ELSEVIER

Theoretical Computer Science 276 (2002) 355–375

Theoretical Computer Science

www.elsevier.com/locate/tcs

Error-detecting properties of languages [☆]

Stavros Konstantinidis [*], Amber O'Hearn

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

Received June 2000; revised April 2001; accepted May 2001
Communicated by A. Salomaa

Abstract

The language property of error-detection ensures that the communications medium cannot transform a word of the language to another word of the language. In this paper we provide some insights on the notion of error-detection from a language theoretic point of view. We define certain error-detecting properties of languages and codes including the notion of error-detection with finite delay which is a natural extension of unique decodability with finite delay. We obtain results about the error-detecting capabilities of regular and other languages, and of known classes of codes. Moreover, we consider the problem of estimating the optimal redundancy of infinite languages with the property of detecting errors of the deletion type. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Channel; Code; Language; Error-detection; Finite delay; Redundancy

1. Introduction

Communication of information presupposes the existence of a communications medium and a language of communications which consists of all the possible messages (words) that can be communicated. Normally, the medium, say γ, is capable of introducing errors in the words of the language, say L, and there is the possibility that, when a word w of L is communicated, γ returns a word w' which is not in L or it is in L but $w' \neq w$. These considerations are important, for instance, in the transmission of digital information and in the typesetting of ASCII characters. In the former case, the channel (medium) is capable of substituting symbols of the message with

[☆] This work was supported by research Grant OGP220259 of the Natural Sciences and Engineering Research Council of Canada.

[*] Corresponding author.

E-mail addresses: s.konstantinidis@stmarys.ca (S. Konstantinidis), a_ohearn@cs.stmarys.ca (A. O'Hearn).

0304-3975/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.

PII: S0304-3975(01)00280-8

other symbols and possibly inserting new or deleting existing symbols in the message. Moreover, in this case, the language of communications is usually coded; that is, L is freely generated by a code. In the latter case, the typesetter can be thought of as being the channel that transmits the words of a language to the computer. The communications language in this case is usually not coded, as it could be a natural language or a programming language.

In the above scenarios, the language property of error-detection is of particular interest. Specifically, if the language L is error-detecting *for* the channel γ, then γ cannot transform a word in L to another word in L. As a consequence, when the channel returns a word w which is in L then w is correct. On the other hand, if the returned word is not in L, one can be sure that it is not equal to the intended word and then take appropriate action—for example, request that the word be retransmitted. The objective of this paper is to provide some insights on the concept of error-detection by defining certain error-detecting properties of languages, including coded languages, and obtaining some basic results concerning error-detecting capabilities of regular and other languages, and of known classes of codes. To keep the basic definitions general, we use the framework of P-channels (see [9]) restricted to the case of finite words. This channel model is very general and includes the case of SID-channels which were presented in [8] and further extended in [10]. 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, in the context of data communications, errors of types ι and δ are called *synchronization errors*, as they cause, or are caused by, loss of synchronization. The operation $\odot$ is used to combine error types. In particular, we consider the following set of error types

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

For every error type τ the expression $\tau(m, \ell)$ denotes the channel that permits a total of at most m errors of type τ in any ℓ (or less) consecutive symbols of a message—see the next section for a formal definition. In this case, we assume that m and ℓ are positive integers.

The paper is organized as follows. The next section gives some basic concepts about words, word factorizations, P-channels, and SID-channels. Section 3 defines the basic error-detecting properties of languages, provides examples to illustrate these properties, and contains a few results on error-detection for P- and SID-channels. 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. In Section 4 the concept of error-detection with finite delay is introduced which is a natural generalization of the code property of unique decodability with finite delay. It is shown that no coded language is error-detecting with delay 0 for any SID-channel that permits insertions. Section 5 discusses certain error-detecting capabilities of uniform, solid and shuffle codes, and provides a construction of a language which is maximal error-detecting for the channel $\delta(m,\ell)$ and whose redundancy is asymptotically optimal. Finally, Section 6 contains a few concluding remarks.

2. Basic notation and 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 *alphabet*, X, is a non-empty set of symbols. A *word* (*over* X) is a mapping $w : \{0, 1, \ldots, n-1\} \to X$ for some $n \in \mathbb{N}_0$. In this case, we write I_w to denote the domain $\{0, 1, \ldots, n-1\}$ 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 $X^+ = X^* \backslash \{\lambda\}$. 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 use the symbol X for an arbitrary but fixed finite alphabet.

Let L be a language. A *factorization over* L is a mapping $\varphi : \{0, 1, \ldots, n-1\} \to L$ for some $n \in \mathbb{N}_0$. As before, we write I_φ to indicate the domain $\{0, 1, \ldots, n-1\}$ 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 $W^{\leqslant n} = \bigcup_{i=0}^{n} W^i$.

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. Since a word over X can be viewed as a factorization over X, we also refer to words as messages (over X).

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 obtained 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,

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

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, let Y be a subset of X^*, and let φ be a factorization over Y. A factorization φ' over $\langle Y\rangle_\gamma$ is *γ-admissible for φ* if

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

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

Example 1. Consider the word $y = aabbaa$, where $a, b \in X$, and its factorization φ over $K = \{aa, bb\}$ such that $\varphi = (aa, bb, aa)$. Consider also a channel γ that allows at most one deletion in any 2 consecutive input symbols—as we shall see next, γ is an SID-channel. Then, $y' = abaa$ is a possible output in $\langle y\rangle_\gamma$ if one deletes the symbols $y(0)$ and $y(2)$ in y. Then, the factorization φ' of y' over $\langle K\rangle_\gamma$ such that $\varphi' = (a, b, aa)$ is γ-admissible for φ. On the other hand, for the same channel γ, and for $K = \{ab, ba\}$ and $y = abba$, one has the following: $\varphi = (ab, ba)$ is a factorization of y over K and $\varphi' = (a, a)$ is a factorization of $y' = aa$ over $\langle K\rangle_\gamma$ such that $\varphi'(i) \in \langle \varphi(i)\rangle_\gamma$ for $i \in \{0, 1\}$. But $y' \notin \langle \varphi(0)\varphi(1)\rangle_\gamma$ since the symbols $y(1)$ and $y(2)$ of y cannot be both deleted. Hence, φ' is not γ-admissible for φ.

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

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

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

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

Channels satisfying properties $\mathscr{P}_1$–$\mathscr{P}_3$ are called *P_*-channels.* They differ from the P-channels defined in [9] 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 $\mathscr{P}_0$ of P-channels is omitted here. We note that properties $\mathscr{P}_2$ and $\mathscr{P}_3$ imply $(y|y) \in \gamma$ for all $y \in X^*$.

We now define a certain class of SID-channels—see [10] for the full class of SID-channels. Although the formal definition is not required for the proofs in the present paper, it is included here so that the reader can clearly understand how these channels affect messages. The main tool in the definition is the set of error functions. An error function is represented by a string of basic error functions and is applied on messages on a symbol-by-symbol basis. Consider, for instance, the message $x = abab$ over the alphabet $\{a, b\}$ and consider a channel that would allow one substitution, one insertion and one deletion in x. As $x = \lambda a\lambda b\lambda a\lambda b\lambda$, we see that there are four possible positions for a substitution (the four symbols of x), five possible positions for an insertion (the five λs), and four possible positions for a deletion. Thus, $baaa$ is a possible output from x by inserting a b in front of x, substituting the first b of x with a, and deleting the last b of x. This effect can be expressed by applying the sequence of basic error functions $\mathbf{i}_b, \mathbf{e}, \mathbf{e}, \mathbf{s}, \mathbf{e}, \mathbf{e}, \mathbf{e}, \mathbf{d}, \mathbf{e}$ to each of the nine positions of x, respectively, where $\mathbf{e}$

is the identity function (no error at that position), $\mathbf{i}_b$ is a function that replaces λ by b (insertion in that position), $\mathbf{s}$ is the function that reverses symbols, and $\mathbf{d}$ is a function that replaces an alphabet symbol by λ (deletion at that position). Thus, if we consider the error function $\mathbf{h} = \mathbf{i}_b\mathbf{eeseeede}$, then

$$\mathbf{h}(x) = \mathbf{i}_b(\lambda)\mathbf{e}(a)\mathbf{e}(\lambda)\mathbf{s}(b)\mathbf{e}(\lambda)\mathbf{e}(a)\mathbf{e}(\lambda)\mathbf{d}(b)\mathbf{e}(\lambda) = baaa.$$

Generally, when x is a message of length n, an error function $\mathbf{h}$ can be applied on x provided that $|\mathbf{h}| = 2n + 1$.

The *alphabet G of basic error functions* consists of the following symbols: The symbol $\mathbf{d}$ which denotes the *deletion function* $\mathbf{d}: X \to \{\lambda\}$ such that $\mathbf{d}(a) = \lambda$ for all $a \in X$. For each non-empty word u, the symbol $\mathbf{i}_u$ which denotes the *insertion function* $\mathbf{i}_u : \{\lambda\} \to \{u\}$. Symbols of the form $\mathbf{s}$ which denotes a *substitution function* $\mathbf{s}: X \to X$ such that $\mathbf{s}(a) \neq a$, for all $a \in X$. Finally, the symbol $\mathbf{e}$ which denotes the *identity function* $\mathbf{e}: X \cup \{\lambda\} \to X \cup \{\lambda\}$ such that $\mathbf{e}(a) = a$, for all $a \in X \cup \{\lambda\}$.

We set $G_\varepsilon = \{\mathbf{e}\}$, $G_\delta = \{\mathbf{d}\}$, $G_\iota = \{\mathbf{i}_u \mid u \in X^+\}$, and G_σ equal to the set of all substitution functions $\mathbf{s}$ from X into X. Hence, the alphabet of error function symbols can be written as $G = G_\varepsilon \cup G_\sigma \cup G_\iota \cup G_\delta$. Moreover, for every error type τ, the set G_τ consists of the basic error functions of all the error types that occur in τ. For example, $G_{\sigma \odot \iota} = G_\sigma \cup G_\iota$ and $G_{\sigma \odot \iota \odot \delta} = G_\sigma \cup G_\iota \cup G_\delta$.

Definition 2. An error function is a word $\mathbf{h}$ over G such that $|\mathbf{h}|$ is odd and, for all $i \in I_{\mathbf{h}}$,

$$\mathbf{h}(i) \in \begin{cases} G_\varepsilon \cup G_\iota & \text{if } i \text{ is even;} \\ G_\varepsilon \cup G_\delta \cup G_\sigma & \text{if } i \text{ is odd.} \end{cases}$$

We use the symbol $\mathscr{H}$ to denote the set of error functions. Moreover, if τ is an error type, we write $\mathscr{H}_\tau$ for the set $\mathscr{H} \cap (G_\tau \cup G_\varepsilon)^*$.

The set of error functions is equipped with a product operation, '$\cdot$', such that $(\mathscr{H}, \cdot)$ is a monoid whose neutral element is $\mathbf{e}$. Specifically, if $\mathbf{h}$ and $\mathbf{g}$ are error functions such that $|\mathbf{h}| = 2n + 1$, their product $\mathbf{h} \cdot \mathbf{g}$ is defined as the usual concatenation of words, except at the point where the last symbol of $\mathbf{h}$ and the first symbol of $\mathbf{g}$ are concatenated; that is the symbols $\mathbf{h}(2n)$ and $\mathbf{g}(0)$ become one symbol, $\mathbf{c}$, as follows:

$$\mathbf{c} = \begin{cases} \mathbf{h}(2n) & \text{if } \mathbf{g}(0) = \mathbf{e}; \\ \mathbf{g}(0) & \text{if } \mathbf{h}(2n) = \mathbf{e}; \\ \mathbf{i}_{u_1 u_2} & \text{if } \mathbf{h}(2n) = \mathbf{i}_{u_1} \text{ and } \mathbf{g}(0) = \mathbf{i}_{u_2}. \end{cases}$$

For example, $(\mathbf{ede}) \cdot (\mathbf{i}_a\mathbf{se}) = \mathbf{edi}_a\mathbf{se}$, $(\mathbf{ede}) \cdot (\mathbf{ese}) = \mathbf{edese}$, and $(\mathbf{edi}_a) \cdot (\mathbf{i}_b\mathbf{se}) = \mathbf{edi}_{ab}\mathbf{se}$. Finally we note that when $\mathbf{h}$ can be written as $\mathbf{f}_1 \cdot \mathbf{g} \cdot \mathbf{f}_2$, the error function $\mathbf{g}$ is called an *$\mathscr{H}$-factor* of $\mathbf{h}$.

The function $\mathcal{N}: \mathcal{H} \to \mathbb{N}_0$ is defined such that, for every $\mathbf{h} \in \mathcal{H}$, $\mathcal{N}(\mathbf{h})$ is the number of errors that occur in $\mathbf{h}$. More formally, $\mathcal{N}(\mathbf{h})$ is the sum of

- the number of symbols $\mathbf{d}$ and $\mathbf{s}$ that occur in $\mathbf{h}$, *plus*
- $|u|$ for each symbol $\mathbf{i}_u$ that occurs in $\mathbf{h}$.

For example, $\mathcal{N}(\mathbf{eeede}) = 1$ and $\mathcal{N}(\mathbf{eei}_a\mathbf{eedeseei}_{ab}) = 5$.

Now for every positive integers m and ℓ and for every error type τ, the set $\mathcal{H}_{m,\ell}(\tau)$ consists of all the error functions $\mathbf{h}$ in $\mathcal{H}_\tau$ such that $\mathcal{N}(\mathbf{g}) \leqslant m$ for every $\mathcal{H}$-factor $\mathbf{g}$ of $\mathbf{h}$ with $|\mathbf{g}| \leqslant 2\ell + 1$. Finally, the expression $\tau(m,\ell)$ denotes the SID-channel:

$$\{(y'|y) \mid y \in X^* \quad \text{and} \quad \exists\, \mathbf{h},\ y' = \mathbf{h}(y),\ \mathbf{h} \in \mathcal{H}_{m,\ell}(\tau)\}.$$

That is, $\tau(m,\ell)$ is the channel in which an input message y can result in the output message y' using at most m errors of type τ in every ℓ consecutive symbols of y. It can be shown that every SID-channel is a P_*-channel [9].

Example 2. Consider the word $x = aaaaaaa$ and the SID-channel $\gamma = \iota \odot \delta(2,5)$ that permits at most 2 insertions and deletions in any 5 consecutive symbols. Let $y = abaaaaab$ and let $z = abbaaaaaba$. Observe that y can be obtained from x when γ deletes $x(2)$, inserts a b between $x(0)$ and $x(1)$, and inserts a b at the end of x; that is, $y = \mathbf{h}(x)$ where $\mathbf{h} = \mathbf{eei}_b\mathbf{eedeeeeeeeei}_b$. Hence, as $\mathbf{h} \in \mathcal{H}_{2,5}(\iota \odot \delta)$, $y \in \langle x \rangle_\gamma$. On the other hand, to obtain z from x using a minimum number of errors, one has to insert three b's in the segment $x(1) \cdots x(5)$—including the endpoints of that segment—of length 5; that is, $z = \mathbf{h}(x)$ where $\mathbf{h} = \mathbf{eei}_{bb}\mathbf{eeeeeeeeei}_b\mathbf{ee}$. Then, for the $\mathcal{H}$-factor $\mathbf{g} = \mathbf{i}_{bb}\mathbf{eeeeeeeeei}_b$ of $\mathbf{h}$ one has $|\mathbf{g}| = 11 = 2 \cdot 5 + 1$ but $\mathcal{N}(\mathbf{g}) = 3$. Hence, $z \notin \langle x \rangle_\gamma$.

3. Error detection: definitions and basic results

The classical theory of error-correcting codes deals with channels that permit only 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 [5] or [16]. In this section, we extend the notion of error-detection to the case of arbitrary channels and codes. Moreover, we also define error-detection for arbitrary languages motivated by the fact that natural languages are not coded but they possess certain error-detecting capabilities (see [1]).

A language L is called *coded* if there is a code C such that $L = C^*$. If C_1 and C_2 are codes satisfying $C_1^* = C_2^*$ then $C_1 = C_2$ (see [18]). Hence, if L is a coded language there is a unique code C such that $L = C^*$. We write C_L to denote that code.

Definition 3. Let γ be a channel.

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

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

(ii) A code K is $(\gamma,*)$-*detecting*, if the coded language K^* is error-detecting for γ.

The concepts of $(\gamma,*)$-detecting code and $(\gamma,*)$-correcting code are defined in [9]—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 $w_1, w_2 \in K^*$. For the first part of the above definition we note that 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.

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 $|v| \leqslant |x|$; therefore, $\lambda \in \langle v\rangle_\gamma$ and $v \in K \cup \{\lambda\}$ imply $v = \lambda$. On the other hand, as $m < \operatorname{len} K$, 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 $v_1 \in \langle v_2\rangle_\gamma$. Then, $|v_1| \geqslant |v_2|$. 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 $K_0 = \{aaa, bbb\}$, where $a, b \in X$, 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 $w_2 = (aaa)^3$ and $w_1 = (aaa)^2$ 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 $K_1 = \{v_1, v_2 \mid v_1 = aabbb,\ v_2 = abababb\}$, where $a, b \in X$, and the channel $\gamma = \delta(1,7)$. Then, $\langle v_1\rangle_\gamma = \{v_1, abbb, aabb\}$ and

$$\langle v_2\rangle_\gamma = \{v_2, bababb, aababb, abbabb, abaabb, ababbb, ababab\}.$$

Obviously, the empty word cannot be obtained from either v_1 and v_2 through γ and, conversely, neither of v_1 and v_2 can be obtained from the empty word through γ. Moreover, $v_1 \notin \langle v_2\rangle_\gamma$ and $v_2 \notin \langle v_1\rangle_\gamma$. Hence, K_1 is error-detecting for γ. Now 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 $\mathscr{P}_1$ 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 $\psi(0) = aabb$. The second case implies $\psi(1) = bababb$ 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.

In many cases, the code property of $(\gamma,*)$-detection can be studied in terms of the weaker notion of (γ,t)-detecting code, where $t\in\mathbb{N}_0$.

Definition 4. Let γ be a channel and let $t\in\mathbb{N}_0$. A code K is (γ,t)-*detecting*, if

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

The following proposition describes certain relationships between the error-detecting properties given in Definitions 3 and 4.

Proposition 1. *For every P_*-channel γ, the following statements hold true*:

(i) *For all $t\in\mathbb{N}_0$, if a code K is $(\gamma,t+1)$-detecting then K is (γ,t)-detecting and K is error-detecting for γ.*

(ii) *A code K is $(\gamma,*)$-detecting if and only if K is (γ,t)-detecting for all $t\in\mathbb{N}_0$.*

(iii) *For all $t\in\mathbb{N}_0$, there exists an SID-channel γ and a code K such that K is (γ,t)-detecting but K is not $(\gamma,t+1)$-detecting.*

Proof. (i) Consider a code K which is $(\gamma,t+1)$-detecting and the messages $w_1\in K^{\leqslant t}$ and $w_2\in K^*$ such that $w_1\in\langle w_2\rangle_\gamma$. Let $v\in K$. By property $\mathscr{P}_2$ of the channel γ, one has $w_1v\in\langle w_2v\rangle_\gamma$. As $w_1v\in K^{\leqslant t+1}$ and $w_2v\in K^*$, it follows that $w_1v=w_2v$. Hence, $w_1=w_2$ and K is (γ,t)-detecting. Moreover, it is easy to see that K is error-detecting for γ.

(ii) The statement follows easily from the definitions.

(iii) For each t in $\mathbb{N}_0$ consider the SID-channel $\gamma=\gamma(t)=\delta(1,t+2)$ and the code $K=K(t)=\{a^{t+2}\}$, where $a\in X$. 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 $w_1\in\langle w_2\rangle_\gamma$ and $m,n\in\mathbb{N}_0$ with $m\leqslant t$. As only deletions are permitted, $|w_1|\leqslant|w_2|$. 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 $m+1\leqslant n$. 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, $|w_1|\geqslant|w_2|-n$ which together with $m+1\leqslant n$ imply

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

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

Now we show that K is not $(\gamma,t+1)$-detecting. Let $w_1=(a^{t+2})^{t+1}\in K^{\leqslant t+1}$ and $w_2=(a^{t+2})^{t+2}\in K^*$. 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 a 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, and 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. It is well-known that a language L is regular if and only if $\operatorname{syn} L$ is finite (see [17]).

Theorem 1. *Let $m, \ell \in \mathbb{N}$, let τ be an error type that contains the deletion or insertion type, and let $\gamma = \tau(m, \ell)$. Then, the following statements hold true*:

(i) *If a code K is $(\gamma, *)$-detecting then $x^n \notin K$ for all $x \in X^{\leqslant m}$ and for all $n \in \mathbb{N}$.*

(ii) *If a regular language L, other than $\emptyset$ and $\{\lambda\}$, is error-detecting for γ then $m < |\operatorname{syn} L|$.*

Proof. (i) Assume τ contains δ and suppose there are $n \in \mathbb{N}$ and $x \in X^{\leqslant m} \cap X^+$ such that $x^n \in K$. Then, $x^{n(n-1)\ell}$ and $x^{n(n\ell)}$ are different messages in K^*. As $|x^{n\ell}| \geqslant \ell$, the channel can delete x in each of the n factors $x^{n\ell}$ of $x^{n(n\ell)}$. Hence, $x^{n(n-1)\ell} \in \langle x^{n(n\ell)} \rangle_\gamma$ and, therefore, K is not $(\gamma, *)$-detecting. The case of τ containing ι is similar.

(ii) Assume L is regular, with $L \notin \{\emptyset, \{\lambda\}\}$, and suppose $m \geqslant |\operatorname{syn} L|$. Let $w \in L \backslash \{\lambda\}$. If $|w| < |\operatorname{syn} L|$ the channel can erase or introduce w depending on whether δ or ι occurs in τ. Hence, L is not error-detecting for γ. Now if $|w| \geqslant |\operatorname{syn} L|$, a pumping lemma of the regular languages (see [19]) implies that there are words x, y, z such that $w = xyz$, $1 \leqslant |y| \leqslant |\operatorname{syn} L|$, and $xy^n z \in L$ for all n in $\mathbb{N}_0$. As xyz and xz are in L the channel can transform one of these words to the other; therefore, L is not error-detecting for γ. □

4. Error detection with finite delay

Although error-detection is a basic property of a communications language, the process of decoding a message of such a language might require unbounded memory. This is because the decoder needs to see the entire message in order to decide whether it is correct. Moreover, the message is either accepted or rejected in its entirety. On the other hand, it is possible to define language and automaton (or transducer) properties that allow one to decode a message, possibly partially, using bounded memory [12, 6, 2–4]. This section introduces the code property of error-detection with finite delay which ensures that, as long as a sufficient number of consecutive codewords is received, the process of decoding those codewords can begin before receiving the rest of the message. On the other hand, the decoder can signal an error if it receives a part of the message which is not the concatenation of a sufficient number of codewords. In this case, the decoding process gets suspended and what follows depends on the communication protocol—usually involving retransmission techniques.

Definition 5. Let γ be a channel.

(i) A code K is said to have *finite* $(\gamma$,*$)$*-detection delay*, if there is a non-negative integer d such that

$$\forall v \in K,\ \forall z \in K^d X^*,\ \forall w \in K^*,$$
$$vz \in \langle w \rangle_\gamma \rightarrow \exists u \in K^*,\ w = vu \quad \text{and} \quad z \in \langle u \rangle_\gamma.$$

In this case, we say that K has $(\gamma, *)$-detection delay d.

(ii) A coded language L is *error-detecting for γ with finite delay*, if it is error-detecting for γ and the code C_L has finite $(\gamma, *)$-detection delay. In this case, if C_L has $(\gamma, *)$-detection delay d, for some $d \in \mathbb{N}_0$, we say that L is error-detecting for γ with delay d.

The property of finite $(\gamma, *)$-detection delay is analogous to the code property of finite decoding delay for error-free channels (see [2] where the term finite deciphering delay is used, or [3] where the term bounded deciphering delay is used). Specifically, a language K is said to have finite decoding delay d if

$$\forall v, v' \in K,\ \forall z \in K^d X^*, \quad vz \in v'K^* \to v = v'.$$

Remark 1. Let γ be a P_*-channel, let $d \in \mathbb{N}_0$, and let K be a code that has $(\gamma, *)$-detection delay d. Then, the following statements hold true:

(i) The code K has $(\gamma, *)$-detection delay $d + 1$.

(ii) The code K has finite decoding delay d.

Lemma 1. *Let K be a code, let γ be a channel, and let $d \in \mathbb{N}_0$. If K has $(\gamma, *)$-detection delay d then the following property, $\mathcal{Q}_\gamma^d(n, K)$, holds for all $n \in \mathbb{N}_0$:*

$$\begin{aligned}&\forall v \in K^n,\ \forall z \in K^d X^*,\ \forall w \in K^*,\\ &vz \in \langle w\rangle_\gamma \to \exists u \in K^*,\ w = vu \quad \text{and} \quad z \in \langle u\rangle_\gamma.\end{aligned}$$

Proof. Assume that K has $(\gamma, *)$-detection delay d. We use induction on $n \in \mathbb{N}_0$. First, it is easy to see that $\mathcal{Q}_\gamma^d(0, K)$ holds. Now assume $\mathcal{Q}_\gamma^d(n, K)$ holds for some $n \geqslant 0$. We show that $\mathcal{Q}_\gamma^d(n+1, K)$ holds too. Let $v \in K^{n+1}$, $z \in K^d X^*$, and $w \in K^*$ such that $vz \in \langle w\rangle_\gamma$. Then, $v = v_1 v_2$ for some words $v_1 \in K$ and $v_2 \in K^n$. Moreover, as $v_2 z \in K^d X^*$ and K has $(\gamma, *)$-detection delay d, there is $u' \in K^*$ such that $w = v_1 u'$ and $v_2 z \in \langle u'\rangle_\gamma$. Then, $\mathcal{Q}_\gamma^d(n, K)$ implies the existence of a message $u \in K^*$ such that $u' = v_2 u$ and $z \in \langle u\rangle_\gamma$. Thus, we have shown that $w = v_1 u' = vu$ and $z \in \langle u\rangle_\gamma$, for some $u \in K^*$, which implies that $\mathcal{Q}_\gamma^d(n+1, K)$ holds. □

Example 6. The code K_1 of Example 5 has $(\gamma, *)$-detection delay 0. This follows from the fact (shown in Example 5) that when a received message starts with a codeword v in K_1 then this codeword is correctly transmitted. Then, as a consequence of $\mathcal{Q}_\gamma^0(3, K_1)$, if $v_1 v_2 v_1 aababb$ is a *prefix* of the received message, the words v_1, v_2, v_1 can be decoded correctly and an error is detected when $aababb$ is encountered.

In [2], it is shown that if a language K has finite decoding delay then K is a code. Motivated by this statement, we consider next the question of whether a code with finite $(\gamma, *)$-detection delay is $(\gamma, *)$-detecting.

Theorem 2. *Let γ be a P_*-channel, let $d \in \mathbb{N}_0$, and let K be a code with $(\gamma,*)$-detection delay d. Then, the following statements hold true*:
(i) *The code K is $(\gamma,*)$-detecting if and only if it is (γ,d)-detecting.*
(ii) *If $|K|>1$ then K is $(\gamma,*)$-detecting.*

Proof. (i) The 'only if' part follows immediately from Proposition 1(ii). For the 'if' part, assume K is (γ,d)-detecting and consider messages $w_1, w_2 \in K^*$ with $w_1 \in \langle w_2\rangle_\gamma$. There is $m \in \mathbb{N}_0$ such that $w_1 = v_1 \cdots v_m$ and each v_i is in K. If $m \leqslant d$ then $w_2 = w_1$ as required. Now assume that $m = n+d$ for some $n>0$. As $w_1 \in K^n K^d X^*$ and $w_1 \in \langle w_2\rangle_\gamma$, property $\mathscr{Q}_\gamma^d(n,K)$ of Lemma 1 implies $w_2 = v_1 \cdots v_n u$ and $v_{n+1} \cdots v_{n+d} \in \langle u\rangle_\gamma$ for some message $u \in K^*$. On the other hand, $v_{n+1} \cdots v_{n+d} \in K^{\leqslant d}$ implies $u = v_{n+1} \cdots v_{n+d}$. Hence, $w_2 = w_1$ as required.

(ii) Assume $|K|>1$. By the first part, it is sufficient to show that K is (γ,d)-detecting. Let $v \in K^{\leqslant d}$ and $w \in K^*$ such that $v \in \langle w\rangle_\gamma$. Then, $v=[\varphi]$ and $w=[\psi]$ for some factorizations φ, ψ over K with $|\varphi| \leqslant d$. Choose a codeword x such that

$$x \in \begin{cases} K\backslash\{\psi(|\varphi|)\} & \text{if } |\psi| > |\varphi|; \\ K\backslash\{\varphi(|\psi|)\} & \text{if } |\varphi| > |\psi|; \\ K & \text{if } |\varphi| = |\psi|. \end{cases}$$

As $v \in \langle w\rangle_\gamma$, $\mathscr{P}_2$ implies $vx^d \in \langle wx^d\rangle_\gamma$. As K has $(\gamma,*)$-detection delay d, there is $u \in K^*$ such that $wx^d = vu$ and $x^d \in \langle u\rangle_\gamma$. Hence, $[\varphi]u = [\psi]x^d$. If $|\varphi| = |\psi|$ then, as K is a code, $\varphi = \psi$ and, therefore, $w = v$ as required. If $|\varphi| > |\psi|$ then $d>0$ and $\varphi(|\psi|) = x$ which contradicts $x \in K\backslash\{\varphi(|\psi|)\}$. Finally, if $|\psi| > |\varphi|$ then $u \in \psi(|\varphi|)K^*$. As $x^d \in \langle u\rangle_\gamma$, one has $x^{d+1} \in \langle ux\rangle_\gamma$. But, as K has $(\gamma,*)$-detection delay d, $ux \in xK^*$ which contradicts $x \in K\backslash\{\psi(|\varphi|)\}$. Hence, K is (γ,d)-detecting. □

We note that the second statement of Theorem 2 is not true in general if $|K|=1$. For example, consider the alphabet $X=\{a,b\}$ and the channel $\gamma = \{(a|a),(\lambda|a),(b|b)\}^*$, where the concatenation between two pairs of words is defined naturally: $(y_1'|y_1)(y_2'|y_2) = (y_1'y_2'|y_1y_2)$. Then, γ is a P_*-channel and, as $\lambda \in \langle a\rangle_\gamma$, the code $\{a\}$ is not $(\gamma,*)$-detecting. On the other hand, $\{a\}$ has $(\gamma,*)$-detection delay 0.

The following lemma gives an expression for the maximum number of deletion errors that can occur when a word w is communicated through an SID-channel $\tau(m,\ell)$ that permits deletions.

Lemma 2. *Let $m,\ell \in \mathbb{N}$ with $\ell > m$ and let $D_{m,\ell}: \mathbb{N}_0 \to \mathbb{N}_0$ be such that, for all $n \in \mathbb{N}_0, D_{m,\ell}(n) = \lfloor n/\ell \rfloor m + \min(m, r_\ell(n))$ where $r_\ell(n)$ is the remainder of the integer division n/ℓ. Let τ be an error type that contains the deletion type and let $\gamma = \tau(m,\ell)$. For every word w the following statements hold true*:
(i) *If $z \in \langle w\rangle_\gamma$ then $|w| - |z| \leqslant D_{m,\ell}(|w|)$.*
(ii) *For every non-negative integer k, $k \leqslant D_{m,\ell}(|w|)$ if and only if $\langle w\rangle_\gamma \cap X^{|w|-k} \neq \emptyset$.*

Proof. Let $n=|w|$ and let $q=\lfloor n/\ell \rfloor$. Then $w=w_1\cdots w_q s$ for some words $w_i \in X^\ell$ and $s\in X^{r_\ell(n)}$.

(i) If $z\in \langle w_1\cdots w_q s\rangle_\gamma$, property $\mathscr{P}_1$ implies that $z=w_1'\cdots w_q' s'$ for some words w_i', s' with $w_i'\in\langle w_i\rangle_\gamma$ and $s'\in\langle s\rangle_\gamma$. As the channel γ permits at most m deletions in a word of length ℓ or less, it follows that $|w_i|-|w_i'|\leqslant m$ and $|s|-|s'|\leqslant \min(m, r_\ell(n))$. Hence, $|w|-|z|\leqslant mq+\min(m,r_\ell(n))$.

(ii) The 'if' part follows easily from the first statement of the lemma. For the 'only if' part, let k be a non-negative integer not exceeding $D_{m,\ell}(n)$ and let $p=\lfloor k/m\rfloor$. Then, $mp+r_m(k)\leqslant qm+\min(m,r_\ell(n))$. If $p\leqslant q$ then, for $y=w_{p+1}\cdots w_q s$, one has $r_m(k)\leqslant |y|$ and a word z can be obtained by deleting in w the first m symbols of every w_j, for $j=1,\ldots,p$, and the first $r_m(k)$ symbols of y. Hence, as $k=mp+r_m(k)$, one has $z\in\langle w\rangle_\gamma\cap X^{n-k}$. If $p>q$ then $p=q+1$, $r_m(k)=0$, and $r_\ell(n)\geqslant m$. In this case, $s\in X^m X^*$ and a word z can be obtained by deleting in w the first m symbols of every w_i, for $i=1,\ldots,q$, and the first m symbols of s. Hence, as $k=(q+1)m$, one has $z\in\langle w\rangle_\gamma \cap X^{n-k}$. □

Next it is shown that if an SID-channel permits insertions then there is no coded language which is error-detecting for that channel with delay 0.

Theorem 3. *Let $m,\ell\in\mathbb{N}$, let τ be an error type, and let L be a coded language which is error-detecting for $\tau(m,\ell)$ with delay* 0. *Then, the following statements hold true*:

(i) *The insertion type is not contained in τ.*

(ii) *If $\ell\geqslant 2m$ and τ contains the deletion type then $vX^*v\cap L=\emptyset$ for all $v\in X^+$ with $|v|\leqslant m$.*

Proof. (i) Assume L is error-detecting for γ with delay 0, where $\gamma=\tau(m,\ell)$, but suppose that τ contains the insertion type. Theorem 1(i) implies that the alphabet, X, of L contains at least two symbols and that no symbol of X is in L. Let w be any word in C_L. Then $w=axb$ for some $a,b\in X$ and $x\in X^*$. Let c be any symbol in $X\backslash\{b\}$ and let $n=|axb|$. As the length of the word $(axb)^{1+\ell}$ is $n(1+\ell)$, $(axb)^{1+\ell}=y_1b_1\cdots y_nb_n$ where $b_i\in X$, $y_i\in X^\ell$ and $b_n=b$. As $\tau(m,\ell)$ permits insertions, one has

$$(axb^m b)(y_1c^m b_1)\cdots(y_n c^m b_n)\in\langle axb(axb)^{1+\ell}\rangle_\gamma.$$

As L is error-detecting for γ with delay 0, one has $z\in\langle(axb)^{1+\ell}\rangle_\gamma$, where z is the word $b^m y_1 c^m b_1\cdots y_n c^m b_n$. In obtaining z from $y_1b_1\cdots y_nb_n$ via γ, consider for every $i=1,\ldots,n$ the number k_i of inserted symbols in y_ib_i to the left of b_i, and the number k_{n+1} of inserted symbols at the end of $y_1b_1\cdots y_nb_n$. Then, $0\leqslant k_j\leqslant m$ for all $j=1,\ldots,n+1$. On the other hand, as $|z|=|(axb)^{1+\ell}|+(n+1)m$, one has $k_1+\cdots+k_{n+1}\geqslant(n+1)m$ (with equality when no deletion errors occur); therefore, $k_{n+1}=m$. This implies that the last m symbols of z, namely $c^{m-1}b_n$, are inserted and the preceding symbol, namely c, is the last symbol of $(axb)^{1+\ell}$. Hence, $c=b$ which is a contradiction.

(ii) Assume that the language L is a error-detecting for γ with delay 0, where $\gamma = \tau(m,\ell)$, but suppose there is a word w in L such that $w = vxv$ for some words x and v with $1 \leqslant |v| \leqslant m$. Then, $w \in C_L^+$. Let $n = |w|$ and let $k = |v|$. Then $w^\ell = y_1 v_1 \cdots y_n v_n$ for some words $y_1, \ldots, y_n \in X^{\ell - m}$ and $v_1, \ldots, v_n \in X^m$. As $|y_1| \geqslant k$ and y_1 is a prefix of w^ℓ, there are words $u \in X^{m-k}$ and $x_1 \in X^{\ell - 2m}$ such that $y_1 = vux_1$. Moreover, as $|v_1| = m$, there are words $u_1 \in X^{m-k}$ and $u_2 \in X^k$ such that $v_1 = u_1 u_2$. Hence,

$$w^{1+\ell} = (vxv)(vux_1u_1u_2)y_2v_2 \cdots y_nv_n.$$

Now consider the word $z = vxvx_1u_1y_2 \cdots y_n$ which can be obtained by deleting in $w^{1+\ell}$ the suffix v of the first w and the words $u, u_2, v_2, \ldots, v_n$. Then, $z \in \langle w^{1+\ell} \rangle_\gamma$ and Lemma 1 implies $y \in \langle w^\ell \rangle_\gamma$, where $y = x_1u_1y_2 \cdots y_n$. By Lemma 2, one has

$$\begin{aligned} |w^\ell| - |y| &\leqslant D_{m,\ell}(|w^\ell|) \\ \Rightarrow n\ell - (\ell - 2m + m - k + (n-1)(\ell - m)) &\leqslant mn \Rightarrow k \leqslant 0 \end{aligned}$$

which is a contradiction. Hence, $vX^*v \cap L = \emptyset$. □

5. Error-detecting languages and codes for SID channels

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. Moreover, in this section, we provide a construction of a maximal error-detecting language for the channel $\delta(m,\ell)$ and give an example of a uniform coded language which is error-detecting for $\delta(m,\ell)$ and $\iota(m,\ell)$ with finite delay. 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 2. *Let $m, \ell \in \mathbb{N}$, 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 $\mathscr{P}_1$ 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, $w_1 = w_2$. □

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 [9]. Recent results on solid codes can be found in [7] and [11].

The proof of the following result 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 as follows:

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

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

Theorem 4. *Let $\ell \in \mathbb{N}$, let γ be the channel $\sigma \odot \iota \odot \delta(1, \ell)$, and let K be a finite solid code with* $\operatorname{maxlen} K \leqslant \ell$. *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 $\mathscr{P}(\gamma, K)$ 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 $p \in I_k$ 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

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

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 $p \leqslant r$ and w is either equal to $\psi(r+1)$ or to a non-empty proper prefix of $\psi(r+1)$. The former case implies $\mu(p) \in \langle K^2K^*\rangle_\gamma \cap K$ 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 $\kappa(r+1)$. Thus, $\psi(r+1)$ is of the form $x_{r+1}y_{r+1}$ or $x_{r+1}a_{r+1}y_{r+1}$. If $|w| \leqslant |x_{r+1}|$ 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} which is a contradiction.

Now 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, $|s| \leqslant |y_p|$. 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. □

Example 7. The code K_1 of Example 5 is a $(\gamma, 1)$-detecting solid code, where $\gamma = \sigma \odot \iota \odot \delta(1, 7)$. Hence, Theorem 4 implies that K_1 is $(\gamma, *)$-detecting as well.

Let us consider now certain 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, $\mathrm{PS}_{n+1} \subseteq \mathrm{PS}_n$ for all $n \in \mathbb{N}$. The class OS_n of *outfix-shuffle codes of index n* is defined analogously: $x_0 \cdots x_n \in K$ 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 $\mathrm{OS}_{n+1} \subseteq \mathrm{OS}_n$ for all $n \in \mathbb{N}$. We refer the reader to [9] for further results on shuffle codes.

Proposition 3. *Let $m, \ell \in \mathbb{N}$ with $m < \ell$, and let K be a code with* $\operatorname{minlen} K > m$ *and* $\operatorname{maxlen} K \leqslant \ell$.

(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 $\gamma = \delta(m, \ell)$. As $\operatorname{minlen} K > m$, $\lambda \notin \langle K \rangle_\gamma$. As γ permits only deletions, $\langle \lambda \rangle_\gamma \cap K = \emptyset$. Moreover, if $x \in K$ and $z \in \langle x \rangle_\gamma$ there is $k \in \mathbb{N}_0$ with $k \leqslant m$ such that $x = x_0 a_0 \cdots x_{k-1} a_{k-1} x_k$ and $z = x_0 \cdots x_{k-1} x_k$, 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 $k \leqslant m$, it follows 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 $\operatorname{minlen} K > m$ 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 $|\kappa(0)| \leqslant \ell$, 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, $0 \leqslant k \leqslant m$ 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^*. Hence, $w_1 = x_0 a_0 \cdots x_{k-1} a_{k-1} x_k s$. 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 3 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.

In the rest of the section we give an example of a uniform coded language M which is error-detecting for $\delta(m,\ell)$ and for $\iota(m,\ell)$ with finite delay, and construct an infinite non-coded language L which is maximal error-detecting for $\delta(m,\ell)$ and whose redundancy is asymptotically optimal. For the language M we follow the approach of separator words used in [13] for constructing uniform codes which are error-correcting with finite delay in the presence of certain SID-channel errors. Moreover, we consider the problem of estimating the redundancy imposed by the error-detecting capabilities of M and L. Normally, the redundancy of a finitely coded language is expressed in terms of the size and the average word length of the code that generates the language. As the language L is not coded, however, we consider the redundancies of the infinite languages M and L as follows (see [14, 9]): Let $W=\{w_n\mid n\in\mathbb{N}_0\}$ be an infinite language such that $|w_n|\leqslant|w_{n+1}|$ for all $n\in\mathbb{N}_0$. Then, the redundancy of W is the function defined by $\varrho_W(n)=|w_n|-\lfloor\log_r n\rfloor-1$ for all $n\in\mathbb{N}_0$, where r is the size of the alphabet X. Intuitively, assuming an unbounded number of possible objects (or messages), $\varrho_W(n)$ gives the number of extra alphabet symbols required to represent the object n using the word w_n as opposed to using the r-ary representation of the number n. Usually, as $n\to\infty$, one gives asymptotic estimates for $\varrho_W(n)$. For two functions $f(n)$ and $g(n)$ we say that

- $f(n)$ is asymptotically upper-bounded by $g(n)$, if $\limsup(f(n)/g(n))\leqslant 1$.
- $f(n)$ is asymptotically lower-bounded by $g(n)$, if $\liminf(f(n)/g(n))\geqslant 1$.
- $f(n)$ is asymptotically equal to $g(n)$, if $f(n)$ is asymptotically upper- and lower-bounded by $g(n)$; or equivalently, if $\lim(f(n)/g(n))=1$.
- $f(n)$ is strictly asymptotically upper-bounded by $g(n)$, if $\limsup(f(n)/g(n))<1$. In this case, $f(n)$ is asymptotically upper-bounded by $g(n)$, but it is not asymptotically equal to $g(n)$.

We also note that, in estimating the redundancy of an infinite language W, the following fact is useful

$$\forall n\in\mathbb{N}_0,\quad N_W(|w_n|-1)\leqslant n<N_W(|w_n|),\tag{1}$$

where $N_W:\mathbb{N}_0\to\mathbb{N}_0$ is the cumulative density function of W such that $N_W(k)$ is the number of words in W whose length is at most k.

Lemma 3. *For every languages L and L', if the redundancy of L' is strictly asymptotically upper-bounded by the redundancy of L then the set $\{t\in\mathbb{N}_0\mid N_{L'}(t)\leqslant N_L(t)\}$ is finite.*

Proof. Assume that the languages L and L' satisfy the premise, but suppose there is a subsequence $\{t_k\}_{k\in\mathbb{N}_0}$ of $\mathbb{N}_0$ such that $N_{L'}(t_k)\leqslant N_L(t_k)$ for all k. Let $L=\{w_n\mid n\in\mathbb{N}_0\}$ and $L'=\{w'_n\mid n\in\mathbb{N}_0\}$ such that $|w_n|\leqslant|w_{n+1}|$ and $|w'_n|\leqslant|w'_{n+1}|$ for all n. Let $f(n)=\varrho_{L'}(n)/\varrho_L(n)$, for all n, and let $s=\limsup f(n)$ with $s<1$. First we show that the set

$\{n \in \mathbb{N}_0 \mid f(n) \geqslant 1\}$ is infinite. For each k, let $n_k = \max\{n \mid |w_n| \leqslant t_k\}$. Then, $|w_{n_k+1}| > t_k$ and $N_L(t_k) = 1 + n_k$. Moreover, $N_{L'}(t_k) < n_k$ which implies $N_{L'}(t_k) < N_{L'}(|w'_{n_k}|)$ using (1). As $N_{L'}$ is monotonically increasing, one has $t_k \leqslant |w'_{n_k}|$. Hence, $|w_{n_k}| \leqslant |w'_{n_k}|$ for all k which implies that $\{n \in \mathbb{N}_0 \mid f(n) \geqslant 1\}$ is infinite.

By the assumption about s, the set $\{n \in \mathbb{N}_0 \mid f(n) > s + \varepsilon\}$ is finite, for all $\varepsilon > 0$. On the other hand, $\{n \in \mathbb{N}_0 \mid f(n) > 1 - \varepsilon\}$ is infinite, for all $\varepsilon > 0$, and a contradiction arises when one considers $\varepsilon = (1 - s)/2$. □

Lemma 4. *Let $p, s \in X^*$ and let $k \in \mathbb{N}$. The redundancy of the language $(pX^ks)^*$ is asymptotically equal to $|ps|/k \log_r n$, where $n \to \infty$ and $r = |X|$.*

Proof. Let $L = (pX^ks)^*$, let $m = |ps|$, and let ℓ be a non-negative integer. As the lengths of the words in L are multiples of $m + k$, one has $N_L(\ell) = \sum_{i=0}^{q} r^{ik}$, where $q = \lfloor \ell/(m+k) \rfloor$. Hence, $N_L(\ell) = (r^{k(q+1)} - 1)/(r^k - 1)$. Now assume $L = \{w_n \mid n \in \mathbb{N}_0\}$ with $|w_n| \leqslant |w_{n+1}|$ for all $n \in \mathbb{N}_0$. Then, $N_L(|w_n| - 1) \leqslant n$ and $n < N_L(|w_n|)$. The first inequality implies

$$\frac{r^{k(q+1)} - 1}{r^k - 1} \leqslant n \Rightarrow q \leqslant \frac{1}{k} \log_r(n(r^k - 1) + 1) - 1,$$

where $q = \lfloor (|w_n| - 1)/(m + k) \rfloor$. It follows then that $|w_n| < (m + k)/k \log_r n + \beta_2$ for some real constant β_2. Similarly, one can verify that the second inequality implies $|w_n| > (m + k)/k \log_r n + \beta_1$ for some real constant β_1. Hence, the redundancy of L is asymptotically equal to $|w_n| - \log_r n = (m/k) \log_r n$. □

Proposition 4. *Let $m, \ell \in \mathbb{N}$ with $\ell > 2m + 1$, let $a, b \in X$ with $a \neq b$, and let $M = (a^m X^{\ell - 2m - 1} b)^*$. Then, the following statements hold true*:

(i) *The language M is error-detecting for $\delta(m, \ell)$ with delay* 0.

(ii) *The language M is error-detecting for $\iota(m, \ell)$ with delay* 1.

(iii) *The redundancy of M is asymptotically equal to $(m + 1)/(\ell - 2m - 1) \log_r n$, where $n \to \infty$ and r is the size of the alphabet X.*

Proof. (i) Let $\gamma = \delta(m, \ell)$, let $w \in M$ and assume $a^m xbz \in \langle w \rangle_\gamma$ for some $x \in X^{\ell - 2m - 1}$ and $z \in X^*$. Then $w = [\varphi]$ for some factorization φ over $C_M = a^m X^{\ell - 2m - 1} b$. By $\mathscr{P}_1$, there is a factorization φ' of $a^m xbz$ which is γ-admissible for φ. First we show that $\varphi(0) \neq \varphi'(0)$ is impossible. Indeed, this implies that $\varphi'(0)$ can be obtained from $\varphi(0)$ using exactly d deletions, where $d = |\varphi(0)| - |\varphi'(0)| > 0$. Moreover, in this case, $|\varphi| \geqslant 2$. As $|\varphi(0) a^m| = \ell$, the number of deletions in the prefix a^m of $\varphi(1)$ is at most $m - d$ which implies that $\varphi'(0) a^d$ is a prefix of $a^m x 1 z$. There is a contradiction, however, when we note that $|\varphi'(0) a^d| = |a^m xb|$ and $a \neq b$. Hence, $\varphi(0) = \varphi'(0)$ which implies $[\varphi] = a^m xbu$ and $z \in \langle u \rangle_\gamma$, where $u = \varphi(1) \cdots \varphi(|\varphi| - 1)$.

(ii) Let $\gamma = \iota(m, \ell)$, let $w \in M$ and assume $a^m x_1 b a^m x_2 b y \in \langle w \rangle_\gamma$ for some $x_1, x_2 \in X^{\ell - 2m - 1}$ and $y \in X^*$. Then $w = [\varphi]$ for some factorization φ over $C_M = a^m X^{\ell - 2m - 1} b$. Let $z = a^m x_1 b a^m x_2 b y$ and let n be the number of insertions to the left of the suffix b of $\varphi(0)$. Then, $z \in X^{\ell - m - 1 + n} b X^*$. If $n \geqslant 1$ then, as $n \leqslant m$, $z \in a^m x_1 b a^n X^*$ which implies $z \in X^{\ell-m+n-1}aX^* \cap X^{\ell-m-1+n}bX^*$; a contradiction. Hence, $n=0$ and, therefore, $[\varphi]=a^m x_1 bu$ and $a^m x_2 by \in \langle u\rangle_\gamma$, where $u=\varphi(1)\cdots\varphi(|\varphi|-1)$.

(iii) The statement follows from Lemma 4. □

Theorem 5. *Let $m,\ell \in \mathbb{N}$ with $\ell > m$ and let $L=\bigcup_{k=0}^{\infty} X^{h_k}$, where $h_0=0$ and $h_{k+1}=h_k + 1 + m + m\lfloor h_k/(\ell-m)\rfloor$. Then, the following statements hold true:*

(i) *The language L is maximal error-detecting for $\delta(m,\ell)$.*

(ii) *The redundancy of L is asymptotically upper-bounded by $m/(\ell-m)\log_r n$, where $n\to\infty$ and r is the size of the alphabet X.*

(iii) *There exists no language L' which is error-detecting for $\delta(m,\ell)$ and whose redundancy is strictly asymptotically upper-bounded by the redundancy of L.*

(iv) *The language L is context-sensitive but not context-free.*

The sequence $\{h_k\}_{k\in\mathbb{N}_0}$ has the property that h_{k+1} is the length of a shortest word which cannot result in a word of length h_k using the maximum number of deletions $D_{m,\ell}(h_{k+1})$.

Lemma 5. *For all $m,\ell\in\mathbb{N}$ with $\ell>m$, the following statements hold true:*

(i) *There is a real constant β such that for all $k\in\mathbb{N}_0$, $(\ell-m)/\ell h_{k+1}+\beta<h_k$.*

(ii) *For the function $g:\mathbb{N}_0\to\mathbb{N}_0$ defined by $g(t)=t+1+m+m\lfloor t/(\ell-m)\rfloor$, one has $g(t)=\min\{n\in\mathbb{N}_0 \mid n-D_{m,\ell}(n)>t\}$ and $g(t)-D_{m,\ell}(g(t))=t+1$ for all $t\in\mathbb{N}_0$, where $D_{m,\ell}(n)$ is the function defined in Lemma 2.*

(iii) *For all $k\in\mathbb{N}_0$, $h_{k+1}=\min\{n\in\mathbb{N}_0 \mid n-D_{m,\ell}(n)>h_k\}$.*

(iv) *For every $n\in\mathbb{N}_0$, there is a mapping $f:S\to X^n$, where $S=X^{n-D_{m,\ell}(n)}\cup\cdots\cup X^n$, that satisfies the following properties, for all $u,u'\in S$ and for $\gamma=\delta(m,\ell)$: (a) $u\in X^n$ implies $f(u)=u$; (b) $u\in\langle f(u)\rangle_\gamma$; (c) $|u|=|u'|$ and $f(u)=f(u')$ imply $u=u'$; and (d) $|u|<|u'|$ and $f(u)=f(u')$ imply $u\in\langle u'\rangle_\gamma$.*

Proof. For the first statement, we note that $\lfloor x\rfloor\leqslant x$ for all reals x and, therefore, $h_{k+1}\leqslant h_k+1+m+m/(\ell-m)h_k$. For the second statement, let $t\in\mathbb{N}_0$, and let q and r be the unique non-negative integers with $t=q(\ell-m)+r$ and $0\leqslant r<\ell-m$. For $n=g(t)-1$, one has that $n=q\ell+m+r$ and $D_{m,\ell}(n)=qm+m$. Then, $n>D_{m,\ell}(n)+t$ implies $qm+m>qm+m$ which is a contradiction. On the other hand, for $n=g(t)$, one verifies that $D_{m,\ell}(n)=qm+m$; therefore, $n>D_{m,\ell}(n)+t$ holds true. Hence, $g(t)=\min\{n\in\mathbb{N}_0 \mid n-D_{m,\ell}(n)>t\}$. Now, as $g(t+1)$ is the smallest non-negative integer for which $g(t+1)-D_{m,\ell}(g(t+1))>t+1$ and g is strictly monotonically increasing, it follows that $g(t)-D_{m,\ell}(g(t))\leqslant t+1$. On the other hand, $g(t)-D_{m,\ell}(g(t))\geqslant t+1$. The third statement follows immediately from the previous one.

For the last statement, we note first that every word $u\in S$ can be written in the form $u_1\cdots u_q s$, where $q=\lfloor |u|/(\ell-m)\rfloor$, $u_i\in X^{\ell-m}$, and $s\in X^*$ with $|s|<\ell-m$. The function f will insert $n-|u|$ symbols in u as follows:

$$f(u)=a^m u_1\cdots a^m u_p a^t u_{p+1}\cdots u_q s,$$

such that $a \in X$, $p = \lfloor (n - |u|)/m \rfloor$, and $n - |u| = pm + t$ with $0 \leqslant t < m$. As $|a^m u_j| = \ell$ for all $j = 1, \ldots, p$, it follows that $u \in \langle f(u) \rangle_\gamma$. If $n = |u|$ then $p = t = 0$ and $f(u) = u$ as required. The function f is well-defined if we prove $n - |u| \leqslant qm + m$. This is shown as follows: As $|u| \geqslant n - D_{m,\ell}(n)$, the second statement of the lemma implies $g(|u|) > n \Rightarrow 1 + m + mq > n - |u|$, as required.

Now consider $u, u' \in S$ with $f(u) = f(u')$. Then,

$$f(u) = a^m u_1 \cdots a^m u_p a^t u_{p+1} \cdots u_q s$$

and

$$f(u') = a^m u'_1 \cdots a^m u'_{p'} a^{t'} u'_{p'+1} \cdots u'_{q'} s'.$$

If $|u| = |u'|$ then $q = q'$ and, as $n - |u| = n - |u'|$, it follows that $p = p'$ and $t = t'$ and, therefore, $u = u'$ as required. On the other hand, if $|u| < |u'|$ then $pm + t > p'm + t'$ which implies that either $p > p'$, or $p = p'$ and $t > t'$. If $p > p'$ one has

$$a^m u_1 \cdots a^m u_{p'} a^{t'} = a^m u'_1 \cdots a^m u'_{p'} a^{t'}$$

and

$$a^{m-t'} u_{p'+1} a^m \cdots a^m u_p a^t u_{p+1} \cdots u_q s = u'_{p'+1} \cdots u'_{q'} s'.$$

Then, $u_j = u'_j$ for all $j = 1, \ldots, p'$ and $u' = u_1 \cdots u_{p'} a^{m-t'} u_{p'+1} a^m \cdots a^m u_p a^t u_{p+1} \cdots u_q s$ which implies $u \in \langle u' \rangle_\gamma$ as required. Finally, the case where $p = p'$ and $t > t'$ can be shown similarly. □

Proof of Theorem 5. (i) Suppose $w_1, w_2 \in L$ and $w_1 \in \langle w_2 \rangle_\gamma$, where $\gamma = \delta(m, \ell)$. We want to show $w_1 = w_2$. As γ permits only deletions, it follows that $|w_1| \leqslant |w_2|$ and that $|w_1| = |w_2|$ implies $w_1 = w_2$. First assume $|w_1| < |w_2|$ and $|w_2| = h_k$ for some $k \in \mathbb{N}$. By Lemma 2 $|w_1| \geqslant h_k - D_{m,\ell}(h_k)$ which implies $h_{k-1} < |w_1| < h_k$ using Lemma 5(iii). This in turn implies $w_1 \notin L$ which is a contradiction. Hence, $|w_1| = |w_2|$ and, therefore, $w_1 = w_2$. Thus, L is error-detecting for γ. Now suppose there is a word w in $X^* \backslash L$ such that $L \cup \{w\}$ is error-detecting for γ. Let $n = |w|$. If $n \leqslant m$ then $\lambda \in \langle w \rangle_\gamma$ and, therefore, $w = \lambda$ which is impossible. Hence, there is $k \in \mathbb{N}$ such that $h_k < n < h_{k+1}$. By Lemma 5(iii), $n < h_{k+1}$ implies $n - D_{m,\ell}(n) \leqslant h_k$. By Lemma 2, there is $z \in X^{h_k}$ with $z \in \langle w \rangle_\gamma$. But, as $L \cup \{w\}$ is error-detecting for γ, $w = z \in L$; a contradiction. Hence, L is maximal error-detecting for γ.

(ii) Assume that $L = \{w_i \mid i \in \mathbb{N}_0\}$ such that $|w_i| \leqslant |w_{i+1}|$ for all $i \in \mathbb{N}_0$. Let $n \in \mathbb{N}_0$ and let $s_n = |w_n| - 1$. Then $N_L(s_n) \leqslant n$ and there is $k \in \mathbb{N}_0$ such that $h_k \leqslant s_n < h_{k+1}$. Moreover, $N_L(s_n) = \sum_{i=0}^{k} r^{h_i}$. As $r^{h_k} \leqslant N_L(s_n)$, one has $h_k \leqslant \log_r n$. Then, using Lemma 5(i) and $s_n < h_{k+1}$, it follows that $(\ell - m)/\ell s_n + \beta < \log_r n$ and, therefore, the redundancy of L is asymptotically upper-bounded by $\ell/(\ell - m) \log_r n - \log_r n = m/(\ell - m) \log_r n$.

(iii) Assume that there is a language L' whose redundancy is strictly asymptotically upper-bounded by the redundancy of L. Then, the set $\{n \in \mathbb{N}_0 \mid N_{L'}(n) \leqslant N_L(n)\}$ is finite by Lemma 3. We obtain a contradiction by showing that $N_{L'}(h_k) \leqslant N_L(h_k)$ for all

$k\in\mathbb{N}_0$. For each $k\in\mathbb{N}_0$, let $M_k=L'\cap S$ with $M_0=L'\cap X^0$, where $S=X^{1+h_{k-1}}\cup\cdots\cup X^{h_k}$. By the definition of h_k and Lemma 5(ii), one has $h_k=g(h_{k-1})$ and $h_k-D_{m,\ell}(h_k)=1+h_{k-1}$. Hence, there is a function $f:S\to X^{h_k}$ as defined in Lemma 5(iv). Let $f_k:M_k\to X^{h_k}$ be the restriction of f on M_k. We show that f_k is injective. Let $u,u'\in M_k$ with $f_k(u)=f_k(u')$ and assume $|u|\leqslant|u'|$. If $|u|=|u'|$ then $u=u'$ as required. If $|u|<|u'|$ then $u\in\langle u'\rangle_\gamma$ which is a contradiction as L' is error-detecting for $\gamma=\delta(m,\ell)$. Hence, f_k is injective and this implies $|M_k|=|f_k(M_k)|\leqslant|X|^{h_k}$ for all $k\in\mathbb{N}_0$. Then, one has

$$N_{L'}(h_k)=|M_0\cup\cdots\cup M_k|\leqslant\sum_{i=0}^{k}|X|^{h_i}=N_L(h_k),$$

as required.

(iv) For any k in $\mathbb{N}_0$, one has $h_{k+1}>h_k+1+mh_k/(\ell-m)$ which implies that h_k grows exponentially with respect to k. But, as $\{h_k\mid k\in\mathbb{N}_0\}$ is the length set of L, the language L cannot be context-free.

Now it is well-known that a language is context-sensitive if and only if there is a linearly bounded Turing machine that accepts exactly the words of the language (see [15], for instance). In the case of the language L, given an input word w between the special markers $ and #, a linearly bounded machine needs to test whether $|w|\in\{h_k\mid k\in\mathbb{N}_0\}$ using only the space between the special markers. This can be done incrementally by testing first whether $|w|=h_0$ and then, in general, if $|w|\neq h_k$ test whether $|w|=h_{k+1}$. □

6. Discussion

In this paper, 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 Theorem 1(ii) 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 Theorem 4 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 Theorem 4 and Proposition 3, it appears that codes in that intersection provide certain $*$-error-detecting capabilities for free.

(4) Are there examples of variable-length codes which are $*$-error-detecting for $\delta(m,\ell)$ with delay 0 and more efficient than the uniform code shown in Section 5?

(5) What is the asymptotic redundancy of the language L of Section 5? The answer is interesting as L is most efficient with the property of detecting errors of the deletion type.

References

[1] P. Beckmann, The Structure of Language: A New Approach, The Golem Press, Boulder, CO, 1972.

[2] J. Berstel, D. Perrin, Theory of Codes, Academic Press, Orlando, FL, 1985.

[3] V. Bruyère, Maximal codes with bounded deciphering delay, Theoret. Comput. Sci. 84 (1991) 53–76.

[4] R.M. Capocelli, L. Gargano, U. Vaccaro, Decoders with Initial State Invariance for Multivalued Encodings, Theoret. Comput. Sci. 86 (1991) 365–375.

[5] J. Duske, H. Jürgensen, Codierungstheorie, BI Wissenschaftsverlag, Mannheim, 1977.

[6] W.E. Hartnett (Ed.), Foundations of Coding Theory, Reidel, Boston, 1974.

[7] H. Jürgensen, M. Katsura, S. Konstantinidis, Maximal solid codes, J. Automata Languages Combin. 6 (2001) 25–50.

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

[9] H. Jürgensen, S. Konstantinidis, Codes, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. I, Springer, Berlin, 1997, pp. 511–607.

[10] S. Konstantinidis, An algebra of discrete channels that involve combinations of three basic error types, Inform. and Comput. 167 (2001) 120–131.

[11] N.H. Lâm, Finite maximal solid codes, Preprint 98/17, Vietnam National Centre for Natural Science and Technology, Institute of Mathematics, 1998, Theoret. Comput. Sci. 262 (2001) 333–347.

[12] V.I. Levenshtein, Self-adaptive automata for decoding messages, Dokl. Akad. Nauk. SSSR 141 (1961) 1320–1323 (English translation: Soviet Phys. Dokl. 6 (1962) 1042–1045).

[13] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Dokl. Akad. Nauk. SSSR 163 (1965) 845–848 (in Russian) (English translation: Soviet Phys. Dokl. 10 (1966) 707–710).

[14] V.I. Levenshtein, On the redundancy and delay of decodable coding of natural numbers, Problemy Kibernet. 20 (1968) 173–179 (in Russian) (English translation: Systems Theory Res. 20 (1971) 149–155).

[15] A. Mateescu, A. Salomaa, Aspects of classical language theory, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. I, Springer, Berlin, 1997, pp. 175–251.

[16] S. Roman, Coding and Information Theory, Springer, New York, 1992.

[17] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. I, Springer, Berlin, 1997.

[18] H.J. Shyr, Free Monoids and Languages, 2nd ed., Hon Min Book Company, Taichung, 1991.

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