Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-18
https://doi.org/10.22049/cco.2026.31006.2705

CCO
Commun. Comb. Optim.

Research Article

Definability of some k-ary relations over second order kinds of logics

Simone Costa[1,†], Marco Dalai[2,‡], Stefano Della Fiore[2,$], Anita Pasotti[1,*]

[1] DICATAM - Sez. Matematica, Università degli Studi di Brescia,
Via Branze 43, I-25123 Brescia, Italy
[†] simone.costa@unibs.it
[*] anita.pasotti@unibs.it

[2] DII, Università degli Studi di Brescia, Via Branze 38, I-25123 Brescia, Italy
[‡] marco.dalai@unibs.it
[$] stefano.dellafiore@unibs.it

Received: 16 September 2025; Accepted: 5 February 2026
Published Online: 23 February 2026

Abstract: We consider the expressibility in monadic second order logic of certain relations of importance in computer science. For integers $n \geq 1$ and $k \leq b$, a k-tuple of sequences in $\{0, 1, \ldots, b-1\}^n$ are said to be k-hashed if there is a coordinate where they all differ. A set $\mathcal{C}$ of sequences is said to be a k-hash code if any k distinct elements are k-hashed. Testing whether a code is k-hashing and determining the largest size of k-hash codes is an important problem in computer science. The use of general purpose solvers for this problem leads to the question of what minimal logic is needed to represent the problem.

In this paper, we prove that the (b, k)-hashing relation on k-tuples is not definable in Monadic Second Order Logic (MSO), highlighting its limitations for this problem. On the other hand, if one extends MSO by adding the possibility of expressing equalities between cardinalities, then the existence of trifferent codes of size m and length n can be specified by a *single sentence*, whose syntactic size does not grow with m, n.

Finally, since to the best of our knowledge there are no available solvers for MSO with such global cardinality constraints, we provide a practical SMT encoding in Z3 for fixed parameters. In addition, we computationally recover the non-existence of a trifferent code of size 11 and length 5.

Keywords: k-hash code, inexpressibility, monadic second order logic

AMS Subject classification:

* *Corresponding Author*

1. Introduction

Let $\mathcal{C}$ be a subset of $\{0, 1, \ldots, b-1\}^n$ with the property that, for any $k \leq b$ distinct elements, there exists a coordinate in which they all differ. A code $\mathcal{C}$ with this property is called a (b, k)-hash code of length n, and plays an important role in computer science. In particular, as n grows, of relevant theoretical interest is the determination of the cardinality $T_{b,k}(n)$ of the largest such (b, k)-hash codes [11, 12]. Upper and lower bounds were derived over the years, showing that the particular case $b = k = 3$, while being the simplest non-trivial instance to formulate, actually represents the most challenging one. A simple polynomial upper bound of the form $T_{3,3}(n) \leq C(3/2)^n$, for some constant C, had remained for years the best known (see [3], [7] or [15] for discussions), and only recently a polynomial improvement of the form $T_{3,3}(n) \leq Cn^{-2/5}(3/2)^n$ was derived in [1]. On the achievabiliy side, the best known lower bound $T_{3,3} \geq C(9/5)^{n/4}$ can be derived by means of an explicit constructions for $n = 4$ ([12]; see also [2] for an alternative method).

In this context, testing for arbitrary n and T whether there exist (b, k)-hash codes of length n and size T represents a relevant question, and explicit algorithms for answering it are of interest ([7], [15]). One naturally derived question is whether general purpose solvers can be used for this task. Methods for proving the existence (or the non-existence) of structures that satisfy given constraints have been recently implemented, as for example in the Satisfiability Modulo Theories (SMT, for short) program MONA [8], where these constraints can be written in the First Order Logic or in the Monadic Second Order Logic (MSO).

Our attempts to use these approaches to improve the results of [7] and [15] lead us to ask ourselves whether the (b, k)-hashing k-ary relation is even expressible in the Monadic Second Order Logic. In this paper, we show that the answer is negative.

In the interest of exposition, we consider explicitly the case $b = k = 3$, the general case being easily derivable from this one. In this special case, three sequences having the $(3, 3)$-hashing property are simply said to be *trifferent*, and we thus present our method by showing that the *trifference* relation for general length n cannot be expressed in MSO. In particular, in Section 3, we will show that there are no formulas $f(X, Y, Z)$ of rank ϱ satisfied if and only if X, Y, Z are trifferent words of length large enough with respect to ϱ. Firstly, in Subsection 3.1, we will provide a proof of the fact that there are no formulas $f(X, Y, Z)$ satisfied whenever X, Y, Z are finite, trifferent words (i.e. the trifference relation is not S3S definable). Then we prove the result more in general for both finite and infinite words.

On the other hand, if we extend MSO by adding the possibility of expressing equalities between cardinalities, it becomes possible to specify trifference and, more generally, (b, k)-hashing constraints at the level of a *single uniform sentence*: the parameters (m, n) (code size and word length) can be supplied by the structure via monadic predicates whose cardinalities encode m and n, while second-order quantification internalizes the "for all triples, there exists a coordinate" pattern. This stands in contrast with standard SMT encodings, which expand such requirements into a family of constraints whose size grows combinatorially with m and linearly with n.

There are, indeed, several well-studied theories on this topic such as the counting MSO (CMSO, in brief), [4, 5], the MSO with cardinalities [10], and extensions which add a weak kind of arithmetics (the so-called Presburger arithmetics), see [9, 13, 14]. However, to the best of our knowledge, there are currently no off-the-shelf solvers implementing an *SbS*-style MSO theory enriched with such global cardinality constraints.

In the last section of this work, we provide a Z3 [6] implementation for fixed parameters (m, n). As a sanity check, we recover the non-existence of a trifferent code of size 11 and length 5, consistently with the exact value $T(5) = 10$ established in [7] (and used as a starting point in [15]). While this does not improve the best known values in the literature, it illustrates a simple and reusable SMT formulation; to the best of our knowledge, this solver-based viewpoint has not been explicitly proposed for trifferent-code feasibility.

2. Preliminars

Monadic Second-Order logic is a widely used and expressive logical framework that balances expressive power with computational tractability. It is particularly well-suited for analyzing finite or countable structures. MSO extends beyond the capabilities of first-order logic by enabling the expression of "mildly recursive" structural properties, such as connectedness and reachability, which are essential for addressing key modeling needs in areas like database theory and knowledge representation.

The *monadic second-order theory* $S3S$ is defined over a specific class of structures representing infinite ternary trees. Formally, the structures considered in $S3S$ are of the form:

$$\mathfrak{T} = (T, \mathtt{r}, \succ_0^{\mathfrak{T}}, \succ_1^{\mathfrak{T}}, \succ_2^{\mathfrak{T}}),$$

where:

- $T = \{0, 1, 2\}^*$ is the set of all finite ternary strings, serving as the *domain* of the structure, and representing the nodes of a complete infinite ternary tree;
- $\mathtt{r}$ is the root of the tree, i.e. the node associated to the empty string;
- $\succ_0^{\mathfrak{T}}$ is a subset of $T \times T$ called the *left successor relation*, defined as:

$$\succ_0^{\mathfrak{T}} = \{(w, w0) \mid w \in \{0, 1, 2\}^*\};$$

- $\succ_1^{\mathfrak{T}}$ is a subset of $T \times T$ called the *middle successor relation*, defined as:

$$\succ_1^{\mathfrak{T}} = \{(w, w1) \mid w \in \{0, 1, 2\}^*\};$$

- $\succ_2^{\mathfrak{T}}$ is a subset of $T \times T$ called the *right successor relation*, defined as:

$$\succ_2^{\mathfrak{T}} = \{(w, w2) \mid w \in \{0, 1, 2\}^*\}.$$

It is straightforward to define, in analogy, the theory SbS over the infinite b-ary tree. Formally, the structures of SbS are of the form

$$\mathfrak{T} = \big(T, \mathtt{r}, \succ_0^{\mathfrak{T}}, \succ_1^{\mathfrak{T}}, \ldots, \succ_{b-1}^{\mathfrak{T}}\big),$$

where:

- $T = \{0, 1, \ldots, b-1\}^*$ is the set of all finite b-ary strings, serving as the *domain* of the structure and representing the nodes of the complete infinite b-ary tree;
- $\mathtt{r}$ and $\succ_i^{\mathfrak{T}}$ are defined exactly as in the case of $S3S$.

Language and Syntax. The language of $S3S$ (resp. SbS) is a monadic second-order logic with vocabulary $\mathbb{S} = \{\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathtt{c}_n\} \cup \{\succ_0, \succ_1, \succ_2\}$ (resp. $\mathbb{S} = \{\mathtt{c}_1, \mathtt{c}_2, \ldots, \mathtt{c}_n\} \cup \{\succ_0, \succ_1, \ldots, \succ_{b-1}\}$). It includes:

- A set of *constants* denoted by $\mathtt{c}_1, \ldots, \mathtt{c}_n$.
- *Individual variables* (e.g., x_1, x_2, x_3) that range over elements of the domain T.
- *Set variables* (e.g., X_1, X_2, X_3) that range over subsets of T.
- Logical connectives: $\wedge, \vee, \neg, \implies, \iff$.
- Quantifiers:
  - First-order quantifiers ($\forall x$, $\exists x$) over individual variables.
  - Second-order quantifiers ($\forall X$, $\exists X$) over set variables.
- Atomic formulas:
  - $x \succ_0 y$, $x \succ_1 y$ and $x \succ_2 y$ (resp. $x \succ_0 y$, $x \succ_1 y, \ldots,$ and $x \succ_{b-1} y$), representing the left, middle and right successor relations.
  - $x \in X$, indicating membership of an element x in a subset $X \subseteq T$.
  - $x = y$, expressing equality between individual elements.

In the following, we will also use the term "infinite words" to denote an infinite path of the ternary tree, starting from the root $\mathtt{r}$.

3. MSO inexpressibility of the hashing property

In this section we first provide a simple proof that the property that three finite words are trifferent is not expressible in MSO, see Subsection 3.1. Then we prove, more in general, that any hashing property is not expressible in MSO for both finite and infinite words, see Subsections 3.2 and 3.3.

3.1. *MSO* inexpressibility of the trifference: finite words

In this paragraph, we will prove, by direct proof, the nonexistence of formulas $f(X,Y,Z)$ satisfied whenever X,Y,Z are finite, trifferent words.
First of all, we note that we can identify the set of finite ternary strings T with the subset of binary strings $T' = \{00,01,10\}^* \subseteq B = \{0,1\}^*$ and the binary relations on T, $\succ_0^{\mathfrak{T}}, \succ_1^{\mathfrak{T}}, \succ_2^{\mathfrak{T}}$ with the corresponding binary relations on T', $\succ_0^{\mathfrak{T}'}, \succ_1^{\mathfrak{T}'}, \succ_2^{\mathfrak{T}'}$. Here we have essentially substitute the 0s with 00s, the 1s with 01s and the 2s with 10s. Since the set T' and the relations $\succ_0^{\mathfrak{T}'}, \succ_1^{\mathfrak{T}'}, \succ_2^{\mathfrak{T}'}$ are $S2S$ definable, if we assume, by contradiction, that the trifference relation would be $S3S$ definable, then we would obtain the existence of a formula $f'(\cdot,\cdot,\cdot)$ such that $f'(X,Y,Z)$ is true if and only if the binary finite words X,Y,Z are in T' and there exists a coordinate i, where i is an odd positive integer, such that $\{X_iX_{i+1}, Y_iY_{i+1}, Z_iZ_{i+1}\} = \{00,01,10\}$.
We now consider the relation $R(X,Y,Z)$ on binary words that is satisfied whenever $f'(X,Y,Z)$ is true.

Lemma 1. *The relation R is not S2S definable.*

Proof. Suppose, by contradiction, that the relation R is $S2S$ definable. Then, according to Theorem $1(iv)$ of [16], we have that, being R a relation among finite words, it is a finite union of special relations. More precisely, we have

$$R = \left(\bigcup_{i\in I_1} R_i^1\right) \cup \left(\bigcup_{i\in I_2} R_i^2\right) \cup \left(\bigcup_{i\in I_3} R_i^3\right)$$

where I_1, I_2 and I_3 are finite sets and each special relation R_i^1 is of the form

$$R_i^1 = \{(a_1a_2, a_1a_3a_4, a_1a_3a_5) : a_1 \in A_i^1, a_2 \in A_i^2, a_3 \in A_i^3, a_4 \in A_i^4, a_5 \in A_i^5\},$$

each R_i^2 is of the form

$$R_i^2 = \{(a_1a_3a_4, a_1a_2, a_1a_3a_5) : a_1 \in A_i^1, a_2 \in A_i^2, a_3 \in A_i^3, a_4 \in A_i^4, a_5 \in A_i^5\},$$

and each R_i^3 is of the form

$$R_i^3 = \{a_1a_3a_4, a_1a_3a_5, a_1a_2) : a_1 \in A_i^1, a_2 \in A_i^2, a_3 \in A_i^3, a_4 \in A_i^4, a_5 \in A_i^5\}$$

where the sets $A_i^1, \ldots, A_i^5$ are regular sets. The definition of regular set is not important for this proof, but the reader can find it in [16].
Given $\ell > n \geq 0$, we consider the words $X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}$ defined as

$$X^{n,\ell} := (00)^n(01)(00)^{\ell-n-1},$$

$$Y^{n,\ell} := (10)^n(00)(00)^{\ell-n-1},$$

$$Z^{n,\ell} := (10)^n(10)(00)^{\ell-n-1}.$$

We have that $R(X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell})$ is satisfied since, set $i = 2n+1$, i is an odd positive integer and

$$\{X_i^{n,\ell}X_{i+1}^{n,\ell}, Y_i^{n,\ell}Y_{i+1}^{n,\ell}, Z_i^{n,\ell}Z_{i+1}^{n,\ell}\} = \{00, 01, 10\}.$$

This means that any triple $X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}$ satisfies at least one special relation of type R_i^j where $j \in \{1,2,3\}$. Therefore, fixed $\ell > |I_1| + |I_2| + |I_3|$, for the pigeonhole principle, there exists two triples $X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}$ and $X^{n',\ell}, Y^{n',\ell}, Z^{n',\ell}$ that satisfy the same relation R_i^j. We split the proof into two cases.
Case 1: $X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}$ and $X^{n',\ell}, Y^{n',\ell}, Z^{n',\ell}$ both satisfy R_i^1. This means that

$$(X^{n,\ell}, Y^{n,\ell}, Z^{n,l\ell}) = (a_1a_2, a_1a_3a_4, a_1a_3a_5)$$

and

$$(X^{n',\ell}, Y^{n',\ell}, Z^{n',\ell}) = (a_1'a_2', a_1'a_3'a_4', a_1'a_3'a_5').$$

Since $X^{n,\ell}$ and $Y^{n,\ell}$ begin differently, we have that a_1 is the emptyword, denoted here by $\underline{0}$, and

$$(X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a_2, a_3a_4, a_3a_5).$$

Similarly, we also have that $a_1' = \underline{0}$. Thus $X^{n,\ell} = a_2$ and $X^{n',\ell} = a_2'$ for some a_2, a_2' in the same regular set A_i^2. But this would also imply that

$$(X^{n',\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a_2', a_3a_4, a_3a_5)$$

satisfies the relation R_i^1 which is a contradiction since $(X^{n',\ell}, Y^{n,\ell}, Z^{n,\ell})$ does not satisfy R.
Case 2: $X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}$ and $X^{n',\ell}, Y^{n',\ell}, Z^{n',\ell}$ both satisfy R_i^2 (or, simmetrically, R_i^3). This means that

$$(X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a_1a_3a_4, a_1a_2, a_1a_3a_5)$$

and

$$(X^{n',\ell}, Y^{n',\ell}, Z^{n',\ell}) = (a_1'a_3'a_4', a_1'a_2', a_1'a_3'a_5').$$

Since $X^{n,\ell}$ and $Y^{n,\ell}$ begin differently, we have that $a_1 = \underline{0}$ and

$$(X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a_3a_4, a_2, a_3a_5).$$

$X^{n,\ell}$ and $Z^{n,\ell}$ begin differently, we have that $a_3 = \underline{0}$ and

$$(X^{n,\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a_4, a_2, a_5).$$

Similarly, we also have that $a'_1 = a'_3 = \underline{0}$. Thus $X^{n,\ell} = a_4$ and $X^{n',\ell} = a'_4$ for some a_4, a'_4 in the same regular set A_i^4. But this would also imply that

$$(X^{n',\ell}, Y^{n,\ell}, Z^{n,\ell}) = (a'_4, a_2, a_5)$$

satisfies the relation R_i^2 which is a contradiction since $(X^{n',\ell}, Y^{n,\ell}, Z^{n,\ell})$ does not satisfy R. □

From Lemma 1, it follows immediately that:

Theorem 1. *There is no S3S formula $f(\cdot,\cdot,\cdot)$ such that $f(X,Y,Z)$ is true if and only if the finite words X, Y, Z are trifferent.*

3.2. Ehrenfeucht–Fraïssé game

In this subsection we introduce the Ehrenfeucht–Fraïssé (EF, for short), a combinatorial game used to explore the expressive power of logical languages by comparing structures, which will play a crucial role in the next subsection. The EF game allows us to formally analyze which properties can or cannot be distinguished within a given logic, such as first-order or monadic second-order logic. Here, we define and analyze the EF game specifically for MSO $S3S$ logic.

The aim of the EF game on MSO $S3S$ logic is to establish whether these structures satisfy the same logical formulas up to a certain quantifier depth within this logic. We will denote this quantifier depth by ϱ.

Given two tree structures $\mathfrak{A}$ and $\mathfrak{D}$ defined on the same domain T, two ℓ-tuples $\underline{a} = (a_1, a_2, \ldots, a_\ell) \in T^\ell$ and $\underline{d} = (d_1, d_2, \ldots, d_\ell) \in T^\ell$, and, finally, two p-tuples $\underline{V} = (V_1, V_2, \ldots, V_p)$ and $\underline{U} = (U_1, U_2, \ldots, U_p)$ where each $V_i, U_i \subseteq T$, we define the structure $\mathfrak{A}' := (\mathfrak{A}, \underline{a}, \underline{V})$ to be the structure with vocabulary $\mathbb{S}_t = \mathbb{S} \cup \{\mathtt{c}_1, \ldots, \mathtt{c}_\ell\} \cup \{\mathtt{R}_1, \ldots, \mathtt{R}_p\}$ where $\mathtt{c}_i$ is a new constant associated to a domain elements $a_i \in T$, i.e., $\mathtt{c}_i^{\mathfrak{A}'} = a_i$ for every $i = 1, \ldots, \ell$, and $\mathtt{R}_i$ is a new relation of arity 1 associated to a set in the domain $V_i \subseteq T$, i.e., $\mathtt{R}_i^{\mathfrak{A}'} = V_i$ for every $i = 1, \ldots, p$. Analogously, we define the structure $\mathfrak{D}' := (\mathfrak{D}, \underline{d}, \underline{U})$.

Rules of the game. In the EF game, the game proceeds over ϱ rounds and there are two players, Alice and Bob, who play on the structures $\mathfrak{A}'$ and $\mathfrak{D}'$ with two different types of moves:

- **Point move.** Alice chooses a structure, $\mathfrak{A}'$ or $\mathfrak{D}'$, and an element that belongs to that structure. Bob responds with an element in the other structure.
- **Set move.** Alice chooses a structure, $\mathfrak{A}'$ or $\mathfrak{D}'$, and a subset that belongs to that structure. Bob responds with a subset of the other structure.

Winning condition. Suppose that in a ϱ-round game we have, for an integer $m \in \{0, \ldots, \varrho\}$, the following point moves and set moves:

- Point moves $(x_1, \ldots, x_m) \in T^m$ and $(y_1, \ldots, y_m) \in T^m$.

- Set moves $(X_1, \dots, X_{\varrho-m})$ and $(Y_1, \dots, Y_{\varrho-m})$ where $X_i, Y_i \subseteq T$ for each $i = 1, \dots, \varrho - m$.

Bob wins the ϱ-round game if the function

$$(\underline{a}, x_1, \dots, x_m) \mapsto (\underline{d}, y_1, \dots, y_m)$$

is a partial isomorphism between the two structures $(\mathfrak{A}', X_1, \dots, X_{\varrho-m})$ and $(\mathfrak{D}', Y_1, \dots, Y_{\varrho-m})$. Alice or Bob has a winning strategy if one can guarantee that the other player will lose regardless of the move played by this player.

Theorem 2 ([17]). *Given two structures $\mathfrak{A}$ and $\mathfrak{D}$, for every $\varrho \geq 0$, the following conditions are equivalent:*

- *Bob has a winning strategy for ϱ-round MSO games on $\mathfrak{A}$ and $\mathfrak{D}$.*
- *$\mathfrak{A} \equiv_\varrho \mathfrak{D}$, i.e., $\mathfrak{A}$ and $\mathfrak{D}$ agree on MSO sentences with a number of quantifiers that does not exceed ϱ.*

If a property Q is true for the structure $\mathfrak{A}$, false for the structure $\mathfrak{D}$, but these two structures can be shown equivalent by providing a winning strategy for Bob for any $\varrho \in \mathbb{N}$, then this shows that the property Q is not expressible in MSO.

3.3. *MSO* inexpressibility of the hashing properties

In this subsection, we consider the expressibility of the hashing properties in MSO. For notation convenience, we will explicitly address the case of the trifferent property on infinite words. On the other hand, with the same proof one can show that the (b, k)-hashing property, for finite and infinite words, is inexpressible by a single formula in MSO.

First, we introduce some notation. We define T_0 to be the subset of T of the nodes which begin with 0, T_1 to be the subset of the nodes which begin with 1 and, finally, T_2 to be the subset of the nodes which begin with 2. In constructing such sets we are assuming that the root $\mathtt{r}$ belongs to T_0, T_1 and T_2.

Given a structure $(\mathfrak{T}, x_1, \dots, x_\ell, X_0, \dots, X_{h-\ell})$ we define its restriction to T_j, for each $j = 0, 1, 2$, as

$$(\mathfrak{T}, x_1, \dots, x_\ell, X_0, \dots, X_{h-\ell})|_{T_j} = (\mathfrak{T}|_{T_j}, x_i : x_i \in T_j, X_0 \cap T_j, \dots, X_{h-\ell} \cap T_j)$$

where

$$\mathfrak{T}|_{T_j} = \left(T_j, (\succ_0^{\mathfrak{T}})|_{T_j}, (\succ_1^{\mathfrak{T}})|_{T_j}, (\succ_2^{\mathfrak{T}})|_{T_j}\right).$$

Lemma 2. *Given $M \in \mathbb{N}$, in the ternary $\mathfrak{T}$ tree, there exist two different infinite words of the form $X_0 = 0 \dots 0100 \dots$ and $Y_0 = 0 \dots 0100 \dots$ (where the number of zeros before the one differs in the two cases) and, for any formula $p(\cdot)$ of length at most M, it results $p(X_0) \iff p(Y_0)$.*

Proof. Denoted by $\mathcal{F}_M$ the family of the formulas of length at most M with exactly one free variable (of set type), we have that $|\mathcal{F}_M|$ is finite. On the other hand, T_0 contains infinitely many infinite words of the form $0\dots0100\dots$. It follows that there are infinitely many infinite words of that form that coincide in any formulas of $\mathcal{F}_M$. Hence we can also find two infinite words $X_0, Y_0 \subseteq T_0$ such that $X_0 = 0\dots0100\dots$ and $Y_0 = 0\dots0100\dots$ and the number of zeros before the one differs in the two cases and such that for any formula $p \in \mathcal{F}_M$, it holds $p(X_0) \iff p(Y_0)$. □

Remark 1. We have already noted that the set of all the formulas of length at most M with exactly one free set variable (resp. individual variable) is finite. The same can also be said for the family $\mathcal{F}_{M,i,j}$ of the formulas of length at most M with exactly i free variables of individual type and j free variables of set type.

Proposition 1. *There exist two infinite words $X_0, Y_0 \subseteq T_0$ of the form $X_0 = 0\dots0100\dots$ and $Y_0 = 0\dots0100\dots$ (with a different number of zeros before the one) such that $(\mathfrak{T}, X_0)|_{T_0} \equiv_\varrho (\mathfrak{T}, Y_0)|_{T_0}$.*

Proof. Let us consider a sequence $M_0, M_1, \dots, M_\varrho$ of natural numbers such that $M_0 = 1$ and, for any h, $1 \leq h \leq \varrho$, take

$$M_h \geq \sum_{i,j \leq \varrho} |\mathcal{F}_{M_{h-1},i,j}|(M_{h-1}+2)+2.$$

Note that such a sequence exists since $|\mathcal{F}_{M_{h-1},i,j}|$ is finite.
According to Lemma 2, we can choose $X_0 = 0\dots0100\dots$ and $Y_0 = 0\dots0100\dots$, where the number of zeros before the one differs in the two cases, be two infinite words of T_0 such that for any formula $p(\cdot)$ of length at most M_ϱ we have $p(X_0) \iff p(Y_0)$. We prove that $(\mathfrak{T}, X_0)|_{T_0} \equiv_\varrho (\mathfrak{T}, Y_0)|_{T_0}$ inductively, using the EF game. More precisely, we prove that, if at the h-th step Alice chooses the point x_ℓ (resp. the set $X_{h-\ell+1}$) and considering the structure $(x_1, \dots, x_\ell, X_0, \dots, X_{h-\ell})$, Bob can choose y_ℓ (resp. the set $Y_{h-\ell+1}$) so that any formula of $\mathcal{F}_{M_{\varrho-h},\ell,h-\ell+1}$ is realized on $(y_1, \dots, y_\ell, Y_0, \dots, Y_{h-\ell})$ if and only if it holds on $(x_1, \dots, x_\ell, X_0, \dots, X_{h-\ell})$. Equivalently we can say that $(y_1, \dots, y_\ell, Y_0, \dots, Y_{h-\ell})$ is equivalent to $(x_1, \dots, x_\ell, X_0, \dots, X_{h-\ell})$ for any formula of length at most $M_{\varrho-h}$.
Base step:
Let us suppose that Alice chooses a node x_1. We denote by $\mathcal{F}^t_{M_{\varrho-1},1,1}$ the set of all the formulas of $\mathcal{F}_{M_{\varrho-1},1,1}$ that are true on (x_1, X_0) and we set

$$\mathcal{F}^f_{M_{\varrho-1},1,1} := \mathcal{F}_{M_{\varrho-1},1,1} \setminus \mathcal{F}^t_{M_{\varrho-1},1,1}.$$

We note that

$$F_\varrho(\cdot) := \exists z : \cup_{f \in \mathcal{F}^t_{M_{\varrho-1},1,1}} f(z,\cdot) \cup_{g \in \mathcal{F}^f_{M_{\varrho-1},1,1}} \neg g(z,\cdot)$$

has length at most M_ϱ. Also, this formula is true on X_0 and so this means that it is true also on Y_0, but then

$$\exists z : \cup_{f\in\mathcal{F}^t_{M_{\varrho-1},1,1}} f(z, Y_0) \cup_{g\in\mathcal{F}^f_{M_{\varrho-1},1,1}} \neg g(z, Y_0).$$

Hence Bob can choose y_1 so that for any formula $p \in \mathcal{F}_{M_{\varrho-1},1,1}$, it holds: $p(x_1, X_0) \iff p(y_1, Y_0)$.
Similarly, if Alice chooses a set X_1, Bob can find a set Y_1 so that for any formula $p \in \mathcal{F}_{M_{\varrho-1},0,2}$, we have $p(X_0, X_1) \iff p(Y_0, Y_1)$.
Inductive Step:
Let us suppose that, at the h-th step, with $h \leq \varrho$, Alice chooses a node x_ℓ. We denote by $\mathcal{F}^t_{M_{\varrho-h},\ell,h-\ell+1}$ the set of all the formulas of $\mathcal{F}_{M_{\varrho-h},\ell,h-\ell+1}$ that are true on $(x_1, \ldots, x_\ell, X_0, \ldots X_{h-\ell})$ and we set

$$\mathcal{F}^f_{M_{\varrho-h},\ell,h-\ell+1} := \mathcal{F}_{M_{\varrho-h},\ell,h-\ell+1} \setminus \mathcal{F}^t_{M_{\varrho-h},\ell,h-\ell+1}.$$

Then, proceeding as before, Bob chooses y_ℓ so that for any formula $p \in \mathcal{F}_{M_{\varrho-h},\ell,h-\ell+1}$, we have:

$$p(x_1, \ldots, x_\ell, X_0, \ldots X_{h-\ell}) \iff p(y_1, \ldots, y_\ell, Y_0, \ldots Y_{h-\ell}).$$

Note that such a y_ℓ exists because we can consider the formula

$$F_{\varrho-(h-1)}(\cdot) := \exists z : \cup_{f\in\mathcal{F}^t_{M_{\varrho-h},\ell,h-\ell+1}} f(\cdots, z, \cdots) \cup_{g\in\mathcal{F}^f_{M_{\varrho-h},\ell,h-\ell+1}} \neg g(\cdots, z, \cdots)$$

which is true on $(x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell})$ and so, since its length is smaller than $M_{\varrho-(h-1)}$, it must be true also on

$$(y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}).$$

Similarly, if Alice chooses a set $X_{h-\ell+1}$, Bob can find a set $Y_{h-\ell+1}$ so that for any formula $p \in \mathcal{F}_{M_{\varrho-h},\ell-1,h-\ell+2}$, we have:

$$p(x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell+1}) \iff p(y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell+1}).$$

□

Proposition 2. *Assume $\varrho \geq h \geq \ell \geq 1$ and*

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell})|_{T_j} \equiv_\varrho (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell})|_{T_j},$$

for $j = 0, 1, 2$. This implies that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell}) \equiv_\varrho (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}).$$

Proof. Note that if $\ell = 1$, no x_i and y_i appears in the structure. We prove the thesis using the EF game inductively on ϱ.

Base step.

Let us suppose, by contradiction, that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell}) \not\equiv_0 (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}).$$

This means that either $x_j \succ_0 x_i$ (resp. $\succ_1$, $\succ_2$) but $y_j \not\succ_0 y_i$ (resp. $\succ_1$, $\succ_2$) or $x_j \in X_i$ but $y_j \notin Y_i$. In the first case, since $x_j \succ_0 x_i$, we may assume that $x_j, x_i \in T_0$. Hence, since

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell})|_{T_0} \equiv_0 (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell})|_{T_0}$$

we also have $y_j, y_i \in T_0$ and $y_j \succ_0 y_i$.

We proceed similarly in the second case. Here we may assume $x_j \in T_0$ and $x_j \in X_i \cap T_0$ and it would follow that also $y_j \in T_0$, and $y_j \in Y_i \cap T_0$. Hence we must have that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell}) \equiv_0 (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}).$$

Inductive step.

Let us suppose that Alice chooses a node $x_{\ell+1} \in T_0$. Since

$$(\mathfrak{T}, x_1, \ldots, x_{\ell}, X_0, \ldots, X_{h-\ell})|_{T_0} \equiv_{\varrho} (\mathfrak{T}, y_1, \ldots, y_{\ell}, Y_0, \ldots, Y_{h-\ell})|_{T_0},$$

Bob can choose $y_{\ell+1} \in T_0$ so that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell}, X_0, \ldots, X_{h-\ell})|_{T_0} \equiv_{\varrho-1} (\mathfrak{T}, y_1, \ldots, y_{\ell}, Y_0, \ldots, Y_{h-\ell})|_{T_0}.$$

Since, for $j = 1, 2$,

$$(\mathfrak{T}, x_1, \ldots, x_{\ell}, X_0, \ldots, X_{h-\ell})|_{T_j} = (\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell})|_{T_j}$$

and

$$(\mathfrak{T}, y_1, \ldots, y_{\ell}, Y_0, \ldots, Y_{h-\ell})|_{T_j} = (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell})|_{T_j}.$$

We also have that, for $j = 1, 2$,

$$(\mathfrak{T}, x_1, \ldots, x_{\ell}, X_0, \ldots, X_{h-\ell})|_{T_j} \equiv_{\varrho-1} (\mathfrak{T}, y_1, \ldots, y_{\ell}, Y_0, \ldots, Y_{h-\ell})|_{T_j}.$$

Due to the inductive hypothesis, we obtain that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell}, X_0, \ldots, X_{h-\ell}) \equiv_{\varrho-1} (\mathfrak{T}, y_1, \ldots, y_{\ell}, Y_0, \ldots, Y_{h-\ell}).$$

But this means that Bob has a winning strategy for the $\varrho - 1$ moves EF game on

$$(\mathfrak{T}, x_1, \ldots, x_\ell, X_0, \ldots, X_{h-\ell}) \text{ and } (\mathfrak{T}, y_1, \ldots, y_\ell, Y_0, \ldots, Y_{h-\ell}).$$

If Alice chooses a set $X_{h-\ell+1}$, proceeding similarly, Bob can choose sets $Y^0_{h-\ell+1} \subseteq T_0$, $Y^1_{h-\ell+1} \subseteq T_1$ and $Y^2_{h-\ell+1} \subseteq T_2$ such that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell+1})|_{T_j} \equiv_{\varrho-1} (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y^j_{h-\ell+1})|_{T_j}$$

for $j = 0, 1, 2$. Setting

$$Y_{h-\ell+1} := Y^0_{h-\ell+1} \cup Y^1_{h-\ell+1} \cup Y^2_{h-\ell+1}$$

we obtain that

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell+1}) \equiv_{\varrho-1} (\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell+1}).$$

It follows that Bob has a winning strategy for the $\varrho - 1$ moves EF game on

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell}, X_{h-\ell+1})$$

and

$$(\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}, Y_{h-\ell+1}).$$

Summing up these two cases, we obtain that Bob has a winning strategy for the ϱ moves EF game on

$$(\mathfrak{T}, x_1, \ldots, x_{\ell-1}, X_0, \ldots, X_{h-\ell})$$

and

$$(\mathfrak{T}, y_1, \ldots, y_{\ell-1}, Y_0, \ldots, Y_{h-\ell}).$$

□

Theorem 3. *There is no S3S formula $f(\cdot, \cdot, \cdot)$ such that $f(X, Y, Z)$ is true if and only if the infinite words X, Y, Z are trifferent.*

Proof. Let us assume, by contradiction, that such a formula f exists and let ϱ be its rank. We consider distinct infinite words $X_0, Y_0 \subseteq T_0$ of the form $0 \ldots 0100 \ldots$ such that $(\mathfrak{T}, X_0)|_{T_0} \equiv_\varrho (\mathfrak{T}, Y_0)|_{T_0}$. Let now consider $X_1, X_2 \subseteq T_1$ of the form $X_1 = 1 \ldots 111 \ldots$ and $X_2 = 1 \ldots 1211 \ldots$ where the number m of ones before the 2 in X_2 coincides with the number of zeros before the 1 of X_0. Set $Y_1 = X_1$ and $Y_2 = X_2$ we have that,

$$(\mathfrak{T}, X_0)|_{T_0} \equiv_\varrho (\mathfrak{T}, Y_0)|_{T_0}$$

and also

$$(\mathfrak{T}, X_1, X_2)|_{T_1} \equiv_\varrho (\mathfrak{T}, Y_1, Y_2)|_{T_1}.$$

Hence, according to Proposition 2, we have that

$$(\mathfrak{T}, X_0, X_1, X_2) \equiv_\varrho (\mathfrak{T}, Y_0, Y_1, Y_2).$$

On the other hand the words X_0, X_1, X_2 are not trifferent and hence $f(X_0,\ X_1, X_2)$ is not realized. Then, we also have that $f(Y_0, Y_1, Y_2)$ should be false. But, since in the $(m+1)$-th coordinate of Y_0, Y_1, Y_2 are, respectively, $0, 1, 2$, these words are trifferent, contradicting the hypothesis that $f(X, Y, Z)$ is true if and only if X, Y, Z are trifferent. □

With the same proof, one can check that the following result holds.

Theorem 4. *Let $3 \leq k \leq b$. Then, there is no SbS formula $f(\cdot, \cdot, \ldots, \cdot)$ such that $f(X_0, X_1, \ldots,\ X_{k-1})$ is true if and only if the infinite words $(X_0, X_1, \ldots, X_{k-1})$ satisfy the (b, k)-hashing property.*
Analogously, there is no SbS formula $f(\cdot, \cdot, \ldots, \cdot)$ such that $f(x_1, x_2, \ldots, x_k)$ is true if and only if the finite words $(x_1, x_2, \ldots, x_k)$ satisfy the (b, k)-hashing property.

4. A computational approach

In this section we complement the definability results with a solver-based experiment. Although MSO is not expressive enough to capture the trifference relation uniformly (Section 3), the constraint itself is naturally phrased as a finite satisfiability problem and can therefore be attacked by general-purpose SMT technology. We emphasize that the goal here is not to outperform the specialized enumeration algorithms available in the literature, but rather to illustrate that the problem admits a clean SMT encoding and to point out a logical fragment in which the trifference relation becomes definable.

4.1. An SMT encoding in Z3

Fix integers $m, n \geq 1$. We model a candidate code $C \subseteq \{0, 1, 2\}^n$ of size m by an $m \times n$ matrix of integer variables

$$C = \big(c_{r,i}\big)_{1 \leq r \leq m,\ 1 \leq i \leq n}, \qquad c_{r,i} \in \{0, 1, 2\}.$$

The trifference requirement says that for every triple of distinct rows (r_1, r_2, r_3) there exists a column i such that the three entries in that column are pairwise different. In SMT this naturally becomes a conjunction over the $\binom{m}{3}$ triples of a disjunction over the n coordinates. Note that the size of the constraint system grows as $O\big(\binom{m}{3}\, n\big)$, and each disjunct is a small, purely quantifier-free condition. This structure is well-suited to modern engines: the solver can branch on a small number of matrix entries and rapidly propagate consequences across many triples.

For concreteness we provide a Z3Py encoding. Compared to a direct brute-force enumeration, the SMT encoding delegates search, propagation, and conflict learning to the solver. In particular, the same encoding can be augmented with symmetry-breaking constraints (e.g. fixing one row to 0^n, normalizing symbol permutations in the first coordinate, etc.) without changing its conceptual core; we did not pursue such optimizations here, as our aim is primarily methodological.

Algorithm 1. **Z3 encoding**

```
from z3 import *
import itertools

def getTrifferenceSolver(m, n):
    # Matrix of integer variables: C[r][i] is the i-th symbol of the r-th codeword
    C = [[Int(f"x_{r+1}_{i+1}") for i in range(n)] for r in range(m)]

    # Vocabulary constraints: restrict each entry to {0,1,2}
    vocab = [And(0 <= C[r][i], C[r][i] <= 2) for r in range(m) for i in range(n)]

    # Trifference constraints:
    # for each triple of distinct rows, enforce existence of a coordinate where the three symbols
      ↪ are pairwise distinct.
    triples = itertools.combinations(range(m), 3)
    triff = []
    for (a, b, c) in triples:
        triff.append(Or([Distinct(C[a][i], C[b][i], C[c][i]) for i in range(n)]))

    s = Solver()
    s.add(vocab)
    s.add(triff)
    return s, C
```

The predicate `Distinct` is a concise way to express pairwise disequality and typically yields a more compact internal representation. Observe also that the constraint set is *uniform* in (m, n) and can be reused verbatim when changing parameters (or when moving to (b, k)-hashing by replacing $\{0, 1, 2\}$ and the arity of the `Distinct` constraints).

4.2. The instance $(m, n) = (11, 5)$ and comparison with the literature

Running the solver produced by Algorithm 1 for $(m, n) = (11, 5)$ returns `unsat`, i.e. there exists no trifferent code of length 5 with 11 codewords. Equivalently, this certifies the upper bound $T(5) \leq 10$.

This outcome is consistent with the exact determination $T(5) = 10$ established in [7], that rule out size 11 at length 5 via a highly optimized exhaustive search procedure. Kurz [15] later uses the value $T(5) = 10$ (together with $T(6) = 13$ from [7]) as a starting point for further exact computations and classifications for lengths $7, 8, 9$.

At present, our SMT experiment does *not* improve the best-known exact values or bounds. Nonetheless, it provides a complementary viewpoint: the trifference condition admits a simple, solver-friendly, and parameter-uniform encoding as quantifier-free constraints over integers. To the best of our knowledge, this SMT-based formulation has not been explicitly proposed in the trifferent-code literature, where the dominant approach relies on strong symmetry reduction and combinatorial pruning. The SMT encoding may become more competitive as solver technology and symmetry-

breaking heuristics are refined, or when additional side-constraints (e.g. structural constraints on codes) are imposed and can be expressed naturally in SMT.

4.3. A specification in MSO with cardinality constraints

The Z3 encoding in Algorithm 1 is quantifier-free but it expands explicitly into $O\big(\binom{m}{3}n\big)$ local constraints, one for each triple of codewords and each coordinate. In contrast, if MSO is extended with equicardinality atoms (as in MSO with cardinalities [10] and related counting extensions), the existence of a trifferent code of *size m and length n* can be expressed by a *single sentence of constant size*, i.e. a formula whose length does not grow with m and n. The parameters (m, n) are provided by the input structure through two monadic predicates whose cardinalities encode m and n.

We assume an atomic predicate $\mathsf{EqCard}(X, Y)$ meaning "X and Y have the same (finite) cardinality". This allows us to compare the size of the code with a parameter set M (encoding m), and to compare depths with a parameter set N (encoding n).

Recall that the binary relations $\succ_j$ are the labelled successor relations of the tree: $u \succ_j v$ means that v is the j-successor (i.e. the j-child) of u. We set

$$\mathrm{Succ}(u, v) \;:=\; \bigvee_{j=0}^{2} (u \succ_j v).$$

The prefix/ancestor relation $\preceq$ ("x is an ancestor of y", i.e. y is reachable from x by iterating Succ) is MSO-definable via closure under successors:

$$x \preceq y \;:=\; \forall X \Big(\big(x \in X \;\wedge\; \forall u \forall v \,((u \in X \wedge \mathrm{Succ}(u, v)) \rightarrow v \in X)\big) \;\rightarrow\; y \in X \Big).$$

Finally, we write $\mathrm{Anc}(A, x)$ for the fact that A is *exactly* the set of ancestors of x (including the root $\mathtt{r}$ and x itself), i.e. the set of prefixes of x:

$$\mathrm{Anc}(A, x) \;:=\; \forall u \,\big(u \in A \leftrightarrow u \preceq x\big).$$

Given a monadic parameter N, we enforce that a node x has depth encoded by N through:

$$\mathrm{DepthIs}(x, N) \;:=\; \exists A \,\big(\mathrm{Anc}(A, x) \wedge \mathsf{EqCard}(A, N)\big).$$

Hence, if N is interpreted as an ancestor chain of size $n{+}1$, then $\mathrm{DepthIs}(x, N)$ forces x to lie at depth n.

For a node $u \preceq x$ and $j \in \{0, 1, 2\}$, the formula

$$\mathrm{Next}_j(u, x) \;:=\; \exists v \,\big(u \succ_j v \wedge v \preceq x\big),$$

meaning that x lies in the subtree rooted at the j-successor of u.

We can now express that three nodes x, y, z (at the same depth) are trifferent by synchronizing a common depth position (using equicardinality on ancestor paths) and requiring a permutation of digits $\{0, 1, 2\}$ at that position:

$$\begin{aligned}\mathrm{Triff}(x,y,z) \;:=\; \exists u_x\,\exists u_y\,\exists u_z\,\exists A_x\,\exists A_y\,\exists A_z\,\Big(&u_x \preceq x \wedge u_y \preceq y \wedge u_z \preceq z \;\wedge\\ \mathrm{Anc}(A_x,u_x)\wedge\mathrm{Anc}(A_y,u_y)\wedge\mathrm{Anc}(A_z,u_z) \;\wedge\; &\mathsf{EqCard}(A_x,A_y)\wedge\mathsf{EqCard}(A_y,A_z)\;\wedge\\ &\bigvee_{\pi\in S_3}\big(\mathrm{Next}_{\pi(1)}(u_x,x)\wedge\mathrm{Next}_{\pi(2)}(u_y,y)\wedge\mathrm{Next}_{\pi(3)}(u_z,z)\big)\Big),\end{aligned}$$

where S_3 is the symmetric group over the set $\{0, 1, 2\}$.
We extend the vocabulary with two monadic predicates M and N that act as *parameters*: the intended instances interpret $|M| = m$ and $|N| = n{+}1$ (e.g. by taking M and N to be ancestor paths of suitable nodes). Then the following MSO+EqCard sentence has *constant size* and does not expand with (m, n):

$$\begin{aligned}\varphi(M,N) \;:=\; \exists C\,\Big(&\mathsf{EqCard}(C,M)\;\wedge\;\forall x\,(x\in C\rightarrow \mathrm{DepthIs}(x,N))\;\wedge\\ &\forall x\,\forall y\,\forall z\,\Big((x\in C\wedge y\in C\wedge z\in C\wedge x\neq y\wedge x\neq z\wedge y\neq z)\;\rightarrow\;\mathrm{Triff}(x,y,z)\Big)\Big).\end{aligned}$$

In other words, unlike the SMT encoding where the constraint set grows combinatorially with m and linearly with n, here *a single quantified formula* captures the entire specification, with m and n being supplied implicitly through the cardinalities of M and N.
While MSO with cardinality features has a rich theory and several decidability results (see, e.g., [9, 10]), we are not aware of an off-the-shelf solver that combines a full *SbS*-style MSO engine with global equicardinality constraints in the above sense. Thus, at present, the practical route is to encode fixed instances (m, n) into standard SMT as in Algorithm 1.
To make the previous definitional discussion operational, one would want an API that can build MSO set-quantifiers, tree predicates, and equicardinality atoms in the same constraint language. In Algorithm 2, we provide a pseudo-code that illustrates the construction of the sentence "there exists a trifferent code of size m and length n" inside such a hypothetical solver. This is *not* executable with current mainstream tools, and is included only to clarify the intended integration point between MSO reasoning and cardinality constraints.

Algorithm 2. **MSO+EqCard encoding**

```
# Hypothetical MSO+EqCard solver API over the ternary tree.
# M and N are fixed monadic predicates in the input structure:
#   |M| = m,  |N| = n+1.
# The constraint is a single sentence phi(M,N), independent of (m,n).

def DepthIs(x, N):
    A = SetVar("A")
    return Exists([A], And(Anc(A, x), EqCard(A, N)))
```

```
def Next(j, u, x):
    v = NodeVar("v")
    return Exists([v], And(Succ(j, u, v), Prefix(v, x)))

def Triff(x, y, z):
    ux, uy, uz = NodeVar("ux"), NodeVar("uy"), NodeVar("uz")
    Ax, Ay, Az = SetVar("Ax"), SetVar("Ay"), SetVar("Az")

    perms = []
    for (a,b,c) in [(0,1,2),(0,2,1),(1,0,2),(1,2,0),(2,0,1),(2,1,0)]:
        perms.append(And(Next(a, ux, x), Next(b, uy, y), Next(c, uz, z)))

    return Exists([ux, uy, uz, Ax, Ay, Az],
        And(Prefix(ux, x), Prefix(uy, y), Prefix(uz, z),
            Anc(Ax, ux), Anc(Ay, uy), Anc(Az, uz),
            EqCard(Ax, Ay), EqCard(Ay, Az),
            Or(perms))
    )

def Phi(M, N):  # single uniform constraint
    C = SetVar("C")
    x, y, z = NodeVar("x"), NodeVar("y"), NodeVar("z")

    return Exists([C],
        And(EqCard(C, M),
            Forall([x], Implies(In(x, C), DepthIs(x, N))),
            Forall([x, y, z],
                Implies(And(In(x, C), In(y, C), In(z, C),
                            x != y, x != z, y != z),
                        Triff(x, y, z)))
        )
    )

# solver.check(Phi(M, N))
```

Acknowledgements: This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, Partnership on "Telecommunications of the Future," Program "RESTART" under Grant PE00000001, "Netwin" Project (CUP E83C22004640001).

Conflict of Interest: The authors declare that they have no conflict of interest.

Data Availability: Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

[1] S. Bhandari and A. Khetan, *Improved upper bound for the size of a trifferent code*, Combinatorica **45** (2025), Article number: 2 https://doi.org/10.1007/s00493-024-00130-2.

[2] A. Bishnoi, J. D'haeseleer, D. Gijswijt, and A. Potukuchi, *Blocking sets, minimal codes and trifferent codes*, J. London Math. Soc. **109** (2024), no. 6, e12938. https://doi.org/10.1112/jlms.12938.

[3] S. Costa and M. Dalai, *A gap in the slice rank of k-tensors*, J. Comb. Theory, Series A **177** (2021), 105335. https://doi.org/10.1016/j.jcta.2020.105335.

[4] B. Courcelle, *The monadic second-order logic of graphs, II: Infinite graphs of bounded width*, Math. Systems Theory **21** (1988), no. 1, 187–221. https://doi.org/10.1007/BF02088013.

[5] ———, *The monadic second-order logic of graphs. I. Recognizable sets of finite graphs*, Info. Comput. **85** (1990), no. 1, 12–75. https://doi.org/10.1016/0890-5401(90)90043-H.

[6] L. De Moura and N. Bjørner, *Z3: An efficient SMT solver*, International conference on Tools and Algorithms for the Construction and Analysis of Systems, vol. 4963.

[7] S. Della Fiore, A. Gnutti, and S. Polak, *The maximum cardinality of trifferent codes with lengths 5 and 6*, Examples and Counterexamples **2** (2022), 100051. https://doi.org/10.1016/j.exco.2022.100051.

[8] J.G. Henriksen, J. Jensen, M. Jørgensen, N. Klarlund, R. Paige, T. Rauhe, and A. Sandholm, *Mona: Monadic second-order logic in practice*, International workshop on tools and algorithms for the construction and analysis of systems, vol. 1019, Springer, 1995, pp. 89–110.

[9] L. Herrmann, V. Peth, and S. Rudolph, *Decidable (ac) counting with Parikh and Muller: adding Presburger arithmetic to monadic second-order logic over tree-interpretable structures*, 32nd EACSL Annual Conference on Computer Science Logic (2024).

[10] F. Klaedtke and H. Rueß, *Monadic second-order logics with cardinalities*, International Colloquium on Automata, Languages, and Programming, vol. 2719, Springer, 2003, pp. 681–696.

[11] J. Körner and S. Gábor, *Trifference*, Studia Scientiarum Mathematicarum Hungarica **30** (1995), 95–103.

[12] J. Körner and K. Marton, *New bounds for perfect hashing via information theory*, Eur. J. Comb. **9** (1988), no. 6, 523–530. https://doi.org/10.1016/S0195-6698(88)80048-9.

[13] V. Kuncak, H.H. Nguyen, and M. Rinard, *An algorithm for deciding BAPA: Boolean algebra with Presburger arithmetic*, International Conference on Automated Deduction, vol. 3632, Springer, 2005, pp. 260–277.

[14] ———, *Deciding Boolean algebra with Presburger arithmetic*, J. Automat. Reasoning **36** (2006), no. 3, 213–239. https://doi.org/10.1007/s10817-006-9042-1.

[15] S. Kurz, *Trifferent codes with small lengths*, Examples and Counterexamples **5** (2024), 100139. https://doi.org/10.1016/j.exco.2024.100139.

[16] H. Läuchli and C. Savioz, *Monadic second order definable relations on the binary tree*, The Journal of Symbolic Logic **52** (1987), no. 1, 219–226. https://doi.org/10.2307/2273878.

[17] L. Libkin, *Elements of Finite Model Theory*, vol. 41, Springer, 2004.