An Algebra of Discrete Channels That Involve Combinations of Three Basic Error Types

Stavros Konstantinidis

Department of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, B3H 3C3, Canada
E-mail: s.konstantinidis@stmarys.ca

Received December 30, 1996; final manuscript received December 23, 1999

Recently, the author introduced a nonprobabilistic mathematical model of discrete channels, the BEE channels, that involve the error-types substitution, insertion, and deletion. This paper defines an important class of BEE channels, the SID channels, which include channels that permit a bounded number of scattered errors and, possibly at the same time, a bounded burst of errors in any segment of predefined length of a message. A formal syntax is defined for generating channel expressions, and appropriate semantics is provided for interpreting a given channel expression as a communication channel (SID channel) that permits combinations of substitutions, insertions, and deletions of symbols. Our framework permits one to generalize notions such as error correction and unique decodability, and express statements of the form “The code $K$ can correct all errors of type $\xi$” and “it is decidable whether the code $K$ is uniquely decodable for the channel described by $\xi$,” where $\xi$ is any SID channel expression.

Key Words: channel; code; decidable; deletion; error; error correction; insertion; substitution; unique decodability.

1. INTRODUCTION

Recently, the author introduced a nonprobabilistic mathematical model of discrete channels that involves the three basic error types substitution, insertion, and deletion (Konstantinidis, 1999). The model is based on the novelty that errors can be expressed as strings over an alphabet of basic error symbols. Some general conditions on errors that bound the error effects on messages, obtaining thus the class of BEE channels—channels of bounded error effects—are defined. In this paper, we derive an important subclass of BEE channels, the SID channels, which include channels that permit a bounded number of scattered errors, and possibly at the same time, a bounded burst of errors in any segment of predefined length of a message—a preliminary version of SID channels with scattered errors was presented in Jürgensen and Konstantinidis (1996a). For example, a channel that permits a total of at most 5 insertions and deletions in any 21 (or less) consecutive symbols of a message and at the same time a burst of substitutions of length at most 7 in any 50 (or less) consecutive symbols of the message is an SID channel.

In the classical theory of error-correcting codes—for instance, Peterson and Weldon (1972), MacWilliams and Sloane (1977), and Duske and Jürgensen (1977)—the channels considered involve only substitution errors (for example, when a 0 is corrupted to 1, or a 1 to 0) and these errors are restricted in two possible ways: at most $m$ (scattered) errors can occur in a codeword of fixed length, or a burst of errors of length at most $m$ can occur in a codeword of fixed length, where $m$ is a predefined parameter. On the other hand, channels that permit synchronization errors—namely, errors that cause insertions and deletions of symbols in a message—have not been studied systematically in the past. One of the first notable attempts toward this direction is the work in Levenshtein (1966b) where channels that permit insertions, deletions, and reversals of symbols are considered. There also the notion of Levenshtein distance is used to investigate codes capable of correcting the errors of such channels. More recently, in Hollman (1993), the Levenshtein distance is extended to deal with error situations where different bounds are used for errors of different types. Other work on synchronization errors exists, for instance, in Sellers (1962), Ullman (1966), Levenshtein (1970, 1992), Tong (1969), Tenengol’ts (1984),

A different approach in modeling discrete channels is taken in Hartnett (1968, 1974), and generalized in Capocelli and Vaccaro (1989) and Capocelli et al. (1991) using multivalued encodings. Multivalued encodings were introduced in Sato (1979). There it is shown that the property of unique decodability of a finite multivalued encoding is decidable. For further details on how this approach can lead to a model of discrete channels, the reader is also referred to Jürgensen and Konstantinidis (1996b).

Most of the material presented here is rather new and, for this reason, an informal description of the results is given next, before we proceed to the technical details.

Section 2 contains the basic notation used and provides the background required to model BEE channels. First, a general definition of channel is given and the notion of uniquely decodable code with respect to a channel is defined. Then the set of error functions is considered. From a syntactic point of view, error functions are strings over an alphabet of basic error function symbols. From a semantic point of view, they denote ordinary functions that are applied to information messages on a symbol-by-symbol basis. As a result, errors can be treated as individual objects of study. In Section 3, sets of bounded error effects are considered. These are sets of error functions satisfying certain general conditions that guarantee that errors have only local effects and cannot affect portions of a message that lie arbitrarily far. This section also reports some important closure properties of sets of bounded error effects.

Section 4 begins with the notion of error type. The basic error types are the symbols $\sigma$, $\iota$, and $\delta$, indicating, respectively, substitutions, insertions, and deletions. Moreover, the “error” type $\varepsilon$ is included to refer to noiseless channels. Then the operation $\odot$ between two error types $\tau_1$ and $\tau_2$ is used to express some dependence between the errors represented by $\tau_1$ and $\tau_2$. Thus, to each error type $\tau$ one associates a possible SID interpretation $Z(\tau)$ that describes the restrictions on the errors permitted by $\tau$. The set $Z(\tau)$ consists of error functions and is of bounded error effects.

In Section 5, error types are used to define SID channel expressions. Such an expression can be of the form $\tau p$, where $p$ is a parameter that refers to a particular SID interpretation. For example, for $p = (m, L)$ with $m < L$, $\tau(m, L)$ is a channel expression that specifies an SID channel such that at most $m$ scattered errors of type $\tau$ are permitted in any $L$, or less, consecutive symbols of a message. Similarly, the parameters in $P_{\text{burst}} = \{[m, L] | m < L\}$ can be used to define channel expressions of the form $\tau[m, L]$ such that, in the corresponding SID channel, a burst of errors of type $\tau$ can be of length at most $m$, in any $L$ (or less) consecutive symbols of a message. Moreover, the operation $\oplus$ between channel expressions is defined to permit combinations of error types with different interpretations. For example, an SID channel expression can also be of the form $\tau_1 p_1 \oplus \tau_2 p_2$, where the errors of type $\tau_1$ are scattered and the errors of type $\tau_2$ occur in bursts, according to the interpretations referred to by the parameters $p_1$ and $p_2$. Section 5 also considers the question of equivalence between channel expressions and obtains a few results concerning simplifications of channel expressions.

Section 6 discusses the notion of unique decodability for SID channels. A consequence of the results in Konstantinidis (1999) is the following: For given finite code $K$ and channel expression $\xi$, it is decidable whether $K$ is uniquely decodable for the SID channel defined by $\xi$. Finally, Section 7 contains a few concluding remarks.

2. BASIC NOTATION AND BACKGROUND

This section serves three main purposes: first, it provides the basic notation used in the paper; second, it gives a general definition of channel and unique decodability with respect to the channel considered; and third, it defines error functions, which are the basic objects for modeling BEE and SID channels.

For a set $S$, we write $|S|$ to denote the cardinality of $S$ and $2^S$ to denote the set of all subsets of $S$. If $S$ is a subset of $S_1 \times S_2$, then $\text{proj}_1(S) = \{s_1 \in S_1 \mid (s_1, s_2) \in S, s_2 \in S_2\}$ and $\text{proj}_2(S) = \{s_2 \in S_2 \mid (s_1, s_2) \in S, s_1 \in S_1\}$. The set of positive integers is denoted by $\mathbb{N}$ and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. For each $n$ in $\mathbb{N}_0$, we use the symbol $n$ to denote the set $\{0, 1, \ldots, n-1\}$. Such a set is called an index set and is equipped with the standard numeric order “$<$,” namely $0 < 1 < \cdots < n - 1$.

An alphabet, $A$, is a nonempty set of symbols. A word (over $A$) is a mapping $w: I \rightarrow A$, where $I$ is an index set. In this case, we write $I_w$ to denote the domain $I$ of the word $w$. Moreover, as usual, we can denote $w$ by juxtaposing its elements: $w = w(0)w(1) \cdots w(n-1)$, where $n = I_w$. The empty word, $\lambda$,
is the unique word with \( I_\circ = 0 \). The length, \(|w|\), of a word \( w \) is the number \(|I_w|\). If \( B \) is a subset of \( A \) and \( w \) is a word over \( A \), the symbol \(|w|_B\) denotes the number of symbols in \( B \) that occur in \( w \), namely \(|w|_B = |\{i \in I_w \mid w(i) \in B\}|\). The set of all words over \( A \) is denoted by \( A^* \) and \( A^n = A^\setminus\{\lambda\} \). The set \( A^* \) equipped with the usual concatenation of words over \( A \) is a free monoid. For a set of words \( W \), a factorization over \( W \) is a mapping \( \varphi : I \to W \) where \( I \) is an index set. As before, we use \( I_w \) for the domain \( I \) of \( \varphi \). The symbol \([\varphi]\) denotes the word \( \varphi(0)\varphi(1) \cdots \varphi(n-1) \), where \( n = |I_w| \). The operation of concatenation is extended naturally as follows: \( wV = \{vw \mid v \in V\} \) and \( WV = \{wv \mid w \in W, v \in V\} \), where \( V \) and \( W \) are sets of words. For \( n \in \mathbb{N}_0 \) and \( w \in A^* \), the symbol \( w^n \) denotes the word \([\varphi]\) such that \( I_w = n \) and \( \varphi(i) = w \) for all \( i \in I_w \). Similarly, for a set of words \( W \), \( W^n = \{w^n \mid w \in W\} \).

In the sequel, we fix a finite alphabet \( X \) which contains at least the two distinct symbols 0 and 1. A code (over \( X \)) is a nonempty subset \( K \) of \( X^+ \). A message over \( K \) is a word \([\varphi]\), where \( \varphi \) is a factorization over \( K \). Then \( K^+ \) is the set of all messages over \( K \) and \( K^+ \) is the set of all nonempty messages. As a result, a word in \( X^+ \) can also be referred to as a message (over \( X \)).

**Definition 2.1.** A code \( K \) is said to be uniquely decodable, if \([\varphi] = [\psi]\) implies \( \varphi = \psi \), for all factorizations \( \varphi \) and \( \psi \) over \( K \).

The well-known result in Sardinas and Patterson (1953) provides a necessary and sufficient condition for testing whether a given finite code is uniquely decodable. An explicit algorithm for deciding unique decodability of finite codes is given in Markov (1962).

**Definition 2.2.** A channel, \( \gamma \), is a binary relation over \( X^* \), namely \( \gamma \subseteq X^* \times X^* \), such that \( \text{proj}_2(\gamma) = Y^* \) for some nonempty subset \( Y \) of \( X^* \). The set \( \text{Input}(\gamma) = \text{proj}_2(\gamma) \) is the set of inputs and \( \text{Output}(\gamma) = \text{proj}_1(\gamma) \) is the set of outputs of \( \gamma \).

We should note that Jürgensen and Konstantinidis (1996b) gives a more general and elaborate definition of channel that makes it possible to include infinite messages as well as to express whether a channel is stationary.

In analogy to conditional probabilities, we prefer to write \( (y' \mid y) \in \gamma \) instead of \( (y', y) \in \gamma \). Thus, if \( (y' \mid y) \in \gamma \) then \( y' \) is a possible output of \( \gamma \) when \( y \) is used as input. A channel \( \gamma \) is noiseless if \( \gamma = \{(y \mid y) \mid y \in \text{Input}(\gamma)\} \); otherwise, it is noisy. The latter case corresponds to physical channels that introduce errors during information transmission. In this case, it is possible that \( (y' \mid y) \in \gamma \) and \( y' \neq y \).

For every \( y \in \text{Input}(\gamma) \) we define \( \langle y \rangle_\gamma \), to be the set of all possible outputs of \( \gamma \) when \( y \) is used as input; that is,

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

More generally, for \( Y \subseteq \text{Input}(\gamma) \), we have \( \langle Y \rangle_\gamma = \bigcup_{y \in Y} \langle y \rangle_\gamma \).

Now we generalize Definition 2.1 to include codes with the ability to decode uniquely messages over noisy channels. The requirement for unique decodability in the presence of errors is that no two distinct messages can result in the same output of the channel under consideration. Thus, when a word in \( \text{Output}(\gamma) \) is received there is only one possible message from \( K^+ \) that was used as input. In this sense, we can also say that the code \( K \) can correct the errors of \( \gamma \) in the messages over \( K \).

**Definition 2.3.** Let \( \gamma \) be a channel and let \( K \) be a code. Then \( K \) is \((\gamma, *)\)-correcting or uniquely decodable for \( \gamma \) if \( \langle [\varphi] \rangle_\gamma \cap \langle [\psi] \rangle_\gamma \neq \emptyset \) implies that \( \varphi = \psi \), for all factorizations \( \varphi \) and \( \psi \) over \( K \).

Jürgensen and Konstantinidis (1996b) contains several code-related properties such as bounded decoding delay and synchronization with respect to a given channel. Moreover, these properties are shown to be natural generalizations of those considered in the case of noiseless channels (see, for instance, Berstel and Perrin (1985)).

To model the effects of BEE channels on messages, we introduce the error function symbols, which are applied on words over \( X \) on a symbol-by-symbol basis. Consider, for instance, the message \( x = 010 \) and consider a communication channel that would allow one insertion and one deletion in \( x \). The word \( x \) can be written as \( x = \lambda \alpha 0 \lambda \beta 1 \lambda \beta 0 \alpha \) and we see that there are four possible positions for insertions (the four \( \lambda \)'s) and three positions for deletions (the three symbols of \( x \)). Thus, 101 is a possible output of the channel by inserting a 1 in front of \( x \) and by deleting its last 0. This effect can be expressed by
applying the sequence of basic error function symbols \( i_1, e, e, e, d, e \) to each of the seven positions of \( x \), respectively, where \( e \) is the identity function (no error in that position), \( i_1 \) is a function that replaces \( \lambda \) by 1 (insertion in that position), and \( d \) is a function that replaces a symbol in \( X \) by \( \lambda \) (deletion at that position). Thus, if we consider the error function symbol \( h = i_1 \text{ e e e d e e} \), then

\[
h(x) = i_1(\lambda) e(0) e(\lambda) e(1) e(\lambda) d(0) e(\lambda) = 101.
\]

In some applications only certain symbols of the input alphabet \( X \) can be inserted or deleted (see, for instance, Levenshtein (1966a), Roth and Siegel (1994), Bours (1994)). In the rest of the paper we fix three nonempty subsets \( X_\sigma, X_i, X_d \) of the alphabet \( X \), which denote, respectively, the symbols of \( X \) that can be substituted, inserted, and deleted. The alphabet \( G \) of basic error function symbols consists of the following symbols:

- \( d \): which symbolizes the deletion function \( d : X_\sigma \to \{ \lambda \} \) such that \( d(a) = \lambda \) for all \( a \in X_\sigma \). We set \( G_d = \{ d \} \) and we also write dom \( d \) to denote the domain \( X_\sigma \) of \( d \).
- \( i_u \): which symbolizes the insertion function \( i_u : \{ \lambda \} \to \{ a \} \), where \( u \in X_\sigma^+ \). We set \( G_i = \{ i_u \mid u \in X_\sigma^+ \} \).
- \( s \): which symbolizes a substitution function \( s : X_\sigma \to X \) such that \( s(a) \neq a \) for all \( a \in X_\sigma \). We set \( G_s \) equal to the set of all substitution functions \( s \) from \( X_\sigma \) into \( X \). Moreover, we write dom \( s \) to denote the domain \( X_\sigma \) of \( s \).
- \( e \): which symbolizes the identity or no error function \( e : X \cup \{ \lambda \} \to X \cup \{ \lambda \} \) such that \( e(a) = a \), for all \( a \in X \cup \{ \lambda \} \). We set \( G_e = \{ e \} \) and we also write dom \( e \) to denote the domain of \( e \).

Hence, the alphabet of error function symbols can be written as \( G = G_e \cup G_s \cup G_i \cup G_d \).

**Definition 2.4.** An error function symbol is a word \( h \) over \( G \) such that \( |h| \) is odd and, for all \( i \in I_h \),

\[
h(i) \in \begin{cases} G_e \cup G_s, & \text{if } i \text{ is even;} \\ G_e \cup G_i \cup G_d, & \text{if } i \text{ is odd.} \end{cases}
\]

We use the symbol \( \mathcal{H} \) to denote the set of error function symbols.

The set \( \mathcal{H} \) is equipped with a concatenation “\( \cdot \)” which, for two error function symbols \( h \) and \( g \), is defined as the usual concatenation of words, except at the point where the last symbol of \( h \) and the first symbol of \( g \) are concatenated. More specifically, if \( |h| = 2n + 1 \), the symbols \( h(2n) \) and \( g(0) \) become one symbol, \( c \), as

\[
c = \begin{cases} h(2n), & \text{if } g(0) = e; \\ g(0), & \text{if } h(2n) = e; \\ i_{a_1} i_{a_2}, & \text{if } h(2n) = i_{a_1} \text{ and } g(0) = i_{a_2}. \end{cases}
\]

For example, \( (\text{ede}) \cdot (i_3 \text{ se}) = \text{ediede}, (\text{ede}) \cdot (\text{ese}) = \text{eedese} \), and \( (\text{edi}) \cdot (i_3 \text{ se}) = \text{ediidese} \). It is easy to verify that the identity symbol \( e \) is the neutral element of \( \mathcal{H} \) and that \( \mathcal{H} \) is a monoid. This monoid should not be confused with the free monoid \( G^* \), the neutral element of which is the empty word \( \lambda \). In fact, when \( h \) and \( g \) are in \( \mathcal{H} \), the word \( hg \in G^* \) is never in \( \mathcal{H} \) since \( |hg| \) is even.

For two error function symbols \( h \) and \( g \) we say that:

- \( g \) is an \( \mathcal{H} \)-prefix of \( h \), if \( h = g \cdot f \) for some error function symbol \( f \). We write \( \text{Pref}_{\mathcal{H}}(h) \) to denote the set of all \( \mathcal{H} \)-prefixes of \( h \).
- \( g \) is an \( \mathcal{H} \)-suffix of \( h \), if \( h = f \cdot g \) for some error function symbol \( f \). We write \( \text{Suff}_{\mathcal{H}}(h) \) to denote the set of all \( \mathcal{H} \)-suffixes of \( h \).
- \( g \) is an \( \mathcal{H} \)-infix of \( h \), if \( h = f_1 \cdot g \cdot f_2 \) for some error function symbols \( f_1 \) and \( f_2 \). We write \( \text{Inf}_{\mathcal{H}}(h) \) to denote the set of all \( \mathcal{H} \)-infixes of \( h \).

Observe that an \( \mathcal{H} \)-prefix (or suffix) \( g \) of \( h \) is also an \( \mathcal{H} \)-infix of \( h \) since \( g = e \cdot g \) (or \( g = g \cdot e \)).
The error function symbol \( h \) can be applied to an input word \( x \) of \( X^* \), provided that \( x \) is in the domain of \( h \). The domain, \( \text{dom} \ h \), of \( h \) with \( |h| = 2n + 1 \) consists of all the messages \( x \) in \( X^* \) such that \( x(i) \) is in the domain of \( h(2i + 1) \), for all \( i \in n \). Then, for \( x \) in the domain of \( h \), \( h(x) \) is the word

\[
\mathbf{c}_0(\lambda)\mathbf{c}_1(x(0))\mathbf{c}_2(\lambda)\mathbf{c}_3(x(1))\cdots \mathbf{c}_{2n-1}(x(n-1))\mathbf{c}_{2n}(\lambda),
\]

where \( \mathbf{c}_i = h(i) \) for \( i \in I_h \). For example, if \( h = i_0 i_{10}i_{20}i_{30}i_{40} \) and \( X_n = X = \{0, 1\} \), then 111 is in the domain of \( h \) and \( h(111) = 0110 \).

In the sequel, for simplicity of reference, we use the term `error function` instead of error function symbol.

### 3. SETS OF BOUNDED ERROR EFFECTS AND BEE CHANNELS

In this section, we consider sets of bounded error effects. These sets consist of error functions that satisfy some general conditions that seem appropriate for modeling communication channels. Then a BEE channel is described by a set of bounded error effects, the error functions of which represent all the possible error situations permitted by the channel.

**Definition 3.1.** Let \( Z \) be an infinite set of error functions. We say that \( Z \) is of bounded error effects, if there is a finite subset \( F \) of \( Z \), called a (finite) support of \( Z \), such that the following conditions hold true, for \( L_F \in \mathbb{N} \) defined by

\[
2L_F + 1 = \max\{|h| : h \in F\}:
\]

(i) An error function \( h \) is in \( Z \) if and only if every \( H \)-infix \( g \) of \( h \) with \( |g| \leq 2L_F + 1 \).

(ii) If \( g \) is in \( F \) and \( h \) is an error function with \( h \in e^*ge^* \) and \( |h| \leq 2L_F + 1 \), then \( h \) is in \( F \).

(iii) No error function in \( (Gd)^{L_F} G \) is in \( F \).

We use the symbol \( \mathfrak{S}_{\text{BEE}} \) to denote the class of sets of bounded error effects.

Condition (i) requires that error functions in \( Z \) are made up of elements in the support of \( Z \). Condition (ii) says that if the error \( g \) is permitted, then also any error of the form \( e^*ge^* \) is permitted. According to condition (iii), there is a bound on the length of messages that can be completely erased by an error in \( Z \).

The next statement provides a necessary and sufficient condition for testing whether the error function \( h_1 \cdot h_2 \) is in \( Z \) for two given error functions \( h_1 \) and \( h_2 \) in \( Z \). We should note that, in Konstantinidis (1999), this condition is included in the definition of set of bounded error effects.

**Proposition 3.2.** Let \( Z \) be a set of bounded error effects with support \( F \), and let \( h_1 \) and \( h_2 \) be error functions in \( Z \) with \( |h_1 \cdot h_2| > 2L_F + 1 \). Then \( h_1 \cdot h_2 \) is in \( Z \) if and only if \( g_1 \cdot g_2 \) is in \( F \) for every \( H \)-suffix \( g_1 \) of \( h_1 \) and for every \( H \)-prefix \( g_2 \) of \( h_2 \) with \( |g_1 \cdot g_2| \leq 2L_F + 1 \).

**Proof.** First, assume \( h_1 \cdot h_2 \) is in \( Z \) and consider two error functions \( g_1 \) and \( g_2 \) satisfying the condition given in the proposition. Then it follows that \( g_1 \cdot g_2 \) is an \( H \)-infix of \( h_1 \cdot h_2 \). Moreover, as \( |g_1 \cdot g_2| \leq 2L_F + 1 \), Definition 3.1 implies that \( g_1 \cdot g_2 \) is in \( F \). For the converse, assume \( g_1 \cdot g_2 \) is in \( F \) for every error function \( g_1 \) and \( g_2 \) satisfying the condition in the proposition, but suppose \( h_1 \cdot h_2 \) is not in \( Z \). Then there is an \( H \)-infix \( g \) of \( h_1 \cdot h_2 \) that is not in \( F \). As \( h_1 \in Z \) and \( h_2 \in Z \), it follows that there are \( g_1 \) and \( g_2 \) with \( g = g_1 \cdot g_2 \) such that \( g_1 \) is an \( H \)-suffix of \( h_1 \) and \( g_2 \) is an \( H \)-prefix of \( h_2 \). This, however, implies that \( g \notin F \), which is a contradiction. Hence, \( h_1 \cdot h_2 \) must be in \( Z \).

Next, we list a few known results on sets of bounded error effects from Konstantinidis (1999).

**Proposition 3.3.** Let \( Z \) be a set of bounded error effects with given support \( F \). Then the following statements hold true:

(i) If \( h \) is in \( Z \) and \( g \) is an \( H \)-infix of \( h \), then \( g \) is in \( Z \).

(ii) If \( g \) is in \( Z \) and \( h \) is an error function with \( h \in e^*ge^* \), then \( h \) is in \( Z \).

(iii) The set \( Z \) is a recursive subset of \( H \).

(iv) The error function \( e^{2i+1} \) is in \( Z \) for all \( i \) in \( \mathbb{N} \).
Theorem 3.4. If the sets $Z_1$ and $Z_2$ of bounded error effects have a common support $F$, then $Z_1 = Z_2$.

As a result of Theorem 3.4, we write $[F]$ to denote the unique set of bounded error effects with support $F$. Now the BEE channels are defined as follows.

Definition 3.5. A BEE channel is a channel $\gamma$ such that Input ($\gamma$) = $X^*$ and

$$\gamma = \{(y' | y) \mid y', y \in X^*, \exists h \in [F] : y' = h(y)\},$$

for some set $[F]$ of bounded error effects. In this case, we say that $\gamma$ is representable by $F$, or that $F$ is a representation of $\gamma$.

We note that if $\gamma$ is a BEE channel then $(y | y) \in \gamma$ for every $y \in X^*$. This follows from statement (iv) of Proposition 3.3. Consequently, if a code $K$ is uniquely decodable for $\gamma$ then $K$ is also uniquely decodable (in the error-free sense). The next theorem states that unique decodability in the presence of errors from BEE channels is a decidable property of finite codes.

Theorem 3.6. The following problem is decidable:

Instance: The representation $F$ of a BEE channel $\gamma$ and a finite code $K$.

Question: Is $K$ uniquely decodable for $\gamma$?

We close this section with some closure results on sets of bounded error effects. These are needed in the next sections to define the class of SID channels.

Theorem 3.7. (i) Let $[F]$ be a set of bounded error effects and let $H$ be a subset of $G$ such that $e \in H$. Then the set $[F] \cap H^*$ is of bounded error effects with support $F \cap H^*$.

(ii) For every $n \in \mathbb{N}$ and for every sets $[F_0], \ldots, [F_{n-1}]$ of bounded error effects, the set $Z = \bigcap_{i \in \mathbb{N}} [F_i]$ is of bounded error effects. Moreover, when the supports $F_0, \ldots, F_{n-1}$ are given, a support of $Z$ can effectively be constructed.

Proof. (i) Let $Y = [F] \cap H^*$ and let $D = F \cap H^*$. We show that $Y = [D]$ using Definition 3.1.

First we note that, by Proposition 3.3(iv), $e^{2L+1} \in F$ and, as $e \in H$, $e^{2L+1} \in D$. Hence, $L_D = L_F$.

Now, for condition (i) of Definition 3.1, assume $h \in Y$ and $g \in \text{Inf}_H(h)$ with $|g| \leq 2L_D + 1$. As $h \in [F]$ and $[F]$ is of bounded error effects, one has $g \in F$. As $g \in H^*$, it follows that $g \in D$. Now assume that $g \in D$ for every $H$-infixed $g$ of $h$ with $|g| \leq 2L_D + 1$. Then also $g \in F$ and, therefore, $h \in [F]$. Moreover, as every $H$-infixed of $h$ is in $H^*$, it follows that $h \in H^*$. Hence, $h \in Y$. For condition (ii), assume $g \in D$ and $h \in e^ge^w$ with $|w| \leq 2L_D + 1$. As $g \in F$, $h$ is in $F$ as well, but also $h \in H^*$, which implies that $h \in D$. For condition (iii), if there is $g \in (Gd)^{L_D}G \cap (F \cap H^*)$, then $g \in (Gd)^{L_D}G \cap F$ as well, which contradicts the fact that $[F]$ is of bounded error effects.

(ii) Let $L = \max\{L_{F_i} \mid i \in \mathbb{N}\}$. For each $i \in \mathbb{N}$, define $F'_i = F_i$ if $L_{F_i} = L$, or $F'_i = \{h \mid h \in [F_i], |h| \leq 2L + 1\}$ if $L_{F_i} < L$. We note that the sets $F'_i$ of the latter case can be constructed by enumerating all $h \in H$ of length at most $2L + 1$, and by testing whether each $h$ is in $[F_i]$. Moreover, the test is possible as a result of Proposition 3.3(iii). Now define $F = \bigcap_{i \in \mathbb{N}} F'_i$. One verifies now that the sets $Z$ and $F$ satisfy the three conditions of Definition 3.1 and, therefore, $Z$ is of bounded error effects and $F$ is a support of $Z$. ■

4. SID ERROR TYPES AND INTERPRETATIONS

This section introduces the notion of SID error type. Error types are syntactic objects and used to construct channel expressions. The notion of SID interpretation is also introduced here. It is the main tool for giving meaning to error types and channel expressions. We begin with the definition of SID error types. The type $e$ is intended to refer to noiseless channels.

Definition 4.1. An SID error type has one of the following forms: it is a symbol in $\{e, \sigma, \iota, \delta\}$; it is an expression $(\tau_1 \odot \tau_2)$, where $\tau_1$ and $\tau_2$ are SID error types. The set of SID error types is denoted by $\mathcal{E}_0$. 
We note that Jürgensen and Konstantinidis (1996a) also permits error types of the form $r_1 \oplus r_2$. It can be shown, however, that for each channel expression containing error types of the form $r_1 \oplus r_2$ there is an equivalent (in the sense of Section 5) channel expression that involves only error types of Definition 4.1.

**Definition 4.2.** The binary relation $\prec$, between SID error types is defined as follows:

1. $\mu \prec \mu$, for all $\mu \in \mathfrak{T}_0$.
2. If $\mu \prec \tau$, then $\mu \prec (\tau \ominus \tau')$ and $\mu \prec (\tau' \ominus \tau)$, for all $\mu, \tau, \tau' \in \mathfrak{T}_0$.
3. If $\mu_1 \prec \tau$ and $\mu_2 \prec \tau$, then $\mu_1 \cup \mu_2 \prec \tau$, for all $\mu_1, \mu_2, \tau \in \mathfrak{T}_0$.

For example, $\delta \prec \delta, \sigma \prec \delta$, and $\delta \cup \delta \prec \sigma \ominus \delta$.

For given SID error type $\tau$, the set $G_\tau$ is defined as follows:

- $G_\tau = G_\tau$ if and only if $i$ is the only symbol from $\{\epsilon, \sigma, \tau, \delta\}$ that occurs in $\tau$. $G_\delta \subseteq G_\tau$ if and only if $\delta \prec \tau$.
- $G_\sigma \subseteq G_\tau$ if and only if $\sigma \prec \tau$.
- $G_\tau \cup G_\tau$ if and only if $i \prec \tau$.

For example, $G_{\delta \cup \delta} = G_\delta \cup G_\delta$ and $G_{\sigma \cup \delta} = G_\sigma \cup G_\delta$.

An interpretation of a given SID error type $\tau$ should express certain restrictions on how the errors contained in $G_\tau$ interact with each other. Such an interpretation is realized as a set of bounded error effects. Thus, to any SID error type $\tau$ we may associate a possible set $Z(\tau)$ in $\mathcal{S}_{BEE}$. This set is defined such that the intended meaning of the operation $\ominus$ is served. That is, $Z(\tau_1 \ominus \tau_2)$ should restrict the space $\mathcal{H}$ in such a way as to indicate some dependence between the errors of $G_{\tau_1}$ and $G_{\tau_2}$.

**Definition 4.3.** An SID interpretation is a mapping $Z: \mathfrak{T}_0 \rightarrow \mathcal{S}_{BEE}$ satisfying the following conditions:

1. $G_\tau \subseteq G_{\tau'}$ implies that $Z(\tau') \subseteq Z(\tau)$, for all $\tau$ and $\tau' \in \mathfrak{T}_0$.
2. $Z(\epsilon) = \mathcal{H}$.

The first condition expresses dependence among the errors involved with SID error types in the following sense: The error dependences that exist in $Z(\tau)$ are strengthened when $\tau$ is multiplied with $\delta$. As a result, the error type $\tau' = \tau \ominus \delta$ restricts the space $\mathcal{H}$ at least as much as $\tau$ does, namely $Z(\tau') \subseteq Z(\tau)$. On the other hand, as $G_{\delta}$ contains no SID errors, there are no restrictions on $\mathcal{H}$ imposed by $\delta$, that is, $Z(\delta) = \mathcal{H}$. Examples of SID interpretations include those with scattered errors and those with bursts of errors. Both of these are defined next.

First, for a given error function $g$, we define the quantity $N(\tau, g)$, which represents the total number of errors that exist in $g$ with respect to the SID error type $\tau$ and, then, the set $\mathcal{U}(\tau, g)$, which represents all the error bursts that occur in $g$ with respect to $\tau$. For all $\tau \in \mathfrak{T}_0$ and $g \in \mathcal{H}$ we have

$$N(\tau, g) = |g|_{G_{\tau} \cap G_\delta} + |g|_{G_{\tau} \cap G_\sigma} + \sum_{i \in G_{\tau}} |u||g|_{i_k}.$$

For example, for $g = \text{eede}$, we have $N(\delta, g) = 1, N(\epsilon, g) = 2, N(\sigma, g) = 1$, and $N(\tau \ominus \delta, g) = 3$.

Now, for $\tau \in \mathfrak{T}_0$ with $G_{\tau} \neq G_\tau$ and for $g \in \mathcal{H}$ we have

$$\mathcal{U}(\tau, g) = \left\{ f \in \text{Inf}(g) \mid (f(0) \in G_{\tau} \vee f(1) \in G_{\tau}) \wedge (f[f] - 2) \in G_{\tau} \vee f[f] - 1 \in G_{\tau} \right\}.$$  

On the other hand, when $G_{\tau} = G_\tau$, we set $\mathcal{U}(\epsilon, g) = \emptyset$ for all error functions $g$. For example, for $g = \text{eede} \cup \text{eede}$, we have $\mathcal{U}(\delta \ominus \delta, g) = \{\text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede}, \text{eede} \cup \text{eede} \}$.

**Remark 4.4.** Let $\tau$ and $\epsilon'$ be SID error types and let $g$ be an error function.

1. If $G_{\tau} \subseteq G_\tau$, then $N(\tau, g) \leq N(\tau', g)$ and $\mathcal{U}(\tau, g) \subseteq \mathcal{U}(\tau', g)$.
2. If $f \in \text{Inf}(g)$, then $N(\tau, f) \leq N(\tau, g)$ and $\mathcal{U}(\tau, f) \subseteq \mathcal{U}(\tau, g)$.
3. If $\tau = \epsilon$ or $g \in \text{ee}^*$, then $N(\tau, g) = 0$ and $\mathcal{U}(\tau, g) = \emptyset$. 


Let $m$ and $L$ be in $\mathbb{N}_0$ with $m < L$. Then the mapping $S_{(m,L)}$ of $\mathcal{Z}_0$ into $2^\mathcal{H}$ such that

$$S_{(m,L)}(\tau) = \left\{ h \in \mathcal{H} \mid \forall g \in \operatorname{Inf}^\mathcal{H}(h) \cdot |g| \leq 2L + 1 \rightarrow \mathcal{N}(\tau, g) \leq m \right\}$$

is an SID interpretation. The set $S_{(m,L)}(\tau)$ consists of all the error functions $h$ in which at most $m$ errors of type $\tau$ are permitted in any $2L + 1$ consecutive basic error function symbols; hence, in any $L$ consecutive symbols of a word over $X$ to which $h$ can be applied. The mapping $B_{(m,L)}$ of $\mathcal{Z}_0$ into $2^\mathcal{H}$ such that

$$B_{(m,L)}(\tau) = \left\{ h \in S_{(m,L)}(\tau) \mid \forall g \in \operatorname{Inf}^\mathcal{H}(h) \cdot |g| \leq 2L + 1 \rightarrow \forall f \in \mathcal{U}(\tau, g) \cdot |f| \leq 2m + 1 \right\}$$

defines sets of error functions $h$ in which in any $2L + 1$ consecutive basic error function symbols a burst of type $\tau$ cannot be longer than $2m + 1$; hence, in any $L$ consecutive symbols of a word over $X$ to which $h$ can be applied, no two erroneous symbols lie more than $m$ symbols apart. The next result shows that the sets $S_{(m,L)}(\tau)$ and $B_{(m,L)}(\tau)$ are of bounded error effects.

**Proposition 4.5.** For every SID error type $\tau$ the sets $S_{(m,L)}(\tau)$ and $B_{(m,L)}(\tau)$, where $m, L \in \mathbb{N}_0$ with $m < L$, are of bounded error effects supported, respectively, by

$$F_{m,L}(\tau) = \{ g \in \mathcal{H} \mid |g| \leq 2L + 1, \mathcal{N}(\tau, g) \leq m \}$$

and

$$D_{m,L}(\tau) = \{ g \in F_{m,L}(\tau) \mid \forall f \in \mathcal{U}(\tau, g) \cdot |f| \leq 2m + 1 \}.$$

**Proof.** We show the case of $B_{(m,L)}(\tau)$. That is, we assume that $S_{(m,L)}(\tau)$ is supported by $F_{m,L}(\tau)$ and we show that the sets $B_{(m,L)}(\tau)$ and $D_{m,L}(\tau)$ satisfy the three conditions of Definition 3.1. For condition (i), let $h \in B_{(m,L)}(\tau)$ and $g \in \operatorname{Inf}^\mathcal{H}(h)$ with $|g| \leq 2L + 1$. Then $h \in S_{(m,L)}(\tau)$ and, therefore, $g \in F_{m,L}(\tau)$. Moreover, as $|f| \leq 2m + 1$ for $f \in \mathcal{U}(\tau, g)$, one has $g \in D_{m,L}(\tau)$. Similarly, one verifies that the converse holds true. For condition (ii), let $g \in D_{m,L}(\tau)$ and let $h = e^{2i+1} \cdot e^{2j+1}$ with $|h| \leq 2L + 1$ for some $i, j \in \mathbb{N}_0$. As $g$ is in $F_{m,L}(\tau)$, $h$ is in $F_{m,L}(\tau)$ too. Also, it follows that $\mathcal{U}(\tau, h) = \mathcal{U}(\tau, g)$, which implies that $|f| \leq 2m + 1$ for all $f \in \mathcal{U}(\tau, h)$. Hence, $h \in D_{m,L}(\tau)$. Finally, condition (iii) is satisfied since $(GD)^{i}G \cap D_{m,L}(\tau) \subseteq (GD)^{i}G \cap F_{m,L}(\tau) = \emptyset$. $lacksquare$

Using Remark 4.4, we verify that the mappings $S_{(m,L)}$ and $B_{(m,L)}$ are SID interpretations for all $m, L \in \mathbb{N}_0$ with $m < L$. We refer to $S_{(m,L)}$ as an SID interpretation with scattered errors and to $B_{(m,L)}$ as an SID interpretation with bursts of errors.

5. SID CHANNEL EXPRESSIONS AND SID CHANNELS

In this section we define formal expressions for denoting channels. Intuitively, an SID channel expression describes the type of errors permitted and the dependences and frequencies of these errors. For example, $(3,12)$ is an SID channel expression denoting the channel that permits at most three deletions of symbols in any 12 or less consecutive symbols of a transmitted message. In that expression, the pair $(3,12)$ is a parameter that indicates how deletion errors are combined. In general, in defining SID channel expressions, we consider a set $P$ of parameters with the understanding that the elements of $P$ are formal (syntactic) objects. Examples of parameter sets are $P_{\text{scatt}} = \{(m, L) \mid m, L \in \mathbb{N}_0, m < L \}$ and $P_{\text{burst}} = \{(m, L) \mid (m, L) \in P_{\text{scatt}} \}$. Moreover, we consider families of SID interpretations indexed by parameter sets. We write $\mathcal{Z} = \{Z_p \mid p \in P\}$ to denote a family of SID interpretations $Z_p$, for all parameters $p$ in $P$. In this case, we use $P_Z$ to refer to the parameter set of the family $\mathcal{Z}$. Of particular interest are the family $\mathcal{S} = \{S_p \mid p \in P_{\text{scatt}}\}$ of SID interpretations with scattered errors, and the family $\mathcal{B} = \{B_p \mid p \in P_{\text{burst}}\}$ of SID interpretations with bursts of errors. Obviously, $P_{\mathcal{S}} = P_{\text{scatt}}$ and $P_{\mathcal{B}} = P_{\text{burst}}$. 

Definition 5.1. Let $Z$ be a family of SID interpretations. The set $E_Z$ of channel expressions with respect to $Z$ is defined as follows:

(i) $\tau p \in E_Z$, for all error types $\tau$ and parameters $p$ in $P_Z$.

(ii) If $\xi_1$ and $\xi_2$ are in $E_Z$, then $\xi_1 \oplus \xi_2$ is in $E_Z$.

It follows that a channel expression with respect to $Z$ is a string of the form

$$\tau_0 p_0 \oplus \cdots \oplus \tau_n p_n$$

for some $n \in \mathbb{N}$, where $\tau_i$ is any SID error type and $p_i$ is any parameter of $Z$, for $i \in n$. Moreover, for brevity, we adopt the notation $\sum_{i \in n} \tau_i p_i$ for SID channel expressions.

Consider the families $S$ and $B$. The following are examples of channel expressions with respect to $S$, $B$, and $S \cup B$, respectively:

$$(\sigma \circ \iota)(2, 10) \oplus (\sigma \circ \delta)(3, 15), \quad \sigma[5, 20] \oplus (\iota \circ \delta)[3, 11], \quad \sigma[5, 20] \oplus (\iota \circ \delta)(3, 11).$$

In contrast to the operation $\odot$ between error types, the operation $\oplus$ between channel expressions $\xi_1$ and $\xi_2$ indicates that the restrictions on errors are applied separately for $\xi_1$ and $\xi_2$. Let $Z$ be a family of SID interpretations. To each channel expression $\xi = \sum_{i \in n} \tau_i p_i$ in $E_Z$, we associate the set

$$G_\xi = G_\tau \cup G_{\tau_1} \cup \cdots \cup G_{\tau_n},$$

which consists of $e$ and all the basic error function symbols implied by $\tau_0, \ldots, \tau_n$, and the set

$$Z(\xi) = \bigcap_{i \in n} Z_{p_i}(\tau_i) \cap G^*_\xi,$$

which consists of all the error functions in $G^*_\xi$ that conform to the interpretation $Z_{p_i}$ of $\tau_i$, for all $i \in n$.

By Theorem 3.7, the set $Z(\xi)$ is of bounded error effects.

Definition 5.2. An SID channel is a BEE channel $c(\xi)$ such that

$$c(\xi) = \{(y' | y) | y' \in X^*, \exists h \in Z(\xi): y' = h(y)\},$$

where $Z$ is a family of SID interpretations and $\xi$ is a channel expression with respect to $Z$.

For example, if $\xi = \sigma[4, 15], c(\xi)$ is the SID channel that permits a burst of substitutions of length at most 4 in any 15 (or less) consecutive symbols of a message. Similarly, the channel expression $\sigma[5, 20] \oplus (\iota \circ \delta)(3, 11)$ denotes the SID channel that permits a total of at most 3 scattered insertions and deletions in any 11 (or less) consecutive symbols of a message and, at the same time, a burst of substitutions of length at most 5 in any 20 (or less) consecutive symbols of the message.

Examples from previous literature include the SID channels $\sigma(m, L), \iota(m, L), \delta(m, L), (\iota \circ \delta)(m, L)$, and $\sigma(\iota \circ \delta)(m, L)$ considered in Levenshtein (1966b, 1992) and Peterson and Weldon (1972), with main results concerning block codes capable of correcting the errors of such channels. The channel $\iota \circ \delta)(1, L)$ has also been considered in Sellers (1962), Ullman (1966), and Teneng’ol’ts (1984), and the channel $\delta(2, L)$ in Levenshtein (1970). Moreover, variable-length codes are constructed in Jürgensen and Konstantinidis (1995) for the channel $\delta(1, L)$. Finally, in Hollman (1993), conditions on codes are considered for correcting certain error situations that correspond to SID channels of the form $\delta(m_1, L) \oplus (\iota(m_2, L)$.

For two SID channel expressions $\xi_1$ and $\xi_2$ it is possible to have $Z(\xi_1) = Z(\xi_2)$. In this case, we say that $\xi_1$ and $\xi_2$ are equivalent. For example, $\delta \circ \delta(3, 5)$ is equivalent to $\delta(3, 5)$, and $\delta(1, 22) \oplus (\iota \circ \delta)(1, 22)$ is equivalent to $\iota \circ \delta(1, 22)$. In the rest of the section we discuss the possibility of converting a given SID channel expression to an equivalent but simpler one. We consider the congruence “$\Xi$” on $E_0$ defined
by the equations

\[(E1) \tau_1 \odot \tau_2 = \tau_2 \odot \tau_1 \quad (E2) \tau_1 \odot (\tau_2 \odot \tau_3) = (\tau_1 \odot \tau_2) \odot \tau_3 \]
\[(E3) \tau \odot \tau = \tau \quad (E4) \tau \odot \varepsilon = \tau,\]

where \(\tau, \tau_1, \tau_2, \tau_3 \in \mathbb{Z}_0\). That is, two error types \(\tau_1\) and \(\tau_2\) are congruent, \(\tau_1 \equiv \tau_2\), if \(\tau_1\) can be transformed to \(\tau_2\) using Eqs. (E1)–(E4) a finite number of times. In particular, we write \(\tau \vdash \tau'\) to indicate that \(\tau\) can be transformed to \(\tau'\) in one step, using one of the equations. For example,

\[(\sigma \odot \varepsilon) \odot \iota \vdash \sigma \odot (\varepsilon \odot \iota) \vdash \sigma \odot (\iota \odot \varepsilon) \vdash (\sigma \odot \iota) \odot \varepsilon \vdash \sigma \odot \iota.\]

**Lemma 5.3.** If \(\tau_1\) and \(\tau_2\) are congruent error types, then \(G_{\tau_1} = G_{\tau_2}\) and \(Z(\tau_1) = Z(\tau_2)\) for every SID interpretation \(Z\).

**Proof.** Let \(\rho_1 = \rho_2\) be any of the Eqs. (E1)–(E4). Then \(G_{\rho_1} = G_{\rho_2}\). Hence, as \(\tau_1\) can be transformed to \(\tau_2\) using only those equations, it follows that \(G_{\tau_1} = G_{\tau_2}\). Moreover, Definition 4.3 implies that \(Z(\tau_1) = Z(\tau_2)\) for any SID interpretation \(Z\). \(\blacksquare\)

**Proposition 5.4.** Every error type is (effectively) congruent to one and only one of the following error types:

1. \(\varepsilon\)
2. \(\sigma\)
3. \(\iota\)
4. \(\delta\)
5. \(\sigma \odot \iota\)
6. \(\sigma \odot \delta\)
7. \(\iota \odot \delta\)
8. \(\sigma \odot \iota \odot \delta\).

**Proof.** One can verify that every error type can be converted to one of the above eight error types by applying Eqs. (E1)–(E4). On the other hand, it is easy to verify that if \(\tau_1\) and \(\tau_2\) are two different error types from the list of eight, then \(G_{\tau_1} \neq G_{\tau_2}\). Hence, \(\tau_1 \neq \tau_2\). \(\blacksquare\)

The eight error types listed in Proposition 5.4 are said to be in normal form.

**Proposition 5.5.** Let \(Z\) be a family of SID interpretations and let \(\xi\) be an SID channel expression in \(\mathcal{E}_Z\).

1. If the expression \(\xi'\) results by replacing every error type in \(\xi\) with its normal form, then \(Z(\xi) = Z(\xi')\).
2. If \(\xi\) contains the terms \(\tau p\) and \(\tau' p\) with \(\tau \vdash \tau'\), then \(Z(\xi) = Z(\xi')\), where \(\xi'\) is the expression that results by removing the term \(\tau p\) from \(\xi\).

**Proof.** Part (i) follows from the definition of \(Z(\xi)\) and Lemma 5.3. For part (ii), we note that \(\tau \vdash \tau'\) implies that \(G_\tau \subseteq G_{\tau'}\) and, therefore, \(Z_p(\tau') \subseteq Z_p(\tau)\). Hence, as \(Z(\xi) \subseteq Z_p(\tau')\), omitting \(\tau p\) from \(\xi\) does not alter \(Z(\xi)\). \(\blacksquare\)

Consider the channel expression \(\xi_1 = \sigma[2, 19] \oplus (\delta \odot \varepsilon)(2, 30) \oplus (\iota \odot \delta)(2, 30)\). By the first part of Proposition 5.5, \(\xi_1\) is equivalent to \(\xi_2 = \sigma[2, 19] \oplus (\delta \odot \varepsilon)(2, 30) \oplus (\iota \odot \delta)(2, 30)\). Moreover, as \(\delta \vdash \iota \odot \delta\), the second part of Proposition 5.5 implies that \(\xi_2\) is equivalent to \(\xi_3 = \sigma[2, 19] \oplus (\iota \odot \delta)(2, 30)\). Further simplifications are possible for specific families of SID interpretations. For example, we can show that if the channel expression \(\xi\) contains the terms \(\tau(m_1, L_1)\) and \(\tau(m_2, L_2)\) such that \(L_1 \leq L_2\) and \(m_1 \geq m_2\), then \(\xi\) is equivalent to the expression that results by removing the term \(\tau(m_1, L_1)\) from \(\xi\).

6. UNIQUE DECODABILITY FOR SID CHANNELS

In this section we consider the notion of unique decodability in the presence of errors from SID channels. We show that for certain families of SID interpretations, including the families with scattered errors and with bursts of errors, it is decidable whether a given finite code is uniquely decodable for the channel described by a given channel expression.

**Definition 6.1.** A family \(Z\) of SID interpretations is said to be finitely representable, if there is an algorithm that computes, for given parameter \(p \in P_Z\) and error type \(\tau\), a finite support of the SID interpretation \(Z_p(\tau)\) of \(\tau\).
**Theorem 6.2.** Let $Z$ be a family of SID interpretations that is finitely representable. Then the following problem is decidable:

**Instance:** A channel expression $\xi$ in $\mathfrak{E}_Z$ and a finite code $K$.

**Question:** Is $K$ uniquely decodable for $c(\xi)$?

**Proof.** The required algorithm consists of the following steps. Let $\xi = \sum_{i \in \mathbb{N}} \tau_i p_i$.

(i) If $K$ is not uniquely decodable, output NO and quit.

(ii) For each $i \in \mathbb{N}$, find a support $F_{\tau_i}(\xi)$ of $Z_{\tau_i}(\xi)$.

(iii) Let $F$ be a support of $Z(\xi)$.

(iv) Decide whether $K$ is uniquely decodable for $c(\xi)$, using the code $K$ and the representation $F$ of $c(\xi)$.

Step (i) can be computed—see, for instance, Markov (1962) or Berstel and Perrin (1985). For step (ii), we use the fact that $Z$ is finitely representable. Step (iii) is also computable by Theorem 3.7. Finally, in step (iv), Theorem 3.6 is used. □

The family $S \cup B$ is finitely representable; hence also the families $S$ and $B$. Indeed, given a parameter $p$ in $P_{\text{scan}} \cup P_{\text{burst}}$ and an error type $\tau$, we compute the set

$$F_p(\tau) = \{ g \in \mathcal{H} \mid |g| \leq 2L + 1, N(\tau, g) \leq m \},$$

where $m$ and $L$ are such that $p = (m, L)$ or $p = [m, L]$. If $p = (m, L)$ then, by Proposition 4.5, $F_p(\tau)$ is a support of $S_p(\tau)$. On the other hand, if $p = [m, L]$ then, again, by Proposition 4.5 the set

$$D_p(\tau) = \{ g \in F_p(\tau) \mid \forall f \in \mathcal{U}_e(g): |f| \leq 2m + 1 \}$$

is a support of $B_p(\tau)$.

**Corollary 6.3.** The following problem is decidable:

**Instance:** A channel expression $\xi$ with respect to $S \cup B$ and a finite code $K$.

**Question:** Is $K$ uniquely decodable for $c(\xi)$?

7. **Discussion**

In the past, traditional models of communication channels were based on the assumption that the cost of synchronization errors is very low. As a consequence, efforts to cope with such errors in a systematic manner were limited, and the resulting models were concerned mainly with substitution errors. Recently, however, there has been a renewed interest in the study of synchronization errors, as a result of the increasing demand for fast and reliable transmissions. In this paper, we have presented a general model of discrete channels that allow one to express various error situations in a systematic manner. The operations on error types can be used to generate channel expressions of arbitrary complexity and the corresponding SID channels include many of the channels considered in isolated studies of the past.

The material presented here complements the work in Konstantinidis (1999) where several new tools for studying the structure of uniquely decodable codes for BEE channels and, therefore, for SID channels have been developed. The channels in Konstantinidis (1999), however, have more abstract representations as opposed to the simpler expressions used for describing SID channels. As a result, using the tools of the present paper, one is able to express in a meaningful manner statements of the form “the code $K$ can correct all errors of type $\xi$,” and “the code $K$ is uniquely decodable for the channel described by the expression $\xi$.”

**References**


