Error-Detecting Properties of Languages[1]

Stavros Konstantinidis

Department of Mathematics and Computing Science
Saint Mary's University
Halifax, Nova Scotia
B3H 3C3, Canada

s.konstantinidis@stmarys.ca

Abstract: In the context of storing/transmitting words of a language L using a noisy medium, the language property of error-detection is fundamental. It ensures that the medium cannot transform a word from L to another word of L. This paper defines some basic error-detecting properties of languages and obtains a few basic results on error-detection. For example, it is shown that the number of synchronization errors that a regular language can detect is bounded by the size of its syntactic monoid. Moreover, some error-detecting capabilities of uniform, solid, and shuffle codes are considered. It is shown that those codes provide certain error-detection either for free or when a simpler condition is satisfied.

Key words: error-detection, channel, code, regular language, solid code, shuffle code.

1. Introduction

Consider the problem of transmitting/storing words of a language L using a medium γ capable of introducing errors in the words of L. Let us call the words of L *permissible* words and the medium γ *channel*. Now it is possible that a permissible word can be transformed to a non-permissible one after it is received/retrieved from the channel γ. In this context, the language property of error-detection is fundamental. Specifically, if the language L is error-detecting *for* the channel γ, then γ cannot transform a permissible word to another permissible word. As a consequence, when the channel returns a word w which is permissible, it is the case that w is the permissible word that was originally transmitted/stored into γ. On the other hand, if the returned word is not permissible, one can be sure that it has been corrupted by the channel and then take appropriate action – for example, request that the word be retransmitted.

The set of permissible words could be any subset of X^*, where X is the alphabet used, or it could be the set K^* that consists of all the messages (words) over a code K. In the latter case, when a permissible message is returned, it can be decoded uniquely and correctly. To keep the basic definitions general, we use the framework of P-channels (see [3]) restricted to the case of finite words. This channel model is very general and includes the case of SID-channels which were presented in [4] and further extended in [6] – see also [7] for a concise description of the SID-channel model and the tools it provides for studying

[1] This work was supported by a research grant of the Natural Science and Engineering Research Council of Canada.

the notion of error-correction. SID-channels are discrete channels represented by formal expressions that describe the type of errors permitted and the frequency of those errors. The basic error types are:

- σ: *substitution.* It means that a symbol in a message can be replaced with another symbol (of the alphabet X).
- ι: *insertion.* It means that a symbol (of the alphabet X) can be inserted in a message.
- δ: *deletion.* It means that a symbol in a message can be deleted, i.e., replaced with the empty word.

We note that errors of type ι or δ are called *synchronization errors,* as they cause, or are caused by, loss of synchronization. Examples of SID-channel expressions are:

(1) $\sigma(m, \ell)$: represents the channel that permits at most m substitutions in any ℓ (or less) consecutive symbols of a message.

(2) $\iota(m, \ell)$: represents the channel that permits at most m insertions in any ℓ (or less) consecutive symbols of a message.

(3) $\delta(m, \ell)$: represents the channel that permits at most m deletions in any ℓ (or less) consecutive symbols of a message.

(4) $\iota \odot \delta(m, \ell)$: represents the channel that permits a total of at most m insertions and deletions in any ℓ (or less) consecutive symbols of a message.

(5) $\sigma \odot \iota \odot \delta(m, \ell)$: represents the channel that permits a total of at most m substitutions, insertions, and deletions in any ℓ (or less) consecutive symbols of a message.

More generally, we use the expression $\tau(m, \ell)$ to denote the channel that permits a total of at most m errors of type τ in any ℓ consecutive symbols of a message. In this case, we assume that m and ℓ are positive integers with $m < \ell$. In this paper we ignore the distinction between the terms SID-channel and SID-channel expression. Moreover, we consider the following set of error types:

$$\mathcal{T}_1 = \{\sigma, \iota, \delta, \sigma \odot \delta, \sigma \odot \iota, \iota \odot \delta, \sigma \odot \iota \odot \delta\}.$$

The paper is organized as follows. The next section gives some basic concepts about words, factorizations, and P-channels. Section 3 defines the basic error-detecting properties of languages, provides examples to illustrate these properties, and contains a few basic results on error-detection. For example, it is shown that the number of synchronization errors that a regular language can detect is bounded by the cardinality of its syntactic monoid. Section 4 discusses certain error-detecting capabilities of uniform, solid and shuffle codes. In particular, a necessary and sufficient condition is obtained for detecting the errors of the channel $\sigma \odot \iota \odot \delta(1, \ell)$ in the messages of a finite solid code. Finally, Section 5 contains a few concluding remarks.

2. Basic Background

For a set S, the notation $|S|$ represents the cardinality of S. The set of positive integers is denoted by $\mathbb{N}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. An *index set* is a subset I of $\mathbb{N}_0$ such that $I = \{0, 1, \ldots, n-1\}$ for some n in $\mathbb{N}_0$. If $n = 0$, the corresponding index set is the empty set $\emptyset$. An *alphabet*, X, is a finite non-empty set of symbols. A *word (over X)* is a mapping $w : I \to X$, where I is an index set. In this case, we write I_w to denote the index set of the word w. Moreover, as usual, we can denote w by juxtaposing its elements: $w = w(0)w(1)\cdots w(n-1)$. The *empty word*, λ, is the unique word with $I_\lambda = \emptyset$. The *length*, $|w|$, of a word w is the number $|I_w|$. The set of all words over X is denoted by X^* and $X^+ = X^* \setminus \{\lambda\}$. A language is a subset of X^*. We write $\operatorname{minlen} L$ to denote the length of a shortest word in the language L. On the other hand, if L is finite we write $\operatorname{maxlen} L$ to denote the length of a longest word in L. If all the words in L are of the same length, we say that L is a *uniform code.* In this case, we use the symbol $\operatorname{len} L$ to denote the length of the words in L. In the sequel, we fix an alphabet X that contains at least the two distinct symbols 0 and 1.

Let L be a subset of X^*, then a *factorization over* L is a mapping $\varphi : I \to L$ where I is an index set. As before, we write I_φ to indicate the index set of φ, and $|\varphi|$ to denote the length of the factorization φ which is equal to $|I_\varphi|$. For a factorization φ over L, we write $[\varphi]$ to denote the word $\varphi(0)\varphi(1)\cdots\varphi(n-1)$, where $n = |\varphi|$. If $|\varphi| = 0$ then $[\varphi] = \lambda$. For $n \in \mathbb{N}_0$ and $w \in X^*$, the symbol w^n denotes the word $[\varphi]$ such that $|\varphi| = n$ and $\varphi(i) = w$ for all $i \in I_\varphi$. Also, for $W \subseteq X^*$, $W^n = \{w^n \mid w \in W\}$ and $W^{\leq n} = \cup_{i=0}^{n} W^i$.

A *code (over X)* is a non-empty subset K of X^+ such that $[\varphi] = [\psi]$ implies $\varphi = \psi$ for all factorizations φ and ψ over K. A *message over* K is a word $[\varphi]$, where φ is a factorization over K. Then, K^* is the set of all messages over K and K^+ is the set of all non-empty messages.

A *channel*, γ, is a binary relation over X^*, namely $\gamma \subseteq X^* \times X^*$. For the elements of a channel γ, we prefer to write $(y'|y)$ rather than (y', y). Then, $(y'|y) \in \gamma$ means that the word y' can be received from y through the channel γ. For a word y we define $\langle y\rangle_\gamma$ to be the set of all possible outputs of γ when y is used as input; that is,

$$\langle y\rangle_\gamma = \bigl\{y' \in X^* \bigm| (y'|y) \in \gamma\bigr\}.$$

More generally, for a set of words Y, we have $\langle Y\rangle_\gamma = \bigcup_{y\in Y} \langle y\rangle_\gamma$.

Definition 1 Let γ be a channel and let υ be a factorization over $Y \subseteq X^*$. A factorization υ' over $\langle Y\rangle_\gamma$ is *γ-admissible for υ* if

$$I_{\upsilon'} = I_\upsilon \quad \text{and} \quad \upsilon'(i)\cdots\upsilon'(i+k) \in \langle \upsilon(i)\cdots\upsilon(i+k)\rangle_\gamma,$$

for all $i \in I_\upsilon$ and $k \in \mathbb{N}_0$ with $i + k \in I_\upsilon$.

Example 1 Consider the message $y = 001100$ and its factorization υ over $K = \{00, 11\}$ such that $\upsilon = (00, 11, 00)$. Consider also a channel γ that allows at most one deletion in any 2 consecutive input symbols. As a result, $y' = 0100$ is a possible output in $\langle y\rangle_\gamma$ if one deletes the symbols $y(0)$ and $y(2)$ in y. Then the factorization υ' of y' over $\langle K\rangle_\gamma$ such that $\upsilon' = (0, 1, 00)$ is γ-admissible for υ. On the other hand, for the same channel γ, and for $K = \{01, 10\}$ and $y = 0110$, one has the following: $\upsilon = (01, 10)$ is a factorization of y over K and $\upsilon' = (0, 0)$ is a factorization of $y' = 00$ over $\langle K\rangle_\gamma$ such that $\upsilon'(i) \in \langle \upsilon(i)\rangle_\gamma$ for $i \in \{0, 1\}$. But $y' \notin \langle \upsilon(0)\upsilon(1)\rangle_\gamma$ since the symbols $y(1)$ and $y(2)$ of y cannot be both deleted. Hence, υ' is not γ-admissible for υ.

In the sequel, we consider only channels γ satisfying the following natural conditions.

($\mathcal{P}_1$) *Input factorizations arrive as γ-admissible output factorizations:* If $(y'|y) \in \gamma$ and υ is a non-empty factorization of y over some subset Y of X^+, then there is a factorization υ' of y' over $\langle Y \rangle_\gamma$ which is γ-admissible for υ.

($\mathcal{P}_2$) *Error-free messages can be received independently of the context:* If $(y'|y) \in \gamma$ then $(xy'z|xyz) \in \gamma$, for all $x, z \in X^*$.

($\mathcal{P}_3$) *Empty input can result into empty output:* $(\lambda|\lambda) \in \gamma$.

Channels satisfying properties $\mathcal{P}_1$–$\mathcal{P}_3$ are called *P_*-channels.* They differ from the P-channels defined in [3] only in the finiteness type of the inputs and outputs; that is, P_*-channels allow only finite words to be used as opposed to P-channels. Consequently, property $\mathcal{P}_0$ of P-channels is omitted here. We note that properties $\mathcal{P}_2$ and $\mathcal{P}_3$ imply $(y|y) \in \gamma$ for all $y \in X^*$. Moreover, every SID-channel is a P_*-channel.

We close this section with an example of how words can be affected by the errors of an SID-channel.

Example 2 Consider the word $x = 0000000$ and the SID-channel $\gamma = \iota \odot \delta(2,5)$ that permits at most 2 insertions and deletions in any 5 consecutive symbols. Let $y = 01000001$ and let $z = 0110000010$. Observe that y can be obtained from x when γ deletes $x(2)$, inserts a 1 between $x(0)$ and $x(1)$, and inserts a 1 at the end of x – all the errors occur at the same time. Hence, $y \in \langle x \rangle_\gamma$. On the other hand, to obtain z from x using a minimum number of errors, one has to insert three 1s in the segment $x(1) \cdots x(5)$ of length 5. Hence, $z \notin \langle x \rangle_\gamma$.

3. Error Detection: Definitions, Examples and Basic Results

The classical theory of error-correcting codes deals with channels that permit substitution errors and considers primarily uniform codes. In that context, a uniform code K is said to be m-error-detecting if $v_1 \in \langle v_2 \rangle_\gamma$ implies $v_1 = v_2$, for all codewords v_1 and v_2, where $\gamma = \sigma(m, \ell)$ and ℓ is the length of the words in K – see [1] or [9]. The notion of error-detection has been generalized in [3] to the case of P-channels, but no results are included there concerning error-detection. In this section we investigate the notion of $(\gamma, *)$-detecting code as defined in [3]. In many cases, this property can be studied in terms of the simpler notion of (γ, t)-detecting code, where $t \in \mathbb{N}_0$. The formal definitions are provided next.

Definition 2 Let γ be a P_*-channel and let $t \in \mathbb{N}_0$.

(i) A language L is *error-detecting for γ*, if

$$\forall\, w_1, w_2 \in L \cup \{\lambda\},\ \ w_1 \in \langle w_2 \rangle_\gamma \ \longrightarrow\ w_1 = w_2.$$

The symbol ED_γ denotes the class of languages which are error-detecting for γ.

(ii) A code K is *$(\gamma, *)$-detecting,* if the language K^* is error-detecting for γ. The symbol ED^*_γ denotes the class of codes which are $(\gamma, *)$-detecting.

(iii) A code K is (γ,t)-*detecting*, if

$$\forall\ w_1 \in K^{\leq t}\ \forall\ w_2 \in K^*, \quad w_1 \in \langle w_2\rangle_\gamma \longrightarrow w_1 = w_2.$$

The symbol ED_γ^t denotes the class of codes which are (γ,t)-detecting.

In part (i) of Definition 2, the use of "$w_1, w_2 \in L \cup \{\lambda\}$" as opposed to "$w_1, w_2 \in L$" is justified as follows. First, it should not be possible for the channel γ to return a non-empty word in L when nothing is sent to γ, i.e., when the input used is λ. That is, $w_1 \in \langle\lambda\rangle_\gamma$ *and* $w_1 \in L \cup \{\lambda\}$ implies $w_1 = \lambda$. Similarly, the channel should not be capable of erasing completely a non-empty word of L. That is, $\lambda \in \langle w_2\rangle_\gamma$ *and* $w_2 \in L \cup \{\lambda\}$ implies $w_2 = \lambda$. These observations do not eliminate from consideration channels that insert or delete symbols. Instead, they ensure that when an error-detecting language is used for γ, it is impossible that γ can erase or introduce an entire non-empty word of L.

Next we show a few examples of error-detecting codes. We also remark that every $(\gamma,*)$-correcting code is $(\gamma,*)$-detecting as well.[2]

Example 3 Every uniform code K is error-detecting for the channel $\gamma = \iota(m,\ell)$, provided $\operatorname{len} K > m$. Indeed, as only insertions are permitted, $x \in \langle v\rangle_\gamma$ implies $|v| \leq |x|$; therefore, $\lambda \in \langle v\rangle_\gamma$ and $v \in K \cup \{\lambda\}$ imply $v = \lambda$. On the other hand, as $v \in \langle\lambda\rangle_\gamma$ implies $|v| \leq m$, one has that $v \in \langle\lambda\rangle_\gamma$ and $v \in K \cup \{\lambda\}$ imply $v = \lambda$. Now let v_1 and v_2 be codewords of K such that $v_1 \in \langle v_2\rangle_\gamma$. As only insertions are permitted, one has that $|v_1| \geq |v_2|$. In particular, $|v_1| = |v_2|$ if and only if no insertion occurs in v_2, if and only if $v_1 = v_2$. Hence, as K is uniform, $v_1 = v_2$. Analogously, one can verify that every uniform code K is error detecting for $\delta(m,\ell)$, provided $\operatorname{len} K > m$.

Example 4 One can verify that the code $K_0 = \{000, 111\}$ is error-detecting for the channel $\gamma = \sigma\odot\iota\odot\delta(1,3)$. But K_0 is not $(\gamma,*)$-detecting. Indeed, consider the messages $w_2 = (000)^3$ and $w_1 = (000)^2$ such that $w_1 \neq w_2$. Then, $w_1 \in \langle w_2\rangle_\gamma$ by deleting appropriately three symbols from w_2.

Example 5 Consider the code $K_1 = \{v_1, v_2 \mid v_1 = 00111,\ v_2 = 0101011\}$ and the channel $\gamma = \delta(1,7)$. From the equalities $\langle v_1\rangle_\gamma = \{v_1, 0111, 0011\}$ and

$$\langle v_2\rangle_\gamma = \{v_2, 101011, 001011, 011011, 010011, 010111, 010101\},$$

one verifies that K_1 is error-detecting for γ. In addition, we claim that K_1 is $(\gamma,*)$-detecting. Indeed, note first that $\lambda \notin \langle w\rangle_\gamma$ and $w \notin \langle\lambda\rangle_\gamma$ for all $w \in K_1^+$. Now consider two messages w_1 and w_2 in K_1^+ such that $w_1 \in \langle w_2\rangle_\gamma$. Then, $w_1 = [\kappa_1]$ and $w_2 = [\kappa_2]$ for some factorizations κ_1 and κ_2 over K_1. By property $\mathcal{P}_1$ of the channel γ, there is a factorization ψ which is γ-admissible for κ_2 such that $[\psi] = w_1 = [\kappa_1]$ and $\psi(i) \in \langle\kappa_2(i)\rangle_\gamma$ for all $i \in I_\psi = I_{\kappa_2}$. It is sufficient to show that $\psi = \kappa_1$; then, as K_1 is error-detecting for γ, $\kappa_1(i) \in \langle\kappa_2(i)\rangle_\gamma$ implies $\kappa_1(i) = \kappa_2(i)$ for all i in I_{κ_2}. So consider the word $\kappa_1(0)$ of K_1 which is a prefix of both, $[\kappa_1]$ and $[\psi]$. If $\kappa_1(0) = v_1$ then $\psi(0) = v_1$ or $\psi(0) = 0011$. The

[2] A code K is $(\gamma,*)$-correcting if $\langle w_1\rangle_\gamma \cap \langle w_2\rangle_\gamma \neq \emptyset$ implies $w_1 = w_2$, for all w_1 and w_2 in K^*.

second case implies $\psi(1) = 101011$ which is impossible, as two deletions would occur in $\kappa_2(0)\kappa_2(1)$ within a segment of length less than 7. Hence, $\psi(0) = v_1$ as well. Similarly, one verifies that if $\kappa_1(0) = v_2$ then $\psi(0) = v_2$ as well. Hence, $\psi(0) = \kappa_1(0)$ and $\psi(1)\psi(2)\cdots = \kappa_1(1)\kappa_1(2)\cdots$. The same argument can be applied repeatedly to obtain $\psi(i) = \kappa_1(i)$ for all i in I_ψ.

The following proposition describes certain relationships between the error-detecting properties given in Definition 2.

Proposition 1 *For every t in $\mathbb{N}_0$ and for every P_*-channel γ, the following relationships are valid.*
(i) $\mathrm{ED}_\gamma^{t+1} \subseteq \mathrm{ED}_\gamma^t$.
(ii) $\mathrm{ED}_\gamma^1 \subseteq \mathrm{ED}_\gamma$.
(iii) $\mathrm{ED}_\gamma^* = \cap_{t=0}^\infty \mathrm{ED}_\gamma^t$.

Proof: Consider a code K which is $(\gamma, t+1)$-detecting and the messages $w_1 \in K^{\leq t}$ and $w_2 \in K^*$ such that $w_1 \in \langle w_2 \rangle_\gamma$. Let $v \in K$. By property $\mathcal{P}_2$ of the channel γ, one has $w_1 v \in \langle w_2 v \rangle_\gamma$. As $w_1 v \in K^{\leq t+1}$ and $w_2 v \in K^*$, it follows that $w_1 v = w_2 v$. Hence, $w_1 = w_2$ and the first inclusion is correct. Obviously, the second inclusion is correct as well. For the third relationship, one can easily verify that $\mathrm{ED}_\gamma^* \subseteq \mathrm{ED}_\gamma^t$ for all t in $\mathbb{N}_0$. Hence, $\mathrm{ED}_\gamma^* \subseteq \cap_{t=0}^\infty \mathrm{ED}_\gamma^t$. On the other hand, consider a code K in $\cap_{t=0}^\infty \mathrm{ED}_\gamma^t$ and $w_1, w_2 \in K^*$ with $w_1 \in \langle w_2 \rangle_\gamma$. Then, there is $t \in \mathbb{N}_0$ such that $w_1 \in K^t$ and, as $K \in \mathrm{ED}_\gamma^t$, it follows that $w_1 = w_2$. Hence, $K \in \mathrm{ED}_\gamma^*$. □

Next it is shown that the inclusion in Proposition 1(i) can be proper for every value of the parameter t.

Proposition 2 *For every t in $\mathbb{N}_0$ there is an SID-channel γ such that ED_γ^{t+1} is properly contained in ED_γ^t.*

Proof: For each t in $\mathbb{N}_0$ consider the SID-channel $\gamma = \gamma(t) = \delta(1, t+2)$ and the code $K = K(t) = \{0^{t+2}\}$. First we show that K is (γ, t)-detecting and then that K is not $(\gamma, t+1)$-detecting.

Let $w_1 \in K^m$ and $w_2 \in K^n$ such that $w_1 \in \langle w_2 \rangle_\gamma$, $m \leq t$, and $n \in \mathbb{N}_0$. As only deletions are permitted, $|w_1| \leq |w_2|$. If $|w_1| = |w_2|$ then $w_1 = w_2$ as required. On the other hand, we show that the assumption $|w_1| < |w_2|$ leads to a contradiction. Indeed, as $|K| = 1$, this assumption implies $m + 1 \leq n$. Now as w_2 consists of n codewords each of length $t+2$, at most one symbol can be deleted in each codeword and, therefore, at most n deletions can occur in w_2. Hence, $|w_1| \geq |w_2| - n$ which together with $m + 1 \leq n$ imply

$$m(t+2) \geq n(t+2) - n \Rightarrow n \leq \frac{m(t+2)}{t+1} \Rightarrow m + 1 \leq \frac{m(t+2)}{t+1} \Rightarrow t + 1 \leq m.$$

The last inequality, however, contradicts $m \leq t$.

Now we show that K is not $(\gamma, t+1)$-detecting. Let $w_1 = (0^{t+2})^{t+1} \in K^{\leq t+1}$ and $w_2 = (0^{t+2})^{t+2} \in K^*$. Clearly $w_1 \neq w_2$. On the other hand, one has that $w_1 \in \langle w_2 \rangle_\gamma$ by deleting appropriately one zero in every $t+2$ consecutive symbols of w_2. □

The following result poses a certain restriction on the words of $(\gamma, *)$-detecting codes for SID-channels that involve insertions or deletions.

Proposition 3 *Let K be a code and let $\gamma = \tau(m, \ell)$ be an SID-channel with $\tau \in \mathcal{T}_1 \setminus \{\sigma\}$. If K is $(\gamma, *)$-detecting, then $x^n \notin K$ for all $x \in X^{\leq m}$ and for all $n \in \mathbb{N}$.*

Proof: As $\tau \neq \sigma$, at least one of ι and δ occurs in τ. Assume that δ occurs in τ and that K is $(\gamma, *)$-detecting, but suppose $x^n \in K$ for some $x \in X^{\leq m} \cap X^+$ and $n \in \mathbb{N}$. Let $v = x^n$. Note that both $w_2 = v^{n\ell}$ and $w_1 = v^{n\ell-1}$ are in K^* and that $w_1 \neq w_2$. We show that $w_1 \in \langle w_2 \rangle_\gamma$ which contradicts the fact that K is $(\gamma, *)$-detecting. Let $y = x^{n\ell-1}$ such that $v^\ell = xy$. Then, $w_2 = (v^\ell)^n = (xy)^n = (xy)(xy)\cdots(xy)$. Moreover, as $|xy| = \ell|v| = \ell n|x| \geq \ell$, it is possible that γ deletes the prefix x in each of the n factors xy of w_2. Hence, $y^n \in \langle w_2 \rangle_\gamma$. But $y^n = x^{(n\ell-1)n} = v^{n\ell-1} = w_1$. The case where only ι occurs in τ can be shown analogously. □

The next proposition gives a certain bound on the number of insertion/deletion errors that a regular language can detect. The symbol $\operatorname{syn} L$ denotes the syntactic monoid of the language L which is the factor monoid defined by the syntactic (or principal) congruence of L. It is well-known that a language L is regular if and only if $\operatorname{syn} L$ is finite (see [3] or [11]).

Proposition 4 *Let τ be an error type in $\mathcal{T}_1 \setminus \{\sigma\}$. No regular language L is error-detecting for $\tau(m, \ell)$, when $m \geq |\operatorname{syn} L|$ and $L \notin \{\emptyset, \{\lambda\}\}$.*

Proof: As $\tau \neq \sigma$, at least one of δ and ι occurs in τ. Let $\gamma = \tau(m, \ell)$ and let A be a minimal complete deterministic finite automaton accepting L; that is, the number of states k of the automaton A is minimum. Then, $k = |\operatorname{syn} L|$ (see [12]). As $L \notin \{\emptyset, \{\lambda\}\}$, there is a non-empty word w in L. If $|w| < k$, then $|w| < m$ and the channel can erase or introduce w depending on whether δ occurs in τ. That is, $\lambda \in \langle w \rangle_\gamma$ or $w \in \langle \lambda \rangle_\gamma$. Hence, as $w \neq \lambda$, the language L is not error-detecting for γ. Now assume $|w| \geq k$. By a pumping lemma of the regular languages (see [12]), there are words x, y, z such that $w = xyz$, $1 \leq |y| \leq k$, and $xy^n z \in L$ for all n in $\mathbb{N}_0$. In particular, $xz \in L$. As $|y| \leq m$, one has $xz \in \langle w \rangle_\gamma$ or $w \in \langle xz \rangle_\gamma$ depending on whether δ occurs in τ. Hence, as $w \neq xz$, it follows that L is not error-detecting for γ. □

4. Error-detecting Uniform, Solid, and Shuffle Codes

In this section we consider certain error-detecting capabilities of some known classes of codes. There are cases where, due to the characteristics of the codes used, $(\gamma, 1)$-detection is sufficient to ensure $(\gamma, *)$-detection. On the other hand, for some classes of codes, $(\gamma, 1)$-detection is provided for free. The first result concerns the channel $\sigma(m, \ell)$ that involves only substitution errors. This result justifies the use of uniform codes for such channels.

Proposition 5 *Let K be a uniform code and let γ be the channel $\sigma(m, \ell)$. Then, K is $(\gamma, *)$-detecting if and only if it is $(\gamma, 1)$-detecting.*

Proof: The 'only if' part follows immediately from Proposition 1(ii). Now assume that K

is a uniform code of length $n \in \mathbb{N}$ and that K is $(\gamma, 1)$-detecting. Let w_1, w_2 be messages in K^* such that $w_1 \in \langle w_2 \rangle_\gamma$. Then, there are factorizations κ_1, κ_2 over K such that $[\kappa_1] = w_1$ and $[\kappa_2] = w_2$. Property $\mathcal{P}_1$ implies that there is a factorization ψ which is γ-admissible for κ_2 such that $w_1 = [\psi]$ and $\psi(i) \in \langle \kappa_2(i) \rangle_\gamma$ for all $i \in I_\psi = I_{\kappa_2}$. As γ permits only substitutions, one has $|\psi(i)| = n$ for all $i \in I_{\kappa_2}$. Hence, $|[\psi]| = n|\kappa_2|$. On the other hand, $|w_1| = n|\kappa_1|$; therefore, $|\kappa_1| = |\kappa_2| = |\psi|$ which implies $\psi = \kappa_1$. Now as $\kappa_1(i) \in \langle \kappa_2(i) \rangle_\gamma$ and K is $(\gamma, 1)$-detecting, it follows that $\kappa_1(i) = \kappa_2(i)$ for all $i \in I_{\kappa_1}$. Hence, $w_1 = w_2$. $\square$

A similar statement follows about finite solid codes for the channel $\sigma \odot \iota \odot \delta(1, \ell)$. A language K is a *solid code,* if it is an infix and overlap-free language; that is, $K \cap (X^*KX^+ \cup X^+KX^*) = \emptyset$ and, for all $u, v \in X^+$ and $x \in X^*$, $vx, xu \in K$ implies $x = \lambda$. Some interesting decoding capabilities of solid codes are discussed in [3]. Recent results on solid codes can be found in [2] and [8].

The proof of the following proposition is based on a special property of the assumed type of solid codes. Let K be a code and let γ be a P_*-channel. A factorization ψ is said to be (γ, K)*-corrupted,* if it is γ-admissible for some factorization κ over K and $\kappa \neq \psi$. Thus, $[\psi] \in \langle [\kappa] \rangle_\gamma$ and there is at least one factor $\psi(i)$ of ψ which is not equal to its corresponding factor $\kappa(i) \in K$. The property we need is the following.

$\mathcal{P}(\gamma, K)$: If ψ is a (γ, K)-corrupted factorization then $[\psi] \notin K^*$.

One can verify that every code satisfying $\mathcal{P}(\gamma, K)$ must be a $(\gamma, *)$-detecting code.

Proposition 6 *Let γ be the channel $\sigma \odot \iota \odot \delta(1, \ell)$ and let K be a finite solid code with* $\operatorname{maxlen} K \leq \ell$. *Then, K is $(\gamma, *)$-detecting if and only if it is $(\gamma, 1)$-detecting.*

Proof: The 'only if' part follows immediately from Proposition 1(ii). Now assume that K is $(\gamma, 1)$-detecting. We show that $\mathcal{P}(\gamma, K)$ holds. Let κ be a factorization over K and let ψ be γ-admissible for κ such that $\psi \neq \kappa$. Then, $|\kappa| = |\psi| > 0$. Now suppose that $[\psi] \in K^*$; that is, $[\psi] = [\mu]$ for some factorization μ over K. If $|\mu| = 0$ then $[\mu] = \lambda \in \langle [\kappa] \rangle_\gamma$ which contradicts the fact that K is $(\gamma, 1)$-detecting. Hence, $|\mu| > 0$.

Let $k = |\kappa| = |\psi|$ and $m = |\mu|$. Then, $[\psi] = \psi(0) \cdots \psi(k-1) = \mu(0) \cdots \mu(m-1)$. As $\kappa \neq \psi$, there is a minimum $p \in I_\kappa$ such that $\kappa(p) \neq \psi(p)$. Then, $[\psi] = \kappa(0) \cdots \kappa(p-1)\psi(p) \cdots \psi(k-1)$ and, as K is a prefix code, $\kappa(i) = \mu(i)$ for all $i < p$. Hence, $\psi(p) \cdots \psi(k-1) = \mu(p) \cdots \mu(m-1)$. Now, for all j in $\{p, p+1, \ldots, k-1\}$ one has

$$\psi(j) = \begin{cases} x_j y_j, & \text{if } \kappa(j) = x_j a_j y_j \text{ with } a_j \in X \text{ deleted;} \\ x_j a_j y_j, & \text{if } \kappa(j) = x_j y_j \text{ with } a_j \in X \text{ inserted; or} \\ & \text{if } \kappa(j) = x_j b_j y_j \text{ with } b_j \in X \text{ substituted with } a_j \in X; \\ \kappa(j), & \text{if no error occurs.} \end{cases}$$

Of course, when $j = p$, $\psi(j) \neq \kappa(j)$. For the lengths of $\mu(p)$ and $\psi(p)$ we distinguish three cases which all lead to contradictions due to the fact that K is a $(\gamma, 1)$-detecting solid code.

First, assume $|\mu(p)| > |\psi(p)|$. Then, $\mu(p) = \psi(p) \cdots \psi(r)w$ where $p \leq r$ and w is either equal to $\psi(r+1)$ or to a non-empty proper prefix of $\psi(r+1)$. The former case implies $\mu(p) \in \langle K^2 K^* \rangle_\gamma \cap K$ which is impossible. Hence, $0 < |w| < |\psi(r+1)|$ and $\psi(r+1) = ws$ with $s \in X^+$. The case $\psi(r+1) = \kappa(r+1)$ is not possible, as otherwise w

would be a proper suffix of $\mu(p)$ and a proper prefix of $\kappa(r+1)$. Hence, $\psi(r+1)$ is of the form $x_{r+1}y_{r+1}$ or $x_{r+1}a_{r+1}y_{r+1}$. If $|w| \leq |x_{r+1}|$ the overlap-freeness of K is violated again. Hence, $ws = x_{r+1}y_{r+1}$ or $ws = x_{r+1}a_{r+1}y_{r+1}$, and $|w| > |x_{r+1}|$. It follows then that $\mu(p+1)$ either is contained in y_{r+1} or it starts with a proper suffix of y_{r+1}.

Second, assume $|\mu(p)| < |\psi(p)|$. Then, $\psi(p) = \mu(p)s$ where $s \in X^+$ and $m > p$. As K is an infix code, it must be $|\mu(p)| > |x_p|$ and, therefore, $|s| \leq |y_p|$. Then, however, $\mu(p+1)$ is either contained in y_p or it starts with a suffix of y_p. Finally, the case $|\mu(p)| = |\psi(p)|$ is also impossible, as it violates the fact that K is $(\gamma, 1)$-detecting. $\square$

The code K_1 of Example 5 is a $(\gamma, 1)$-detecting solid code, where $\gamma = \sigma \odot \iota \odot \delta(1, 7)$. Hence, Proposition 6 implies that K_1 is $(\gamma, *)$-detecting as well.

Let's consider now the classes of shuffle codes, as they provide error-detecting capabilities for SID-channels that involve either insertions or deletions. A language K is a *prefix-shuffle code of index* $n \in \mathbb{N}$, if $x_0 \cdots x_{n-1} \in K$ and $x_0 y_0 \cdots x_{n-1} y_{n-1} \in K$ imply $y_0 = \cdots = y_{n-1} = \lambda$, for all words x_i and y_i in X^*. Let PS_n be the class of prefix-shuffle codes of index n. Then, $\mathrm{PS}_{n+1} \subseteq \mathrm{PS}_n$. The class SS_n of *suffix-shuffle codes of index* n is defined analogously: $x_0 \cdots x_{n-1} \in K$ and $y_0 x_0 \cdots y_{n-1} x_{n-1} \in K$ imply $y_0 = \cdots = y_{n-1} = \lambda$. Again, one has $\mathrm{SS}_{n+1} \subseteq \mathrm{SS}_n$. The class IS_n of *infix-shuffle codes of index* n consists of all codes K such that $x_0 \cdots x_{n-1} \in K$ and $y_0 x_0 \cdots y_{n-1} x_{n-1} y_n \in K$ imply $y_0 = \cdots = y_{n-1} = y_n = \lambda$ for all x_i and y_j in X^*. Then, $\mathrm{IS}_{n+1} \subseteq \mathrm{IS}_n$. Finally, for the class OS_n of *outfix-shuffle codes of index* n, one has that $x_0 \cdots x_n \in K$ and $x_0 y_0 \cdots x_{n-1} y_{n-1} x_n \in K$ imply $y_0 = \cdots = y_{n-1} = \lambda$. Again, one has $\mathrm{OS}_{n+1} \subseteq \mathrm{OS}_n$. Moreover, for all $n \in \mathbb{N}$,

$$\mathrm{PS}_{n+1} \cup \mathrm{SS}_{n+1} \subseteq \mathrm{IS}_n \cap \mathrm{OS}_n \quad \text{and} \quad \mathrm{IS}_n \cup \mathrm{OS}_n \subseteq \mathrm{PS}_n \cap \mathrm{SS}_n.$$

We refer the reader to [3] for further results on shuffle codes.

Proposition 7 *Let $m, \ell \in \mathbb{N}$ with $m < \ell$, and let K be a code with* $\mathrm{minlen}\, K > m$ *and* $\mathrm{maxlen}\, K \leq \ell$.

(i) If K is outfix-shuffle of index m then it is error-detecting for $\iota(m, \ell)$ and for $\delta(m, \ell)$.

(ii) If K is prefix-shuffle of index $m+1$ then it is $(\gamma, 1)$-detecting, where $\gamma = \iota(m, \ell)$.

Proof: (i) Let $\gamma = \delta(m, \ell)$. Then, if $z \in \langle x \rangle_\gamma$ and $|x| \leq \ell$, at most m symbols can be deleted from x to obtain z. Observe that, if k is the number of symbols deleted, then x can be written in the form $x_0 a_0 \cdots x_{k-1} a_{k-1} x_k$ and z in the form $x_0 \cdots x_{k-1} x_k$, where $a_0, \ldots, a_{k-1} \in X$ are the deleted symbols and $x_0, \ldots, x_k \in X^*$. From this observation and the fact $\mathrm{OS}_m \subseteq \mathrm{OS}_k$ for $k \leq m$, it follows easily that if K is outfix-shuffle of index m then it is error-detecting for $\delta(m, \ell)$. Using a similar argument, one can show that K is also error-detecting for $\iota(m, \ell)$.

(ii) Let K be prefix-shuffle of index $m+1$ and let $w_1 \in K \cup \{\lambda\}$ and $w_2 \in K^*$ such that $w_1 \in \langle w_2 \rangle_\gamma$. As $\mathrm{minlen}\, K > m$ and γ permits at most m insertions in any ℓ or less consecutive symbols of w_2, it follows that when one of w_1 and w_2 is empty they must both be empty. Now assume $w_1 \in K$ and $w_2 \in K^n$ for some n in $\mathbb{N}$. Then, $w_2 = [\kappa]$ and $w_1 = [\psi]$, where κ is a factorization over K of length n and ψ is γ-admissible for κ. We show that $w_1 = \kappa(0)$. As $\psi(0) \in \langle \kappa(0) \rangle_\gamma$ and $|\kappa(0)| \leq \ell$, at most m insertions can occur in $\kappa(0)$. More specifically, let k be the number of insertions in $\kappa(0)$ and let $a_0, \ldots, a_{k-1} \in X$ be the symbols inserted. Then, $0 \leq k \leq m$ and, $\psi(0) = x_0 a_0 \cdots x_{k-1} a_{k-1} x_k$ and $\kappa(0) = x_0 \cdots x_{k-1} x_k$ for some words $x_0, \ldots, x_{k-1}, x_k$. Now $[\psi] = \psi(0)s$ and $s \in \langle \kappa(1) \cdots \kappa(n-1) \rangle_\gamma$, for some s in X^*, and $w_1 = x_0 a_0 \cdots x_{k-1} a_{k-1} x_k s \in K$. As K is prefix-shuffle of index $m+1$, it is also prefix-shuffle of index $k+1$ and, therefore, $w_1 = \kappa(0)$ which implies $k = 0$ and $s = \lambda$. Moreover, $\kappa(1) \cdots \kappa(n-1) = \lambda$ implies $n = 1$ and $w_2 = \kappa(0)$. Hence, $w_1 = w_2$ as required. □

We note that a code satisfying the premises of Proposition 7 is not necessarily $(\gamma, *)$-detecting. For example, the code K_0 of Example 4 is prefix-shuffle of index 2 and $(\gamma, 1)$-detecting, where $\gamma = \iota(1, 3)$. But K_0 is not $(\gamma, *)$-detecting.

5. Discussion

In this paper, we have argued that error-detection is a fundamental language property when it comes to storing/communicating data. We have presented some initial results on error-detection at the general level of P- and SID-channels, and examined certain error-detecting capabilities of uniform, solid, and shuffle codes. Some potentially interesting questions that arise from this work are the following:

(1) With Proposition 4 in mind, what other bounds exist on the insertion/deletion-detecting capabilities of languages?

(2) Is it possible to show that solid codes possess stronger error-detecting capabilities than the one shown in Proposition 6 for the SID-channel $\sigma \odot \iota \odot \delta(1, \ell)$?

(3) How large is the intersection between certain shuffle codes and solid codes? In view of Proposition 6 and Proposition 7, it appears that codes in that intersection provide certain $*$-error-detecting capabilities for free.

A related concept which is desirable from a practical point of view is the property of error-detection with finite delay. This property allows the detection of errors in a word w by examining consecutive segments of w of bounded length, one at a time. Some initial results on this topic exist in [5].

References

[1] J. Duske, H. Jürgensen: *Codierungstheorie*. BI Wissenschaftsverlag, Mannheim, 1977.

[2] H. Jürgensen, M. Katsura, S. Konstantinidis: Maximal solid codes. Report 533, Department of Computer Science, University of Western Ontario, 1999. (Submitted for publication).

[3] H. Jürgensen, S. Konstantinidis: Codes. In Rozenberg and Salomaa [10], 511–607.

[4] H. Jürgensen, S. Konstantinidis: Error correction for channels with substitutions, insertions, and deletions. In J.-Y. Chouinard, P. Fortier, T. A. Gulliver (editors): *Information Theory and Applications 2, Fourth Canadian Workshop on Information Theory. Lecture Notes in Computer Science* **1133**, 149–163, Berlin, 1996. Springer-Verlag.

[5] S. Konstantinidis: Error-detection with finite delay. In preparation.

[6] S. Konstantinidis: An algebra of discrete channels that involve combinations of three basic error types. Report CS-02-96, University of Lethbridge, 1996. (Submitted for publication).

[7] S. Konstantinidis: Relationships between different error correcting capabilities of a code. In *Proceedings, IEEE Inform. Theory Workshop, 1998, Killarney, Ireland*. 122–123, 1998.

[8] N. H. Lâm: Finite maximal solid codes. Preprint 98/17, Vietnam National Centre for Natural Science and Technology, Institute of Mathematics, 1998.

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

[10] G. Rozenberg, A. Salomaa (editors): *Handbook of Formal Languages*, vol. I. Berlin, 1996. Springer-Verlag.

[11] H. J. Shyr: *Free Monoids and Languages*. Hon Min Book Company, Taichung, second ed., 1991.

[12] S. Yu: Regular languages. In Rozenberg and Salomaa [10], 41–110.