbols 0 and 1, and followed by a string of i symbols b and j symbols c where $i + j = 2m$

and $i \neq j$. Thus,

$$| L |_u = n \cdot (2m) \tag{2}$$

The set $| L \triangle R_k |_u$ has all strings of L of length u beginning with $f(i)$, $k < i \leq n$, and for any $1 \leq i \leq k$ it contains one string $f(i)b^j c^j$, where $j = m$. Thus,

$$| R_k \triangle L |_u = (n-k) \cdot (2m) + k. \tag{3}$$

Next let $v = r + 2m - 1$, $m \geq 1$. Strings of L having length v again begin with a sequence of r symbols 0 and 1, followed by all strings in b^*c^* having length $2m-1$. Thus,

$$| L |_v = n \cdot (2m) \tag{4}$$

Also, similarly as above it is easy to see that

$$| R_k \triangle L |_v = (n-k) \cdot (2m). \tag{5}$$

Always when $i > r$, $| L |_i < | \Sigma^* |_i$. Hence using (2), (3), (4), and (5), we see that the $(r+2m)^{th}$ term in the limit on the right side of (1) is $\frac{2m(n-k)+k}{2mn}$ and the $(r+2m-1)^{th}$ term is $\frac{2m(n-k)}{2mn}$. Thus as the limit we obtain

$$App(R_k, L) = \frac{n-k}{n}.$$

Since k can be an arbitrary integer between 0 and n, the claim follows. ■

Similar to Example 2.1 we can, of course, construct languages R and L such that $App(R, L)$ is any rational number greater than one. This can be done by making $R - L$ sufficiently large, and this would correspond to situations where R is a bad approximation of L.

Next we define what we mean by "good" regular approximations of a language L. We distinguish the notions of *top approximation* that contains L, *bottom approximation* that is contained in L and *mixed approximation* (or just approximation) that can have any relation with L.

Definition 2.1 *Let $L \subseteq \Sigma^*$. The language L has a (good)* regular m-approximation *(mixed approximation) if there exists a regular language R such that $App(R, L) = 0$. We say that L has a* regular t-approximation, *or top approximation, (respectively,* b-approximation, *or bottom approximation) if above R can be chosen such that $L \subseteq R$ (respectively, $R \subseteq L$).*

The language L is said to have a regular m-approximation in the limit *if for any $\epsilon > 0$ there exists a regular language R_ϵ such that $App(R_\epsilon, L) < \epsilon$. Again we say that L has a regular* t-approximation *(respectively,* b-approximation*)* in the limit *if above R_ϵ can always be chosen such that $L \subseteq R_\epsilon$ (respectively, $R_\epsilon \subseteq L$).*

In the following, we call regular x-approximations simply x-approximations, where $x \in \{m, b, t\}$. Note that any t-approximation or b-approximation is by definition also an m-approximation.

The following negative approximation result for the language

$$L_{majority} = \{w \in \{a,b\}^* \mid w \text{ has more } a\text{'s than } b\text{'s } \}$$

has been established in [5]. We state the result using our definitions. Note that having an m-approximation in the limit is the weakest type of approximability property given in Definition 2.1.

Proposition 2.1 [5] *The language $L_{majority}$ does not have an m-approximation in the limit.*

Proof. Let $\Sigma = \{a,b\}$. In [5] it is shown that for any regular language $R \subseteq \Sigma^*$,

$$\lim_{n\to\infty} \frac{\mid R \triangle L_{majority} \mid_n}{\mid \Sigma^* \mid_n} = \frac{1}{2}$$

Since for any $n \geq 0$,

$$\mid L_{majority} \mid_n \leq \frac{\mid \Sigma^* \mid_n}{2}$$

it follows that

$$\limsup_{n\to\infty} \frac{\mid R \triangle L_{majority} \mid_n}{\mid L_{majority} \mid_n} \geq 1,$$

and by definition (1) it follows that $App(R, L_{majority}) \geq 1$. This holds for any regular language R and, consequently, $L_{majority}$ does not have an m-approximation even in the limit. ■

3 Results

First we illustrate that there exist languages having regular top approximations but no regular bottom approximations and vice versa. We consider the linear context-free language

$$T = \{a^i b^j \mid i,j \geq 0, i \neq j\} \tag{6}$$

Example 3.1 We show that $R_t = a^*b^*$ is a t-approximation of T. We note that $T \subseteq R_t$ and

$$\mid R_t \triangle T \mid_m = \begin{cases} 0 \text{ if } m \text{ is odd,} \\ 1 \text{ of } m \text{ is even.} \end{cases}$$

The number of strings in T of length m is either m or $m+1$ and clearly $\mid T \mid_m < \mid \Sigma^* - T \mid_m$, for all $m \geq 2$. Thus

$$App(R_t, T) = \limsup_{m\to\infty} \frac{1}{\mid T \mid_m} \leq \limsup_{m\to\infty} \frac{1}{m} = 0.$$

■

Lemma 3.1 *The language T from (6) has no b-approximation.*

Proof. Let R_b be any regular language such that $R_b \subseteq T$. It is sufficient to show that $App(R_b, T) > 0$.

Let $\Sigma = \{a, b\}$ and $M = (\Sigma, Q, q_0, Q_F, \delta)$ be a complete DFA that recognizes R_b.

Consider computations of M on strings in a^* and find the first state that repeats. That is, we find $0 \leq i < |Q|$ and $0 < j \leq |Q|$ such that $\delta(q_0, a^i) = p_1$ and $\delta(p_1, a^j) = p_1$. Since M is complete, the state p_1 always exists.

Now we consider computations of M starting from state p_1 on strings in b^* and find the first cycle. We find $0 \leq k < |Q|$ and $0 < m \leq |Q|$ such that $\delta(p_1, b^k) = p_2$ and $\delta(p_2, b^m) = p_2$.

We choose an integer r as follows. If $k \leq i$, then $r = i - k$. If $k > i$, then we choose r to be the smallest integer such that

$$j \mid k - i + r \tag{7}$$

The integer r can always be found such that $r \leq |Q|$.

Now there exists $p_3 \in Q$ such that for all $x, y \geq 0$, we have

$$\delta(q_0, a^{i+xj} b^{k+ym+r}) = p_3. \tag{8}$$

By (7), we can choose x such that $i + xj = k + r + 0 \cdot m$. Thus if p_3 is an accepting state, $R_b = L(M)$ contains some string not in T and R_b is not a b-approximation.

Hence p_3 cannot be an accepting state and from (8) we know that R_b does not contain any strings of the form $a^{i+xj} b^{k+ym+r}$, where $x, y \geq 0$.

Let z be an arbitrary positive integer and denote

$$n = i + k + r + z \cdot j \cdot m.$$

We know that T contains, in total, n or $n + 1$ strings of length n. By (8) we know that the following strings of length n cannot be in R_b:

$$a^i b^{k+r+zjm}, \quad a^{i+jm} b^{k+r+(z-1)jm}, \quad \ldots, \quad a^{i+zjm} b^{k+r},$$

and it follows that $| R_b \triangle T |_n \geq z + 1$. Since for large n, $| T |_n < | \Sigma^* - T |_n$, we get

$$\begin{aligned} App(R_b, T) &= \limsup_{n \to \infty} \frac{| R_b \triangle T |_n}{| T |_n} \geq \limsup_{n \to \infty} \frac{| R_b \triangle T |_n}{n} \\ &\geq \limsup_{z \to \infty} \frac{z + 1}{i + k + r + z \cdot j \cdot m + 1} = \frac{1}{jm} > 0. \end{aligned}$$

■

The above proof shows that the value of $App(R_b, T)$ is positive, but does not give for it any positive lower bound. Indeed it is easy to see that the language T has a b-approximation in the limit. If a DFA M_k checks that the input is of the form $a^i b^j$ where $i \not\equiv j$ modulo some of the integers $2, \ldots, k$, then $L(M_k) \subseteq T$ and for large enough k, $App(L(M_k), T)$ becomes smaller than any given positive constant.

Above we have seen that the languge T has a t-approximation but no b-approximation. This naturally begs the question whether there exists a language with a b-approximation but no t-approximation. The following correspondence between b- and t-approximations turns out to be useful.

Proposition 3.1 *Let L be a language over the alphabet Σ.*

(i) *If L has a t-approximation, then $\Sigma^* - L$ has a b-approximation.*

(ii) *If L has a b-approximation, then $\Sigma^* - L$ has a t-approximation.*

Proof. Assume that R_t is a regular language such that $L \subseteq R_t$ and $App(R_t, L) = 0$.
We observe that $(\Sigma^* - L) \triangle (\Sigma^* - R_t) = L \triangle R_t = R_t - L$. Thus,

$$\begin{aligned} App(\Sigma^* - R_t, \Sigma^* - L) &= \limsup_{m\to\infty} \frac{|\ (\Sigma^* - R_t) \triangle (\Sigma^* - L)\ |_m}{\min_1(|\ \Sigma^* - L\ |_m, |\ L\ |_m)} \\ &= \frac{|\ R_t \triangle L\ |_m}{\min_1(|\ L\ |_m, |\ \Sigma^* - L\ |_m)} = App(R_t, L) = 0. \end{aligned}$$

This proves (i) since $\Sigma^* - R_t \subseteq \Sigma^* - L$ and $\Sigma^* - R_t$ is regular. The case (ii) is completely symmetric. ■

Now Example 3.1, Lemma 3.1 and Proposition 3.1 give the following.

Corollary 3.1 *With T as in (6), the language $\Sigma^* - T$ has a b-approximation but no t-approximation.*

Using a direct analysis it can be shown that also the language $\{a^i b^i \mid i \geq 0\}$ has no t-approximation. This estimation will be done below in the proof of Lemma 3.2.

Next we address the question whether there exist languages with m-approximations but no b- or t-approximations. In the following let $\Sigma = \{a, b, c, d\}$ and denote

$$X = \{a^i b^i \mid i \geq 0\} \cup \{c^i d^j \mid i \neq j,\ \ i, j \geq 0\} \tag{9}$$

Example 3.2 The language c^*d^* is an m-approximation of X. To see this we observe that for any even length $2n$, c^*d^* contains all strings of X except $a^n b^n$, and for any odd length $2n+1$, c^*d^* contains all strings of X. Similarly, for any even length $2n$, c^*d^* contains the string $c^n d^n$ not in X and for any odd length $2n+1$, all strings of length $2n+1$ in c^*d^* are also in X. Since for all non-negative integers n, $|\ X\ |_n = n + 1$, we get

$$App(c^*d^*, X) \leq \limsup_{n\to\infty} \frac{2}{n+1} = 0.$$

■

Lemma 3.2 *The language X as in (9) does not have a b-approximation or a t-approximation.*

Proof. First we show that X does not have a b-approximation. Let R_b be any regular language such that $R_b \subseteq X$ and let M be a complete DFA having q states, that recognizes R_b. Exactly as in the proof of Lemma 3.1 we can find integers $i, j, k, m, r \leq q$, where j divides $(k - i + r)$ such that M reaches the same state p after reading any string $c^{i+xj} d^{k+ym+r}$ independently of the values $x, y \geq 0$, and it is then observed that p cannot be an accepting state. Then similarly as in the proof of Lemma 3.1 we can calculate that

$App(R_b, X)$ is at least $\frac{1}{j \cdot m}$. The only difference in the estimation is that now, for any length n, X contains exactly $n+1$ strings of length n, and the value $n+1$ was used in the estimation for the lower bound in the proof of Lemma 3.1.

Next we show that the language X cannot have a t-approximation. Assume that R_t is a regular language, $X \subseteq R_t$ and let $M = (\Sigma, Q, q_0, Q_F, \delta)$ be a complete DFA recognizing R_t. We consider computations of M on strings of a^* and find the first cycle. That is, we find $0 \leq i < |Q|$, $1 \leq m \leq |Q|$ such that for some $p_1 \in Q$, $\delta(q_0, a^i) = p_1$ and $\delta(p_1, a^m) = p_1$.

Then we consider computations of M starting in state p_1 on strings in b^* and find the first cycle. That is, we find $0 \leq j < |Q|$ and $1 \leq n \leq |Q|$ such that for some $p_2 \in Q$, $\delta(p_1, b^j) = p_2$ and $\delta(p_2, b^n) = p_2$.

Thus for all $x, y \geq 0$,

$$\delta(q_0, a^{i+xm} b^{j+yn}) = p_2.$$

For large values of x, $i + xm > j$. Since $X \subseteq L(M)$ and X contains all strings $a^r b^r$, $r \geq 0$, this implies that an accepting state must be reachable from p_2 by reading a string of b's. Thus there exists $k < |Q|$ and $p_3 \in Q_F$ such that for all $x, y \geq 0$,

$$\delta(q_0, a^{i+xm} b^{j+k+yn}) = p_3. \tag{10}$$

Let z be a positive integer and denote

$$n_z = i + j + k + z \cdot m \cdot n.$$

By (10) we know that M must accept the following strings of length n_z,

$$a^i b^{j+k+zmn}, \quad a^{i+mn} b^{j+k+(z-1)mn}, \quad \ldots, \quad a^{i+zmn} b^{j+k}.$$

At most one of the above $z+1$ strings can be in X and hence we conclude that for any integer n_z, $| L(M) \triangle X |_{n_z} \geq z$. Now we get

$$\begin{aligned} App(R_t, X) &= \limsup_{m \to \infty} \frac{| R_t \triangle X |_m}{\min_1(| X |_m, | \Sigma^* - X |_m)} \\ &\geq \limsup_{z \to \infty} \frac{| R_t \triangle X |_{n_z}}{\min_1(| L_0 |_{n_z}, | \Sigma^* - X |_{n_z})} \\ &\geq \limsup_{z \to \infty} \frac{z}{1 + i + j + k + zmn} = \frac{1}{mn} > 0. \end{aligned}$$

■

For easier readability, the language X used in Example 3.2 and Lemma 3.2 is defined over a four letter alphabet. By using a simple coding, exactly the same argument can be used to show that there is a language over a binary alphabet having an m-approximation but no b-approximation or t-approximation.

Next we show that there exist context-free languages that do not have m-approximations, or even m-approximations in the limit. A convenient language for this purpose is the set of marked palindromes:

$$L_0 = \{w\$w^R \mid w \in \{a, b\}^*\}$$

Lemma 3.3 *The language L_0 does not have an m-approximation in the limit.*

Proof. Assume to the contrary that there exists a regular approximation R for L_0 such that

$$App(R, L_0) = \limsup_{n\to\infty} \frac{|\ R \triangle L_0\ |_n}{|\ L_0\ |_n} < \frac{1}{2}.$$

Hence, there exists an integer n_0 such that we have

$$|\ R \triangle L_0\ |_n < \frac{1}{2} \cdot 2^{\lfloor \frac{n}{2} \rfloor} \tag{11}$$

for all $n \geq n_0$ where n is odd. Since if not, there is an infinite subsequence $(m_i)_{i=0}^{\infty}$ of odd lengths for which $|\ R \triangle L_0\ |_{m_i} \geq \frac{1}{2} \cdot 2^{\lfloor \frac{m_i}{2} \rfloor}$, and hence

$$\limsup_{n\to\infty} \frac{|\ R \triangle L_0\ |_n}{2^{\lfloor \frac{n}{2} \rfloor}} \geq \limsup_{i\to\infty} \frac{\frac{1}{2} \cdot 2^{\lfloor \frac{m_i}{2} \rfloor}}{2^{\lfloor \frac{m_i}{2} \rfloor}} = \frac{1}{2}.$$

Let $M = (\Sigma, Q, q_0, Q_F, \delta)$ be a DFA for R, and observe that, for odd $n \geq n_0$, M must accept at least half of the $2^{\lfloor \frac{n}{2} \rfloor}$ strings of length n in L_0. For if not, then R will violate (11) since M will reject at least half of the $2^{\lfloor \frac{n}{2} \rfloor}$ strings of length n in L_0.

In the following we consider n to be an arbitrary odd integer greater or equal to n_0. Let c be the number of states of Q. By the pigeonhole principle, for some state $q_\$ \in Q$ there are at least $\frac{1}{c} 2^{\lfloor \frac{n}{2} \rfloor - 1}$ strings of L_0 that are accepted through computations of M of the form

$$\begin{array}{ccccc} & s & \$ & s^R & \\ q_0 & \rightarrow & q_\$ & \rightarrow & q_f \end{array}$$

where $s \in \{a, b\}^{\lfloor \frac{n}{2} \rfloor}$. We denote

$$X_n = \{w \in \{a, b\}^{\lfloor \frac{n}{2} \rfloor} \mid \delta(q_0, w\$) = q_\$,\ \delta(q_\$, w^R) \in Q_F\}.$$

We have seen that X_n has at least $\frac{1}{c} 2^{\lfloor \frac{n}{2} \rfloor - 1}$ strings and, for any $s_1, s_2 \in X_n$, we know that M accepts $s_1 \$ s_2^R$. The number of new strings which can be created by combining different s_1's and s_2's is

$$\frac{1}{c} 2^{\lfloor \frac{n}{2} \rfloor - 1} \cdot \frac{1}{c} 2^{\lfloor \frac{n}{2} \rfloor - 1} = \frac{1}{c^2} 2^{n-3}.$$

(Recall that n is assumed to be odd.) For any particular string $s_1 \in X_n$ there is only one string $s_2 \in X_n$ such that the string $s_1 \$ s_2^R$ is in L_0. Hence the number of strings not in L_0 which M accepts is at least

$$\frac{1}{c^2} 2^{n-3} - \frac{1}{c} 2^{\lfloor \frac{n}{2} \rfloor - 1} \tag{12}$$

Since c is a constant depending on the number of states of M, it is clear that as n increases, the ratio of (12) and $2^{\lfloor \frac{n}{2} \rfloor}$ grows without bound, so since $|\ R \triangle L_0\ |_n$ is at least as large as (12),

$$\limsup_{n\to\infty} \frac{|\ R \triangle L_0\ |_n}{2^{\lfloor n/2 \rfloor}} = \infty.$$

We have produced a contradiction. ■

The proof of Lemma 3.3 is less involved than the proof required to show that $L_{majority}$ does not have an m-approximation in the limit [5]. This is due to the fact that L_0 is chosen to have, roughly, $2^{\frac{n}{2}}$ words of length n which makes a density argument straightforward to use.

Combining the results of Lemmas 3.1, 3.2 and 3.3, Corollary 3.1 and Examples 3.1 and 3.2 we can summarize the situation as follows.

Theorem 3.1 *There exist context-free languages L_t, L_b, L_m and L_0 such that*

(i) *L_t has a t-approximation but no b-approximation.*

(ii) *L_b has a b-approximation but no t-approximation.*

(iii) *L_m has an m-approximation but no b-approximation or t-approximation.*

(iv) *L_0 does not have any m-approximation, and not even an m-approximation in the limit.*

Since t- and b-approximations are always also m-approximations, Theorem 3.1 says, in particular, that for any combination of $x, y \in \{t, b, m\}$ such that $x \neq y$ and x-approximations are not a special case of y-approximations, there exists a context-free language L such that L has an x-approximation and L does not have an y-approximation.

In Lemma 3.1 we saw that the language T from (6) does not have a b-approximation, and after this result it was observed that T, on the other hand, has a b-approximation in the limit. Using the correspondence between b- and t-approximations as in Proposition 3.1 it follows that there exist languages that do not have a t-approximation but do have a t-approximation in the limit. Theorem 3.1 (iv) leaves open the question whether there exist languages having an m-approximation in the limit and do not have an m-approximation. When using a density argument, as in the proof of Lemma 3.3, to establish the non-existence of an m-approximation, it is not clear whether the same languages can have an m-approximation in the limit.

When considering the types of approximations introduced in Definition 2.1, a language has the strongest approximation properties when it has both a t-approximation and a b-approximation. The obvious question is then whether non-regular languages can have both t- and b-approximations. This is answered affirmatively by the following example.

Example 3.3 Let $\Sigma = \{a, b, c\}$ and

$$L_1 = \{a^{2^i} \mid i \geq 0\} \cup \{b, c\}^* \tag{13}$$

Denote $L_2 = \{b, c\}^*$ and $L_3 = a^* \cup \{b, c\}^*$. We show that L_2 is a b-approximation and L_3 is a t-approximation of L_1.

Since for all n, the value of $\mid L_1 \mid_n$ is 2^n or $2^n + 1$, and for $n \geq 4$,

$$\mid L_1 \mid_n \leq 2^n + 1 < 3^n - (2^n + 1) \leq \mid \Sigma^* - L_1 \mid_n.$$

As well, we have that because there is at most one string of the form a^{2^i} for any n, so $\mid L_2 \triangle L_1 \mid_n \leq 1$, and hence

$$App(L_2, L_1) \leq \limsup_{n\to\infty} \frac{\mid L_2 \triangle L_1 \mid_n}{2^n} \leq \limsup_{n\to\infty} \frac{1}{2^n} = 0.$$

We note that for all $n \geq 0$, $| L_3 \triangle L_1 |_n \leq 1$, and using the same estimation as above we see that also $App(L_3, L_1) = 0$. ■

In Example 3.3, the language L_1 is not context-free. Let $\Sigma = \{a, b, c, d\}$ and define

$$L_4 = \{a^i b^i \mid i \geq 0\} \cup \{c, d\}^*.$$

Using exactly the same estimation as in Example 3.3 we see that L_4 has both a t-approximation and a b-approximation.

Corollary 3.2 *There exist non-regular context-free languages that have both a t-approximation and a b-approximation.*

4 Conclusion

Here we have continued the work of [5, 7, 8] in attempting to classify different types of regular approximations for nonregular languages. Naturally much more remains to be done. For example, our results leave open the question whether there exists a (context-free) language L such that L does not have any m-approximation but L has an m-approximation in the limit. We have shown that for b-approximations and t-approximations such examples do exist.

Also, the corresponding decision problems would be of interest. Given a context-free language L, can we decide whether or not L has an m-approximation (respectively, b- or t-approximation)? It is likely that many of these questions are undecidable; nevertheless, it would be interesting to find that some cases can be decided effectively.

References

[1] C. Calude, K. Salomaa and S. Yu. Additive distances and quasi-distances between words. Journal of Universal Computer Science, 8 (2002), pp. 141–152.

[2] C. Câmpeanu and A. Păun. Tight bounds for the state complexity of deterministic cover automata. In Proceedings of Descriptional Complexity of Formal Systems, DCFS 2006, Las Cruces, NM, June 21–23, 2006.

[3] C. Câmpeanu, N. Sântean and S. Yu. Minimal cover automata for finite languages. Theoretical Computer Science, 267 (2001), pp. 3–16.

[4] M. Domaratzki, J. Shallit and S. Yu. Minimal covers of formal languages. In Developments in Language Theory, DLT 2001, Lecture Notes in Computer Science 2295, Springer, 2001, pp. 319–329.

[5] G. Eisman and B. Ravikumar. Approximate recognition of non-regular languages by finite automata. Australasian Computer Science Conference, ASCS 2005, pp. 219–228.

[6] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, 1979.

[7] M. Kappes and C. Kintala. Tradeoffs between reliability and conciness of deterministic finite automata. Journal of Automata, Languages and Combinatorics, 9 (2004), pp. 281–292.

[8] M. Kappes and F. Niessner. Succinct representations of DFA with different levels of reliability. Theoretical Computer Science, 330 (2005), pp. 299–310.

[9] M. Mohri and M.-J. Nederhof. Regular approximation of context-free grammars through transformation. In: Robustness in Language and Speech Processing, (J.-C. Junqua and G. van Noord, Eds.) Kluwer Academic Publishers, 2001, pp. 153–163.

[10] M.-J. Nederhof. Practical experiments with regular approximation of context-free languages. Computational Linguistics, 26 (2000), pp. 17–44.

[11] Gh. Păun, L. Polkowski and A. Skowron. Rough-set-like approximations of context-free and regular languages. In: Proceedings of IPMU–96: Information Processing and Management of Uncertainty on Knowledge Based Systems, July 1–5, 1996, Granada, Spain, Universidad de Granada, vol. II, pp. 891–895.

[12] Gh. Păun, L. Polkowski and A. Skowron. Rough set approximations of languages. Fundamenta Informaticae, 32 (1997), pp. 149–162.

[13] J. Shallit and Y. Breitbart. Automaticity I: Properties of a measure of descriptional complexity. Journal of Computer and System Sciences, 53 (1996), pp. 10–25.

[14] S. Yu. Regular languages. In: Handbook of Formal Languages, Vol. I (G. Rozenberg and A. Salomaa, Eds.) Springer, 1997, pp. 41–110.