Abstract. This paper deals with the satisfiability of requirements put on the identifiability of unions of language families. We consider identification in the limit from a text with bounds on mindchanges and anomalies. We show that, though these identification types are not closed under the set union, some of them still have features that resemble closedness. To formalize this, we generalize the notion of closedness. Then by establishing "how closed" these identification types are we solve the satisfiability problem.

1 Introduction

In this paper a problem in inductive inference of recursively enumerable languages is considered. Inductive inference as a term used for finding out an algorithm from sample computations was first used by E. M. Gold in [3]. Since then many identification types have been introduced.

E. M. Gold in [3] proved that there are two families of languages that are identifiable in the limit, while their union is not. Similar results were obtained for other identification types. However, a further research shows that these non-union theorems do not convey in full the structure of these identification types in terms of unions of language families. Though not closed under union, these types are not absolutely "open." For instance, if every union of three families out of $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$ is identifiable in the limit, so is the union $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3 \cup \mathcal{L}_4$. So not all the requirements put on the identifiability of the unions of families are satisfiable. In the previous example, the requirement that $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \mathcal{L}_3 \cup \mathcal{L}_4$ is not identifiable would be unsatisfiable.

The main intention of this paper is to uncover the closedness properties and to find out which sets of requirements are satisfiable and which are not for the identification in the limit from a text with bounds on mindchanges and anomalies. (For the identification of recursive functions, this problem was investigated...
in [1, 8]. We deal with this problem by introducing a natural generalization of the notion of closedness. The identification type is called \( n \)-closed in case for every set of \( n \) families of languages: if all the unions of \( n-1 \) of these families are identifiable, so is the union of all \( n \) families. It turns out that this notion is sufficient for establishing if the requirements are satisfiable. Interestingly, the \( n \)-closedness is related to team learning, a field that is intensively investigated at the moment.

We proceed as follows. Section 2 contains preliminaries. In Section 3 we define \( n \)-closedness and establish some properties of it. In Section 4 we reduce the problem of satisfiability of requirements to finding out, for which \( n \) the identification type is \( n \)-closed. In Section 6 we find these \( n \) for identification in the limit with bounds on mindchanges and anomalies. Before that, in Section 5 we point to the connection of \( n \)-closedness with team learning. Section 7 summarizes the results.

2 Preliminaries

2.1 Notation

Any recursion theoretic notation not explained below is from [7]. \( \mathbb{N} \) denotes the set of natural numbers, \( \{0, 1, 2, \ldots \} \). \( * \) denotes “an arbitrary finite (natural) number.” In inequalities \( (\forall n \in \mathbb{N})[n < * < \infty] \), \( \forall^\infty \) means “for all but finitely many,” \( \exists^\infty \) means “there exist infinitely many.” When applied to sets, \( \subseteq \) denotes proper subset. \( \langle x_1, x_2, \ldots, x_n \rangle \) denotes a Cantor number of \( \langle x_1, x_2, \ldots, x_n \rangle \).

We fix a Gödel numbering of the partial recursive functions of one argument and denote it by \( \varphi \). The function computed by the program \( i \) we denote by \( \varphi_i \). Its domain \( \mathcal{W}_i \) is the recursively enumerable language accepted by \( \varphi_i \). Letter \( \mathcal{E} \) denotes the set of recursively enumerable languages. If a function \( f \) is undefined at point \( x \), we write \( f(x) \uparrow \); otherwise we write \( f(x) \downarrow \).

For \( L_1, L_2 \in \mathcal{E}, a \in \mathbb{N} \cup \{\#\} \), by \( L_1 =^a L_2 \) we mean that \( \text{card}((L_1 - L_2) \cup (L_2 - L_1)) \leq a \). These \( a \) differences between the languages are called anomalies.

We will consider finite and infinite sequences with values from \( \mathbb{N} \cup \{\#\} \), where \( \# \) means “no data.” The length of a finite sequence \( \sigma \) is denoted by \( |\sigma| \). For a sequence \( \sigma \), the initial sequence of length \( n \) \( (n \leq |\sigma| \) if \( \sigma \) is finite) is denoted by \( \sigma[n] \). The content of a sequence \( \sigma \) is the set of natural numbers in the range of \( \sigma \), denoted \( \text{content}(\sigma) \). An infinite sequence \( T \) is a text for a language \( L \) iff \( \text{content}(T) = L \). \( \sigma \subseteq \tau \) means that \( \tau \) is an extension of \( \sigma \); \( \sigma \subset \tau \) means that \( \tau \) is a proper extension of \( \sigma \).

2.2 Identification

An identification strategy \( F \) is an algorithm that receives as input a finite sequence and it outputs either a natural number (a hypothesis) or the symbol \( \bot \) in case it has no hypothesis to issue.

The sequence of outputs produced by \( F \), when it receives larger and larger initial segments of a text for a language, should satisfy the requirements of the