What is a Maximal Error-Detecting Capability of a Formal Language?

Stavros Konstantinidis

Department of Mathematics and Computing Science
Saint Mary's University
Halifax, Nova Scotia B3H 3C3 CANADA
s.konstantinidis@smu.ca

Abstract

We introduce the concepts "maximal error-correcting capability" and "maximal error-detecting capability" of a given formal language (set of words), with respect to a certain class of combinatorial channels. A combinatorial channel is a set of pairs of words describing all the possible input/output channel situations. This paper is mainly intended to obtain basic general results on these new concepts and discuss possible research directions with an emphasis on the problem of computing maximal error-detecting (or -correcting) capabilities of a given regular language.

Key words: formal language, regular language, channel, combinatorial channel, error-correction, error-detection, maximal, algorithm, automaton, error model

1 Introduction

A (combinatorial) *channel* is a set of pairs of words describing all the possible input-output situations permitted by the channel. The fact that the pair (w, z) is in the channel means that the word z can be received via the channel when the word w is used as input. In this case, if $w \neq z$ then we say that w was received with errors. An *error model* is a set $\mathbb{C}$ of channels. Intuitively, $\mathbb{C}$ contains the possible channels that appear to model the error situations arising in some application where information needs to be transmitted or stored. Besides the error model, we consider a (formal) language – this is simply a set of words [6] – that is intended to be used for representing information. Depending on the application, it is required that the language satisfies certain properties, for instance, error-correction or error-detection with respect to a channel γ in $\mathbb{C}$. A language L is *error-detecting for* γ, [3], when no pair (w, z) in the channel is such that $w \neq z$ and both w and z are in L; that is, the channel cannot transform a word of L into another different word of L. This fact allows one to detect whether or not a transmitted word has been received with errors. The language L is *error-correcting for* the channel γ, [3], when no two different words of L can result via γ into the same output; that is, if (w_1, z) and (w_2, z) are in γ and w_1, w_2 are in L then w_1 must be the same as w_2. This fact allows one to correct a given channel output z to a unique word of L.

Note: *In previous papers we used $L \cup \{\lambda\}$ in the above definitions, where λ is the empty word. Here, to simplify matters we assume that the empty word λ is always a member of the language L. This assumption corresponds to the situation where no data is sent to the channel. We also note that the above definitions constitute natural generalizations of the corresponding definitions of error-detection/correction found in traditional coding theory – see [5], for instance, as well as the two examples below.*

Consider now the problem of testing whether a *given* language is error-correcting or error-detecting for a *given* channel. The answer to this problem often requires careful mathematical reasoning even for apparently simple languages and channels. On the other hand, the theory of automata provides tools that allow one to decide the above problem relatively easy. We want to exploit those tools in the context of error-correction and error-detection. In particular, consider the following problem:

Problem 1 *Let L be a language and let $\mathbb{C}$ be an error model. Compute a channel γ in $\mathbb{C}$ such that γ is a $\mathbb{C}$-maximal error-detecting (or error-correcting) capability of L.*

A channel γ is said to be a $\mathbb{C}$-maximal error-detecting capability of L if (i) γ is in $\mathbb{C}$, (ii) L is error-detecting for γ, and (iii) L is not error-detecting for γ', for any channel γ' that properly contains γ, that is, $\gamma' \supsetneq \gamma$.

In the sequel we shall focus on *regular languages*, that is, languages accepted by finite automata. Recall that a finite automaton has finitely many states, one of which is its initial state and some of them are its final states, and a set of transitions of the form (p, a, q). Such a transition says that if the automaton is in state p and the current input starts with a then it can enter the state q. The automaton can consume a given input word by following its transitions and, in this case, accepts the word when it is in some final state – see [7], for instance, for more details on finite automata and regular languages.

Example 1 A typical channel considered in coding theory (in many cases implicitly) is the channel $\sigma(m, \infty)$ consisting of all pairs of words (w, z) such that z can be received using at most m substitutions of symbols in w, that is, $H(w, z) \leq m$, where $H(w, z)$ is the Hamming distance between the words w and z. For example, $(0000, 0110) \in \sigma(2, \infty)$, where we assume that 0 and 1 are elements of the alphabet. In this case, we consider the error model

$$\mathbb{C}_\sigma[\infty] = \{\sigma(m, \infty) \mid m \geq 0\}$$

and the error-detection version of Problem 1 is equivalent to computing, for a given language L, the maximum value of m such that $H(L) > m$, where the quantity $H(L)$ is the smallest Hamming distance between any two different words in L – indeed, note that L is error-detecting for $\sigma(m, \infty)$ if and only if $H(L) > m$ [5]. This instance of the problem can be solved efficiently when L is any regular language [2]. Using this result one can also solve the error-correction version of the problem by noting that L is error-correcting for $\sigma(m,\infty)$ if and only if $H(L) > 2m$.

Example 2 Another typical channel is the channel $\mathrm{sid}(m,\infty)$ consisting of all pairs of words (w,z) such that z can be received using a total of at most m substitutions, insertions, and deletions of symbols in w, that is, $\Lambda(w,z) \leq m$, where $\Lambda(w,z)$ is the Levenshtein distance (also called edit distance) between the words w and z. In this case, we consider the error model

$$\mathbb{C}_{\mathrm{sid}}[\infty] = \{\mathrm{sid}(m,\infty) \mid m \geq 0\}$$

and the error-detection version of Problem 1 is equivalent to computing, for a given language L, the maximum value of m such that $\Lambda(L) > m$, where the quantity $\Lambda(L)$ is the smallest Levenshtein distance between any two different words in L – indeed, note that L is error-detecting for $\mathrm{sid}(m,\infty)$ if and only if $\Lambda(L) > m$. This instance of the problem can be solved in polynomial time when L is any regular language [4]. Using this result one can also solve the error-correction version of the problem by noting that L is error-correcting for $\mathrm{sid}(m,\infty)$ if and only if $\Lambda(L) > 2m$.

We argue now that Problem 1 is important. First, the problem is natural from a theoretical point of view. Moreover, the version of the problem described in Example 1 is the converse of the following classical problem of coding theory: Let m be the maximum number of substitution errors permitted by the channel in any word of length l. Construct a language L whose words are of uniform length l such that L is m-error-correcting. In addition, many *existing* languages, natural or artificial, are presently in use and it is desirable to know their error-handling capabilities. For example, one might consider a DNA language which is "processed" by various enzymes in determining the growth of the organism – here a DNA language is a set of words over the DNA alphabet $\{a,c,g,t\}$. Similarly, this approach could offer an alternative method of evaluating the error-handling capabilities of existing codes used in information transmission.

2 Basic Results, Rational Channels

We use Σ to denote a fixed, but arbitrary alphabet. Then the symbol Σ^* denotes the set of all words (or strings) over Σ. A binary relation ρ over Σ is a subset of $\Sigma^* \times \Sigma^*$. The inverse of ρ is the relation $\rho^{-1} = \{(u,v) \mid (v,u) \in \rho\}$. If L is a language, then $\rho \downarrow L$ is the relation $\rho \cap (L \times \Sigma^*) = \{(x,y) \in \rho \mid x \in L\}$, and $\rho \uparrow L$ is the relation $\rho \cap (\Sigma^* \times L) = \{(x,y) \in \rho \mid y \in L\}$. Note that $\rho \uparrow L = (\rho^{-1} \downarrow L)^{-1}$.

A *channel* is a domain preserving binary relation over some alphabet Σ, that is, $\gamma \subseteq \Sigma^* \times \Sigma^*$ and γ contains the pair (v,v) when the word v is a possible input to the channel – this requirement represents the fact that error-free communication over the channel is always possible. The *domain* $\mathrm{dom}\,\gamma$ of the channel γ is the set of all possible inputs of γ, that is,

$$\operatorname{dom}\gamma = \{w \mid (w,z) \in \gamma, \text{ for some word } z\}.$$

Lemma 1 *Let γ_1, γ_2 be channels and L be a language.*

1. *If $\gamma_1 \subseteq \gamma_2$ and L is error-detecting (respectively, error-correcting) for γ_2 then L is error-detecting (respectively error-correcting) for γ_1.*

2. *If L is error-detecting for γ_1 and for γ_2 then L is error-detecting for $\gamma_1 \cup \gamma_2$.*

Proof. For the second statement, let (w_1, w_2) be in $\gamma_1 \cup \gamma_2$, with $w_1, w_2 \in L$. We need to show that $w_1 = w_2$. If (w_1, w_2) is in γ_1 then $w_1 = w_2$, as L is error-detecting for γ_1. The same holds if (w_1, w_2) is in γ_2. Thus, L is error-detecting for $\gamma_1 \cup \gamma_2$. The proof of the first statement is left to the reader. ■

The next result establishes some interesting connections between the concepts of error-detection and error-correction. The composition $\gamma_2 \circ \gamma_1$ of two relations is the relation that consists of all pairs (w_1, w_2) such that (w_1, z) is in γ_1 and (z, w_2) is in γ_2, for some word z. Because of this connection, we focus in the sequel on the error-detection version of Problem 1.

Proposition 1 *Let γ be a channel and let L be a language.*

1. *The language L is error-correcting for L if and only if it is error-detecting for $\gamma^{-1} \circ \gamma$.*

2. *Suppose that all words of L are possible inputs of γ, that is, $L \subseteq \operatorname{dom}\gamma$. Then L is error-detecting for γ if and only if L is error-correcting for $\gamma \uparrow L$.*

Proof. The proof of the first statement is based on the definition of the operation '$\circ$', and is left to the reader. We proceed with the proof of the second statement.

Firstly, let $(w_1, z), (w_2, z)$ be in $\gamma \uparrow L$ with $w_1, w_2 \in L$. We need to show that $w_1 = w_2$. As z is in L and L is error-detecting for γ, it follows that $w_1 = z$ and $w_2 = z$, hence $w_1 = w_2$.

Now consider any pair (w_1, w_2) in γ with w_1, w_2 in L. Then $(w_1, w_2) \in \gamma \uparrow L$. As $\operatorname{dom}\gamma$ contains L and γ is domain preserving, $(w_2, w_2) \in \gamma \uparrow L$. As L is error-correcting for $\gamma \uparrow L$, $w_1 = w_2$.

■

Consider the error model $\mathbb{C}[\text{rat}]$ of all *rational channels*, that is, all channels γ that can be realized by finite transducers. Recall that a *finite transducer* is a finite state machine, with an initial state and some final states, having a finite set of transitions of the form $(p, x/y, q)$. Such a transition says that if the transducer is in state p and the current input starts with x then it enters in state q and outputs the word y. The transducer consumes a given input word and produces some output word by following its available transitions. Our focus on rational channels should not be considered as a restriction because, to our knowledge, most channels can be described by finite-state machines.

Next we obtain a few basic results that confirm our intuition about the legitimacy of the concepts of maximal error-detection and -correction.

Proposition 2 *Let L be a language and let γ be a a channel such that all words of L are possible inputs of γ, that is, $L \subseteq \operatorname{dom}\gamma$. If γ is a $\mathbb{C}$[rat]-maximal error-detecting capability of L then L is maximal error-detecting for γ.*

Proof. Assume for the sake of contradiction that L is not maximal error-detecting for γ. Then, there is a word w not in L such that the language $L' = L \cup \{w\}$ is error-detecting for γ. We choose any word v_0 from L and define the channel

$$\gamma' = \gamma \cup \{(v_0, w)\}.$$

Note that $v_0 \in \operatorname{dom}\gamma$ implies that indeed γ' is domain preserving. Moreover, the channel γ' is in $\mathbb{C}[\mathrm{rat}]$, as the class of rational relations is closed under union. Obviously L' is not error-detecting for γ'. As L' is error-detecting for γ, we have that γ is a proper subset of γ' and, as γ is a maximal error-detecting capability of L, it follows that L is not error-detecting for γ'. Thus, there are two different words v_1, v_2 in L such that (v_1, v_2) is in γ'. As $v_2 \neq w$, it must be the case that (v_1, v_2) is in γ, which contradicts the fact that L is error-detecting for γ. ■

In the next result, for a given language L, the symbol D_L denotes the diagonal relation $\{(w, w) \mid w \in L\}$.

Proposition 3 *For every regular language L there is exactly one $\mathbb{C}$[rat]-maximal error-detecting capability, denoted by μ_L, which is equal to*

$$\mu_L = D_L \cup [L \times (\Sigma^* - L)] \cup [(\Sigma^* - L) \times \Sigma^*]$$

Proof. Note that μ_L is a $\mathbb{C}[rat]$-maximal error-detecting capability of L. Suppose that γ is another one such that $\mu_L \neq \gamma$. If either $\mu_L \subseteq \gamma$ or $\gamma \subseteq \mu_L$ holds, then there is a contradiction. Hence, both γ and μ_L are proper subsets of $\mu_L \cup \gamma$ which implies that L is not error-detecting for $\mu_L \cup \gamma$. On the other hand, L must be error-detecting for $\mu_L \cup \gamma$ by Lemma 1-(2). ■

When the language L is given as a deterministic finite automaton A, say, one can compute a transducer realizing μ_L in time linear with respect to the size of A – this quantity is simply the number of states in A. Indeed, given A, one can construct in linear time an automaton for $\Sigma^* - L$, and transducers for each of the relations $D_L, L \times (\Sigma^* - L), (\Sigma^* - L) \times \Sigma^*$. Hence, we have the following result.

Corollary 1 *Let A be a deterministic finite automaton. The $\mathbb{C}$[rat]-maximum error-detecting capability of $L(A)$ can be computed in linear time.*

It turns out that the analogue of Proposition 2 for the case of error-correction holds true as well, albeit with a little extra work in proving its correctness.

Proposition 4 *Let L be a language and let γ be a a channel such that all words of L are possible inputs of γ, that is, $L \subseteq \operatorname{dom}\gamma$. If γ is a $\mathbb{C}$[rat]-maximal error-correcting capability of L then L is maximal error-correcting for γ.*

Proof. Assume for the sake of contradiction that L is not maximal error-correcting for γ. Then, there is a word w not in L such that the language $L' = L \cup \{w\}$ is error-correcting for γ. We choose a word v_0 from L as follows:

- If there is a word v in L such that (v, w) is in γ then v_0 is any such v; hence, (v_0, w) is in γ.
- If there is no word v in L such that (v, w) is in γ then v_0 is any word in L.

Note that (v_0, v_0) is in γ, as $L \subseteq \operatorname{dom} \gamma$. Define the channel

$$\gamma' = \gamma \cup \{(w, w), (v_0, w)\}.$$

The channel γ' is in $\mathbb{C}[\text{rat}]$, as the class of rational relations is closed under union. Obviously L' is not error-correcting for γ'. As L' is error-correcting for γ, we have that γ is a proper subset of γ' and, as γ is a maximal error-correcting capability of L, it follows that L is not error-correcting for γ'. Thus, there are two different words v_1, v_2 in L such that (v_1, z) and (v_2, z) are in γ', for some word z. We obtain a contradiction as follows.

Case 1: At least one of (v_1, z) and (v_2, z) is not in γ. Without loss of generality, suppose that (v_1, z) is not in γ. Then, (v_1, z) must be in $\{(w, w), (v_0, w)\}$. As w is not in L and v_1 is in L, it must be $(v_1, z) = (v_0, w)$. Also, as $v_0 \neq v_2$, the pair (v_2, w) must be in γ. Then, by the choice of v_0, it follows that (v_0, w) must be in γ, which is a contradiction.

Case 2: Both of (v_1, z) and (v_2, z) are in γ. This implies that L is not error-correcting for γ, which is a contradiction. ■

Unlike the case of error-detection, a language L can have more than one maximal rational error-correcting capability. To see this, consider the finite language $L = \{00001, 1001\}$ and the channels $\sigma(1, \infty)$ – see Example 1 – and $\delta(1, \infty)$ that consists of all pairs (w, z) such that z results by deleting at most one symbol from w. Firstly, note that there is no word z such that both $(00001, z)$ and $(1001, z)$ are in $\delta(1, \infty)$, hence, L is error-correcting for $\delta(1, \infty)$. Similarly, L is error-correcting for $\sigma(1, \infty)$. If there were a unique maximal rational error-correcting capability of L, say γ, then γ would include both $\delta(1, \infty)$ and $\sigma(1, \infty)$. Then, however, a contradiction arises when we note that both $(00001, 0001)$ and $(1001, 0001)$ would be in γ.

3 SID Channels and SID Error Models

The class $\mathbb{C}[\text{rat}]$ of rational channels is interesting from a theoretical point of view but includes channels that do not correspond to physical ones. We turn our attention to SID channels [3] of the form $\tau(m, l)$, where τ is an error type in

$$\{\sigma, \iota, \delta, \sigma \odot \iota, \sigma \odot \delta, \iota \odot \delta, \sigma \odot \iota \odot \delta\}$$

and (m, l) is a pair of nonnegative integers with $m < l$. This channel consists of all pairs of words (w, z) such that z results by using at most m errors of type τ

in any segment of length l (or less) of the input word w. The symbol $\odot$ is used simply as a connective for the simpler types σ, ι, δ, which denote substitutions, insertions, and deletions, respectively. We call the ratio m/l the *error density* of the channel $\tau(m, l)$.

Example 3 The pair (w, z) is in the channel $\sigma \odot \delta(2, 7)$ if and only if z can be obtained by substituting and/or deleting no more than 2 symbols in every segment of length 7 of w. For example the pair

$$(100000000, 01000010)$$

is in $\sigma \odot \delta(2, 7)$, but the pair

$$(100000000, 01001000)$$

is not in $\sigma \odot \delta(2, 7)$ as one has to use more than 2 errors in the prefix 1000000 of 100000000 in order to obtain 01001000.

3.1 The error model $\mathbb{C}^1_\tau[l] = \{\tau(m, l) \mid 0 \leq m < l\}$

In this error model, the parameters τ and l are fixed and, therefore, there are only finitely many channels:

$$\tau(0, l), \ldots, \tau(l - 1, l).$$

In this case, Problem 1 can be solved using the following results of [3]:

- There is an algorithm that constructs a transducer realizing the SID channel $\tau(m, l)$, for any given parameters τ, m and l.
- One can decide in polynomial time, for given rational channel and given regular language, whether the language is error-detecting for the channel.

We note that the construction of a transducer realizing $\tau(m, l)$ would normally require a very large number of states and transitions – see [1] for a relevant discussion.

3.2 $\mathbb{C}_\tau[\mathbf{den} \geq d] = \{\tau(m, l) \mid m < l$ and $m/l \geq d\}$

This error model consists of all channels $\tau(m, l)$ whose error density is at least d, for some fixed parameters τ and d with $0 \leq d < 1$. Note that if m is fixed as well then there is an upper bound on the parameter l of $\tau(m, l)$: $l \leq \lfloor m/d \rfloor$. Now consider a regular language L. We want to *compute* a $\mathbb{C}_\tau[\text{den} \geq d]$-maximal error-detecting (resp. error-correcting) capability of L, where τ and d are fixed. It appears that there are infinitely many channels $\tau(m, l)$ with $m/l \geq d$ that need to be considered. We demonstrate, however, that there is an upper bound on m which implies (using $l \leq \lfloor m/d \rfloor$) that only a finite number of channels $\tau(m, l)$ need to be considered.

1. If the error type τ contains at least one of ι and δ (namely, $\tau \neq \sigma$) then $m < s_L$, where s_L is the shortest length of a word in L. Indeed, if $m \geq s_L$ and $|w| = s_L$ for some $w \in L$, then the pair (w, λ) is in $\tau(m, l)$ if τ contains δ, or the pair (λ, w) is in $\tau(m, l)$ if τ contains ι, for every $l > m$ – recall that λ is the empty word. Hence, L cannot be error-correcting (or error-detecting) for $\tau(m, l)$.

2. If $\tau = \sigma$ the channel only permits substitution errors. If every two different words of L have different lengths then L is error-correcting for $\sigma(m, l)$ for all values of m and l. On the other hand, if there are different words of L that are of the same length, let s'_L be the *shortest* length of two such words. It follows that $m < s'_L$.

Now let M be the maximum value of m according to the above cases. For each value $m = 0, \ldots, M$, we can consider the possible values $l = m + 1, \ldots, \lfloor m/d \rfloor$ for which L is error-detecting for $\tau(m, l)$ by using the results of [3] that were mentioned in Section 3.1. This process would identify a maximal channel $\tau(m, l)$ for a particular m. Now suppose that there are two values m_1, m_2 with $m_1 < m_2$, and two values l_1, l_2 such that both, $\tau(m_1, l_1)$ and $\tau(m_2, l_2)$ are error-detecting capabilities of L. We consider two cases.

- Case of $l_1 \leq l_2$. Here $\tau(m_1, l_1)$ is a proper subset of $\tau(m_2, l_2)$. Hence, $\tau(m_1, l_1)$ can be disregarded and $\tau(m_2, l_2)$ can be kept as a potential maximal error detecting capability of L.

- Case of $l_1 > l_2$. Here the two channels might be incomparable in terms of '$\subseteq$'. In the sequel we consider only the case of $\tau = \sigma$. First consider the word $z = 1^{m_1+1}0^{l_2-(m_1+1)}$ – a symbol of the form a^n denotes the word consisting of n concatenated copies of a. Note that
$$(0^{l_2}, z) \in \sigma(m_2, l_2) \text{ and } (0^{l_2}, z) \notin \sigma(m_1, l_1),$$
as there are more than m_1 errors in the prefix 0^{l_1} of 0^{l_2}. Let $r_1 = \min\{m_1, l_2\%l_1\}$, where $l_2\%l_1$ is the remainder of the division l_2/l_1. If $m_2 < \lfloor l_2/l_1 \rfloor m_1 + r_1$ then one can verify that
$$(0^{l_2}, (1^{m_1}0^{l_1-m_1})^{\lfloor l_2/l_1 \rfloor}1^{r_1}0^{l_2\%l_1-r_1}) \in \sigma(m_1, l_1) - \sigma(m_2, l_2).$$
Hence, the two channels are incomparable! If $m_2 \geq \lfloor l_2/l_1 \rfloor m_1 + r_1$ then $\sigma(m_1, l_1)$ is a subset of $\sigma(m_2, l_2)$ and, therefore, the channel $\sigma(m_1, l_1)$ can be discarded and the channel $\sigma(m_2, l_2)$ can be kept as a potential maximal error-detecting capability of L. To see that indeed $\sigma(m_1, l_1)$ is a subset of $\sigma(m_2, l_2)$, assume for the sake of contradiction that there is a pair (u, v) in $\sigma(m_1, l_1) - \sigma(m_2, l_2)$. Then there is a segment u_2 of u of length l_2 with more than m_2 errors. Let v_2 be the corresponding segment of v. Then (u_2, v_2) is in $\sigma(m_1, l_1) - \sigma(m_2, l_2)$ and, as u_2 is of length l_2, $H(u_2, v_2) > m_2$. Hence, $H(u_2, v_2) > \lfloor l_2/l_1 \rfloor m_1 + r_1$. On the other hand, we can have at most m_1 errors in every segment of u_2 of length l_1, which implies that there can be at most $\lfloor l_2/l_1 \rfloor m_1 + r_1$ errors in u_2; a contradiction.

3.3 The error model $\mathbb{C}_\tau^2[m] = \{\tau(m,l) \mid l > m\}$

In this error model, the parameters τ and m are fixed. Given an automaton A we want to find the smallest l (if any) such that $L(A)$ is error-detecting for $\tau(m,l)$. This turns out to be difficult. Possibly the problem might be easier when $\tau = \sigma$.

4 Discussion

We have introduced the concepts of maximal error-detecting and -correcting capability of a given language, with respect to a certain error model, and have argued that these concepts are meaningful at least from a theoretical point of view. Possible directions for future research include the following.

- Investigate to what extend the algorithmic methods for computing various maximal error-handling capabilities outlined here can be improved in terms of efficiency.
- For each error type τ, resolve the containment relations between any two channels $\tau(m_1, l_1)$ and $\tau(m_2, l_2)$.
- Apply the algorithms to real world languages such as gene languages and codes for data communications.

References

[1] L. Kari, S. Konstantinidis. Descriptional Complexity of Error/Edit Systems, *Journal of Automata, Languages and Combinatorics* 9 (2004) 2/3, 293309.

[2] L. Kari, S. Konstantinidis, S. Perron, G. Wozniak, J. Xu. Computing the Hamming Distance of a Regular Language in Quadratic Time, *WSEAS Transactions on Information Science & Applications* 1 (2004), pp 445-449. Also in: *Proceedings of "8th WSEAS International Conference on Computers, Vouliagmeni, Greece, July 12-15, 2004."*

[3] S. Konstantinidis. Transducers and the Properties of Error-Detection, Error-Correction and Finite-Delay Decodability, *Journal of Universal Computer Science* 8 (2002), 278-291.

[4] S. Konstantinidis. Computing the Levenshtein distance of a regular language. In: *Proceedings of "IEEE Information Theory Workshop on Coding and Complexity, Rotorua, New Zealand, Aug. 29 - Sep. 1, 2005," pp 113-116.*

[5] S. Roman. *Coding and Information Theory.* Springer-Verlag, New York, 1992.

[6] G. Rozenberg, A. Salomaa (eds). *Handbook of Formal Languages, Vol. I.* Springer-Verlag, Berlin, 1997.

[7] S. Yu. Regular Languages. In [6], 41–110.