TUCS

Vesa Halava | Tero Harju | Tomi Kärki

Relational codes of words

Turku Centre for Computer Science

TUCS Technical Report
No 767, April 2006

Relational codes of words

Vesa Halava
Department of Mathematics and
TUCS - Turku Centre for Computer Science
University of Turku FIN-20014 Turku, Finland
Supported by the Academy of Finland under grant 208414.
vehalava@utu.fi

Tero Harju
Department of Mathematics and
TUCS - Turku Centre for Computer Science
University of Turku FIN-20014 Turku, Finland
harju@utu.fi

Tomi Kärki
Department of Mathematics and
TUCS - Turku Centre for Computer Science
University of Turku FIN-20014 Turku, Finland
topeka@utu.fi

TUCS Technical Report
No 767, April 2006

Abstract

We consider words, i.e., strings over a finite alphabet together with a compatibility relation induced by a relation on letters. This notion generalizes that of partial words. The theory of codes on combinatorics on words is revisited by defining (R, S)-codes for arbitrary relations R and S. We describe an algorithm to test whether or not a finite set of words is an (R, S)-code. Coding properties of finite sets of words are explored by finding maximal and minimal relations with respect to relational codes.

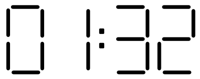
Keywords: code, relational code, partial word, NP-completeness

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

Codes are an essential tool in information theory, and the theory of variable length codes is firmly related to combinatorics on words [2]. The object of the theory is to study factorization of words into sequences of words taken from a given set. In the semigroup X^+ generated by a code X there does not exist two distinct factorizations in X for any word. This coding property can be strengthened by requiring that two nearly similar, i.e., *compatible* words, have the same, or at least similar, factorizations. This is attained here by introducing *word relations* and *relational codes*. The similarity of words is described by using relations on letters of the alphabet. If some of the letters in a message are changed to related letters, the message can still be factorized, in other words decoded, in a proper manner. Thus these codes possess some error correction capabilities.

As an example we may think of a digital clock with at most one broken led. The possible interpretations of the following display are $01 : 32$, $81 : 32$, $07 : 32$ and $01 : 92$. Thus with some extra information we may still conclude the right



time quite reliably despite of the broken led. Note that the similarity relation does not need to be transitive. There is a difference of one led between the displays of numbers 5 and 6 and also 6 and 8, but you cannot confuse 5 and 8 with each other if only one malfunctioning led is possible.

On the other hand, we may think of situations where part of the information is just missing. This is the concept of partial words introduced by J. Berstel and L. Boasson in 1999 [1]. Partial words can be interpreted as total words with a "do not know" symbol $\diamondsuit$. Two partial words are said to be compatible if there exists a total word such that by replacing each $\diamondsuit$ symbol of the partial words with the letter in a corresponding position of the total word we make the two partial words equal. For example, we see that the following partial words are compatible by comparing them with the total word "knowledge".

$$\begin{array}{ccccccccc} k & n & \diamondsuit & w & l & \diamondsuit & d & g & e \\ \diamondsuit & n & o & w & \diamondsuit & \diamondsuit & d & g & \diamondsuit \\ k & n & o & w & l & e & d & g & e \end{array}$$

Word relations can also model this kind of missing information. Namely the compatibility relation of partial words can be seen as a special case of word relations induced by a relation on letters of the alphabet $A_\diamondsuit = A \cup \{\diamondsuit\}$.

Combinatorics on partial words has been widely studied under the recent years; see [3–10]. Motivation for this research comes from the study of biological sequences such as DNA, RNA and proteins. In *sequence comparison* you align two sequences, for example genes, in order to find correspondence between them. This alignment corresponds to a construction of compatible partial words.

Clearly also the word relations can be made to describe this phenomenon with specific similarity relations on symbols. Another important operation in molecular biology is *DNA sequencing.* There the role of partial words as well as word relations is to model the task of fragment assembly. We introduce gaps (indicated by $\diamondsuit$) in DNA pieces ($addg, dgtgc, ccad$) to let the nucleotides align perfectly.

$$\begin{array}{ccccccccc} \diamondsuit & \diamondsuit & a & d & d & g & \diamondsuit & \diamondsuit & \diamondsuit \\ \diamondsuit & \diamondsuit & \diamondsuit & \diamondsuit & d & g & t & g & c \\ c & c & a & d & \diamondsuit & \diamondsuit & \diamondsuit & \diamondsuit & \diamondsuit \end{array}$$

Partial words have also been considered in DNA computing as good solutions to DNA encodings [14].

In this paper we introduce word relations together with (R, S)-codes for given compatibility relations R and S of words, and we consider algorithmic questions on these codes. We show that the maximal relation problem is NP-complete. In this problem one is given a finite set X and a compatibility relation S induced by letters, and one is to determine whether X is an (R, S)-code for some compatibility relation R induced by at least k pairs.

We end this section with some notation. An *alphabet* A is a nonempty finite set of symbols and a *word* over A is a (finite or infinite) sequence of symbols from A. The empty word is denoted by ε. The sets of all finite words and finite nonempty words over A are denoted by A^* and A^+, respectively. With the operation of catenation A^* is a free monoid and A^+ is a free semigroup generated by the letters of A. The *length* of a word w, denoted by $|w|$, is the total number of (occurrences of) letters in w. The ith symbol of the word w is denoted by $w(i)$. A word w is a *factor* of a word u (resp. a left factor or a *prefix*, a right factor or a *suffix*), if there exist words x and y such that $u = xwy$ (resp. $u = wy, u = xw$). If $w = uv$ then we denote $v = u^{-1}w$.

For subsets $L, K \subseteq A^*$, we let

$$\begin{aligned} LK &= \{uv \mid u \in L, v \in K\}, \\ L^+ &= \bigcup_{i \geq 1} L^i, \quad L^* = L^+ \cup \{\varepsilon\}, \\ L^{-1}K &= \{u^{-1}w \mid u \in L, w \in K\}. \end{aligned}$$

2 Word relations

Let $R \subseteq X \times X$ be a relation on a set X. We often write $x\,R\,y$ instead of $(x, y) \in R$. Then R is a *compatibility relation* if it is both reflexive and symmetric, i.e., (i) $\forall x \in X : \; x\,R\,x$, and (ii) $\forall x, y \in X : \; x\,R\,y \implies y\,R\,x$.

The *identity relation* on a set X is defined by

$$\iota_X = \{(x, x) \mid x \in X\}$$

and the *universal relation* on X is defined by

$$\Omega_X = \{(x,y) \mid x, y \in X\}.$$

Subscripts are often omitted when they are clear from the context. Clearly, both ι_X and Ω_X are compatibility relations on X.

A compatibility relation $R \subseteq A^* \times A^*$ on the set of all words will be called a *word relation* if it is induced by its restriction on the letters, i.e.,

$$a_1 \cdots a_m \, R \, b_1 \cdots b_n \quad \Longleftrightarrow \quad m = n \text{ and } a_i \, R \, b_i \text{ for all } i = 1, 2, \ldots, m$$

whenever $a_1, \ldots, a_m, b_1, \ldots, b_n \in A$.

Let S be a relation on A. By $\langle S \rangle$ we denote the compatibility relation *generated* by S, i.e., $\langle S \rangle$ is the reflexive and symmetric closure of the relation S. Sometimes we need to consider the restriction of a relation R on a subset X of A^*. We denote $R_X = R \cap (X \times X)$. Words u and v satisfying $u \, R \, v$ are said to be *compatible* or, more precisely, *R-compatible*. For example, in the binary alphabet $A = \{a, b\}$ the compatibility relation R induced by $\langle \{(a,b)\} \rangle = \{(a,a), (b,b), (a,b), (b,a)\}$ makes all words with equal length compatible with each other. In the ternary alphabet $\{a, b, c\}$ we have $abba \, R \, baab$ but, for instance, words abc and cac are not compatible.

Clearly a word relation R satisfies the following two conditions:

$$\begin{array}{lccc} \text{multiplicativity:} & u \, R \, v, u' \, R \, v' & \Longrightarrow & uu' \, R \, vv', \\ \text{simplifiability:} & uu' \, R \, vv', |u| = |v| & \Longrightarrow & u \, R \, v, \; u' R v'. \end{array}$$

However, a word relation R does not need to be transitive. *From now on the relations on words considered in this presentation are supposed to be word relations induced by some compatibility relation on letters.*

Let 2^X denote the *power set* of X, that is, the family of all subsets of X including the empty set $\emptyset$ and X itself. For a word relation R on A^*, let the corresponding function $R \colon 2^{A^*} \to 2^{A^*}$ be defined by

$$R(X) = \{u \in A^* \mid \exists \, x \in X : \, x \, R \, u\}.$$

If X contains only one word $w \in A^*$, we denote $R(X)$ shortly by $R(w)$. The function R is multiplicative in the following sense.

Proposition 1. *Let R be a word relation on A^*. Then $R(X)R(Y) = R(XY)$ for all $X, Y \subseteq A^*$. Especially, $R(X)^* = R(X^*)$ for all $X \subseteq A^*$.*

Proof. Suppose that w belongs to $R(X)R(Y)$. Then there exist words $u \in R(X)$ and $v \in R(Y)$ such that $w = uv$. In other words there exist $x \in X$ and $y \in Y$ such that $u \, R \, x$ and $v \, R \, y$. By the multiplicativity of the relation R we have $uv \, R \, xy$, and thus $w \in R(XY)$.

Conversely, let w belong to $R(XY)$. Then there exist words $x \in X$ and $y \in Y$ such that $w \, R \, xy$. By the definition of a word relation this means that

$|w| = |x| + |y|$. Thus w can be factored into two parts u and v satisfying $w = uv$ with $|u| = |x|$ and $|v| = |y|$. By simplifiability of R, we have $u\,R\,x$ and $v\,R\,y$. Hence, $w = uv \in R(X)R(Y)$.

By induction, we see that $R(X)^n = R(X^n)$ for all $n \geq 0$. Thus, also the second claim follows. □

Example 1. Consider partial words introduced by Berstel and Boasson in [1]. A *partial word* of length n over an alphabet A is a partial function

$$w\colon \{1, 2, \ldots, n\} \to A.$$

The domain $D(w)$ of w is the set of positions $p \in \{1, 2, \ldots, n\}$ such that $w(p)$ is defined. The set $H(w) = \{1, 2, \ldots, n\} \setminus D(w)$ is the set of *holes* of w. To each partial word we may associate a total word $w_\diamondsuit$ over the extended alphabet $A_\diamondsuit = A \cup \{\diamondsuit\}$. This *companion* of w is defined by

$$w_\diamondsuit(p) \quad = \quad \begin{cases} w(p) & \text{if } p \in D(w), \\ \diamondsuit & \text{if } p \in H(w). \end{cases}$$

Thus, the holes are marked with the "do not know" symbol $\diamondsuit$. Clearly, partial words are in one-to-one correspondence with words over $A_\diamondsuit$.

The compatibility relation of partial words is defined as follows. Let x and y be two partial words of equal length. The word x is *contained* in y if $D(x) \subseteq D(y)$ and $x(k) = y(k)$ for all k in $D(x)$. Two partial words x and y are said to be *compatible* if there exists a partial word z such that z contains both x and y. Then we write $x \uparrow y$.

From another viewpoint partial words with compatibility relation $\uparrow$ can be seen as words over the alphabet $A_\diamondsuit$ with the relation

$$R_\uparrow = \langle \{(\diamondsuit, a) \mid a \in A\} \rangle.$$

Namely, consider two compatible partial words x and y. Let z be a partial word which contains both x and y. Suppose that their companions are $x_\diamondsuit = a_1 \cdots a_n$, $y_\diamondsuit = b_1 \cdots b_n$ and $z_\diamondsuit = c_1 \cdots c_n$. According to the definition of compatible partial words, we have four possibilities for each position $i \in \{1, 2, \ldots, n\}$:

$$\begin{array}{ll} (i) & c_i = \diamondsuit,\ a_i = b_i = \diamondsuit \\ (ii) & c_i \neq \diamondsuit,\ a_i = \diamondsuit,\ b_i = c_i \\ (iii) & c_i \neq \diamondsuit,\ b_i = \diamondsuit,\ a_i = c_i \\ (iv) & c_i \neq \diamondsuit,\ a_i = b_i = c_i. \end{array}$$

We see that in each case $a_i\,R_\uparrow\,b_i$, and thus $x_\diamondsuit\,R_\uparrow\,y_\diamondsuit$. On the other hand, for $R_\uparrow$-compatible words $x_\diamondsuit = a_1 \cdots a_n$ and $y_\diamondsuit = b_1 \cdots b_n$ we may find a word $z_\diamondsuit = c_1 \cdots c_n$ such that the corresponding partial words x and y are contained in z and therefore $x \uparrow y$. We simply choose the letter c_i in such a way that it corresponds to one of the cases $(i) - (iv)$ above. Thus, partial words are equivalent to words on alphabet $A_\diamondsuit$ with a specific relation $R_\uparrow$ and all results concerning word relations can be applied also for the compatibility relation of partial words.

3 Relational codes

Let R and S be two word relations on the monoid A^*. A subset $X \subseteq A^*$ is an (R,S)-*code* if for all $n, m \geq 1$ and $x_1, \ldots, x_m, y_1, \ldots, y_n \in X$, we have

$$x_1 \cdots x_m \, R \, y_1 \cdots y_n \implies n = m \text{ and } x_i \, S \, y_i \text{ for } i = 1, 2, \ldots, m.$$

If S is the identity relation ι, then an (R,S)-code is called a *strong R-code*, or shortly just an *R-code*. A strong R-code is always a set where the elements are pairwise incompatible, but the converse is clearly false. An (R,R)-code is called a *weak R-code*. An (ι,ι)-code is simply called a *code*. The definition coincides with the original definition of a variable length code.

Consider the partial ordering of the word relations: $R_1 \subseteq R_2$ if $u \, R_1 \, v$ implies $u \, R_2 \, v$. The following proposition manifests Galois type connections of different relational codes. This is illustrated in Figure 1.

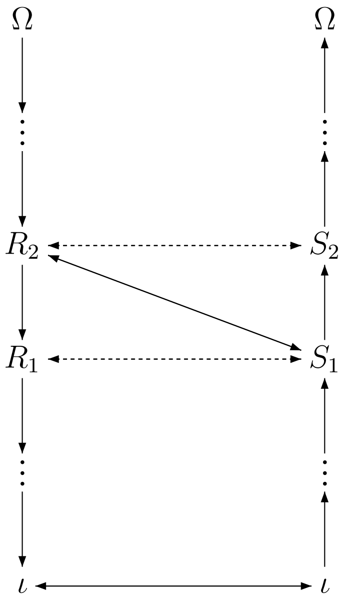


Figure 1: The Galois type connection of relational codes: an (R_2, S_1)-code is also an (R_1, S_1)-code and (R_2, S_2)-code.

Proposition 2. *(i) Let R_1, R_2 and S be word relations on A^* with $R_1 \subseteq R_2$. If X is an (R_2, S)-code, then X is an (R_1, S)-code.*

(ii) Let R, S_1 and S_2 be relations on A^* and let $S_1 \subseteq S_2$. If X is an (R, S_1)-code, then X is an (R, S_2)-code.

Proof. For (i), suppose that X is an (R_2, S)-code, and let $x_1, \ldots, x_m, y_1, \ldots, y_n \in X$ satisfy

$$x_1 \cdots x_m \, R_1 \, y_1 \cdots y_n.$$

Then also $x_1 \cdots x_m \, R_2 \, y_1 \cdots y_n$, which implies that $n = m$ and hence $x_i \, S \, y_i$ for all $i = 1, 2, \ldots, m$.

For (ii), suppose that X is an (R, S_1)-code. Let $x_1, \ldots, x_m$, $y_1, \ldots, y_n \in X$ satisfy

$$x_1 \cdots x_m \, R \, y_1 \cdots y_n.$$

Then $n = m$ and $x_i \, S_1 \, y_i$ for all $i = 1, 2, \ldots, m$, and thus $x_i \, S_2 \, y_i$ for all $i = 1, 2, \ldots, m$. □

When we consider unions and intersections of word relations the previous result implies the following corollary.

Corollary 1. *Let X be an (R_1, S_1)-code and let R_2 and S_2 be two words relations on A^*. Then X is an $(R_1 \cap R_2, S_1 \cup S_2)$-code.*

Proof. Since $R_1 \cap R_2 \subseteq R_1$, X is an $(R_1 \cap R_2, S_1)$-code by Theorem 2(i). Since $S_1 \subseteq S_1 \cup S_2$, X is an $(R_1 \cap R_2, S_1 \cup S_2)$-code by Theorem 2(ii). □

For sets that are both (R, S_1)-codes and (R, S_2)-codes the coding property can be preserved also when the S-relation is restricted to the intersection of S_1 and S_2 relations.

Proposition 3. *Let X be both an (R, S_1)-code and an (R, S_2)-code. Then it is also an $(R, S_1 \cap S_2)$-code.*

Proof. Let the words $x_1, \ldots, x_m, y_1, \ldots, y_n \in X$ satisfy $x_1 \cdots x_m \, R \, y_1 \cdots y_n$. Therefore $n = m$ and $x_i \, S_j \, y_i$ for all $i = 1, 2, \ldots, m$ and for both $j = 1$ and $j = 2$. Thus $x_i \, (S_1 \cap S_2) \, y_i$ for all $i = 1, 2, \ldots, m$, and, consequently, X is an $(R, S_1 \cap S_2)$-code. □

Note that X is not necessarily an $(R_1 \cup R_2, S)$-code even when it is both an (R_1, S_1)-code and (R_2, S_1)-code.

Example 2. Define $X = \{ab, c\}$, $R_1 = \langle \{(a, c)\} \rangle$ and $R_2 = \langle \{(b, c)\} \rangle$. Clearly, X is both an (R_1, ι)-code and an (R_2, ι)-code. Now choose $R = R_1 \cup R_2 = \langle \{(a, c), (b, c)\} \rangle$. Then we have $ab \, R \, cc$. Thus X is not an $(R_1 \cup R_2, \iota)$-code.

The next theorem shows that the (R, S)-codes are always codes in the usual meaning, but (R, S)-codes can be more restrictive. If a subset $X \subseteq A^*$ is an (R, S)-code for the relations R and S different from the identity relation, it means that the words in X^* can be uniquely factored, and, moreover, in X the relations R and S have a special order.

Theorem 1. *Every (R, S)-code X is a code such that $R_X \subseteq S_X$.*

Proof. Suppose that X is an (R, S)-code. Then X must be a code. Indeed, otherwise, there exists a nontrivial relation $x_1 \cdots x_m = y_1 \cdots y_n$ with $x_1, \ldots, x_m$, $y_1, \ldots, y_n \in X$, where we may assume that $x_1 \neq y_1$, and in particular $|x_1| \neq |y_1|$. Since X is an (R, S)-code, we have $x_1 \cdots x_m \, R \, y_1 \cdots y_n$, and hence also $x_1 \, S \, y_1$.

Since S is a word relation, this implies that $|x_1| = |y_1|$; a contradiction. The latter claim follows directly from the definition of an (R, S)-code in the case for $n = m = 1$. $\square$

Note that the converse of Theorem 1 is not true. As in the previous example, assume that $X = \{ab, c\}$. Now X is a code. Let $S = \iota$ and $R = \langle\{(a, c), (b, c)\}\rangle$. We do have $R_X \subseteq S_X$, but X is not an (R, S)-code, since $abRcc$.

By the previous theorem every (R, S)-code is an (ι, ι)-code, but we may say even more.

Theorem 2. *Every (R, S)-code is an (R, R)-code.*

Proof. Suppose that X is an (R, S)-code. By Theorem 2(ii), X is an (R, Ω)-code. This simply means that if $x_1 \cdots x_m \, R \, y_1 \cdots y_n$ with $x_i, y_j \in X$, then $m = n$ and $|x_i| = |y_i|$ for all $i = 1, 2, \ldots, m$. Then, by the simplifiability of the word relations, we have $x_i \, R \, y_i$ for all $i = 1, 2, \ldots, m$. $\square$

Theorem 2 gives another proof for Theorem 1. Namely every (R, S)-code is an (ι, S)-code by Theorem 2(i) and thus an (ι, ι)-code by the previous theorem.

Note that the roles of the relations R and S are not symmetric. Indeed, not all (R, S)-codes are (S, S)-codes. To see this, consider once again $X = \{ab, c\}$, and suppose that $R = \langle\{(a, c)\}\rangle$ and $S = \langle\{(a, c), (b, c)\}\rangle$. Now X is an (R, R)-code, but not an (S, S)-code.

Finally, we give a new characterization to relational codes using the previous results.

Theorem 3. *A subset $X \subseteq A^*$ is an (R, S)-code if and only if X is an (R, R)-code such that $R_X \subseteq S_X$.*

Proof. Suppose first that X is an (R, S)-code. Then it is also an (R, R)-code by Theorem 2 and by the definition of an (R, S)-code we have $R_X \subseteq S_X$. Conversely, let X be an (R, R)-code and $R_X \subseteq S_X$. Now consider words $x_1, \ldots, x_m$, $y_1, \ldots, y_n \in X$ satisfying $x_1 \cdots x_m \, R \, y_1 \cdots y_n$. Since X is an (R, R)-code, we have $n = m$ and $x_i \, R \, y_i$ for all $i = 1, 2, \ldots, m$. By the assumption $R_X \subseteq S_X$, we have $x_i \, S \, y_i$ for all $i = 1, 2, \ldots, m$. $\square$

4 Algorithm for relational codes

In [16] A.A. Sardinas and G.W. Patterson gave their famous algorithm for deciding whether a given set X of words is a code or not. F. Blanchet-Sadri proved in [5] that the corresponding problem for partial words is decidable. The proof seems to be quite technical compared to the case of total words. It is based on a domino technique by Head and Weber introduced in [13]. Here we give a simple algorithm for the more general problem of deciding whether a given set X is an (R, S)-code or not. The essential part of the algorithm is to solve the problem for (R, R)-codes. We use a suitable modification of the Sardinas-Patterson algorithm.

Algorithm 1. (Modified Sardinas-Patterson) *Let the input be a finite set $X \subseteq A^+$. Let $U_1 = R(X)^{-1}X \setminus \{\varepsilon\}$, and define*

$$U_{n+1} = R(X)^{-1}U_n \, \cup \, R(U_n)^{-1}X$$

for $n \geq 1$. Let $i \geq 2$ satisfy $U_i = U_{i-t}$ for some $t > 0$. Then X is a weak R-code if and only if

$$\varepsilon \notin \bigcup_{j=1}^{i-1} U_j.$$

The proof of correctness for this algorithm is modified from the proof for the Sardinas-Patterson algorithm in [2]. We need the following lemma.

Lemma 1. *Let $X \subseteq A^+$. For all $n \geq 1$ and $1 \leq k \leq n$, we have $\varepsilon \in U_n$ if and only if there exist $u \in U_k$ and integers $i, j \geq 0$ such that*

$$uX^i \cap R(X^j) \neq \emptyset \quad \textit{and } i + j + k = n. \tag{1}$$

Proof. We prove the statement for all n by descending induction on k. Assume first that $k = n$. If $\varepsilon \in U_n$, then the condition (1) is satisfied with $u = \varepsilon$ and $i = j = 0$. Conversely, if the condition is satisfied, then $i = j = 0$ and $\{u\} \cap \{\varepsilon\} \neq \emptyset$. Thus $u = \varepsilon$ and consequently $\varepsilon \in U_n$.

Now let $n > k \geq 1$ and suppose that the claim holds for $n, n-1, \ldots, k+1$. If $\varepsilon \in U_n$, then by the induction hypothesis, there exists a word $u \in U_{k+1}$ and integers $i, j \geq 0$ such that $uX^i \cap R(X^j) \neq \emptyset$ and $i + j + (k+1) = n$. Thus there exist words $x_1, \ldots, x_i, y_1, \ldots, y_j \in X$ such that

$$ux_1 \cdots x_i \, R \, y_1 \cdots y_j.$$

Now $u \in U_{k+1}$, and there are two cases: either there exists $y \in R(X)$ such that $yu \in U_k$ or there exists $v \in R(U_k)$ such that $vu \in X$. In the first case we have $y \, R \, y'$ for some $y' \in X$ and

$$yux_1 \cdots x_i \, R \, y'y_1 \cdots y_j.$$

Consequently there exist a word $yu \in U_k$ and integers $i, j+1 \geq 0$ such that $yuX^i \cap R(X^{j+1}) \neq \emptyset$ and $i + (j+1) + k = n$. In the second case there exists $v' \in U_k$ such that $v \, R \, v'$ and we have

$$vux_1 \cdots x_i \, R \, v'y_1 \cdots y_j.$$

Hence there exist a word $v' \in U_k$ and integers $i, j+1 \geq 0$ such that $v'X^j \cap R(X^{i+1}) \neq \emptyset$ and $j + (i+1) + k = n$.

Conversely, assume that there are a word $u \in U_k$ and integers $i, j \geq 0$ such that $uX^i \cap R(X^j) \neq \emptyset$ and $i + j + k = n$. Then

$$ux_1 \cdots x_i \, R \, y_1 \cdots y_j$$

for some $x_1, \ldots, x_i, y_1, \ldots, y_j \in X$. If $j = 0$, then $i = 0$, $k = n$ and $u = \varepsilon$. If $j > 0$, then we consider two cases:

Case 1: Assume that $|u| \geq |y_1|$. We write $u = y_1' v$, where $y_1' \, R \, y_1$ and $v \in A^*$. Then $v \in U_{k+1}$ and $vx_1 \cdots x_i \, R \, y_2 \cdot v_j$. Thus $vX^i \cap R(X^{j-1}) \neq \emptyset$ and $i + (j - 1) + (k + 1) = n$. By the induction hypothesis, $\varepsilon \in U_n$.

Case 2: Assume that $|u| < |y_1|$. We write $y_1 = u'v$, where $u' \, R \, u$ and $v \in A^+$. Then $v \in U_{k+1}$ and $x_1 \cdots x_i \, R \, vv_2 \cdot v_j$. Thus $vX^{j-1} \cap R(X^i) \neq \emptyset$ and $(j - 1) + i + (k + 1) = n$. Thus again $\varepsilon \in U_n$ by the induction hypothesis. □

Theorem 4. *The set X is a weak R-code if and only if none of the sets U_n contains the empty word.*

Proof. If X is not a weak R-code, then there exist positive integers m and n and words $x_1, \ldots, x_m, y_1, \ldots, y_n \in X$ such that

$$x_1 \cdots x_m \, R \, y_1 \cdots y_n \quad \text{and not } x_1 \, R \, y_1$$

from which it follows that $|x_1| \neq |y_1|$. By symmetry we may assume that $|x_1| > |y_1|$, i.e., $x_1 = y_1' u$ for some $u \in A^+$ and $y_1' \, R \, y_1$. Now $u \in U_1$ and $uX^{m-1} \cap R(X^{n-1}) \neq \emptyset$. According to Lemma 1 we have $\varepsilon \in U_{m+n-1}$.

Conversely, if $\varepsilon \in U_n$, then choose $k = 1$ in Lemma 1. Hence, there exist a word $u \in U_1$ and integers $i, j \geq 0$ such that $i + j = n - 1$ and $uX^i \cap R(X^j) \neq \emptyset$. Since $u \in U_1$, we have $y = xu$ for some $x \in R(X)$ and $y \in X$. Furthermore, $x \neq y$, since $u \neq \varepsilon$ by the definition $U_1 = R(X)^{-1} X \setminus \{\varepsilon\}$. Since $x \in R(X)$, there exists $x' \in X$ such that $x \, R \, x'$. It follows from $xuX^i \cap xR(X^j) \neq \emptyset$ that $yX^i \cap R(x'X^j) \neq \emptyset$. This means that X is not a weak R-code. □

Note that there exist only finitely many different sets U_n, since all the lengths of the elements of U_n are less than $\max\{|x| \mid x \in X\}$. Secondly, if $U_i = U_j$ then, for any $t \geq 0$, $U_{i+t} = U_{j+t}$. Thus once a repetition in the sequence $U_1, U_2, \ldots$ is found, all U_i sets are found as well. Now it is clear by the previous theorem and Theorem 3 that the (R, S)-coding property of a finite subset X of A^* can be verified using the following algorithm.

Algorithm 2. (Test for relational codes) *Let the input be a finite set $X \subseteq A^+$ and two word relations R and S.*

1. *Determine whether X is a weak R-code by Algorithm 1*
2. *If X is a weak R-code then check whether $R_X \subseteq S_X$. If $R_X \subseteq S_X$, then X is an (R, S)-code; otherwise, it is not.*

5 Minimal and maximal relations

Let X be a subset of A^*. We define minimal and maximal relations with respect to X as follows. Let $S_{\min}(X, R)$ be the set of the relations S such that X is an

(R, S)-code, and for all S' with $S' \subset S$, X is not an (R, S')-code. Similarly, let $S_{\max}(X, R)$ be the set of relations S such that X is an (R, S)-code, and for all S' with $S \subset S'$, X is not an (R, S')-code. Relations belonging to $S_{\min}(X, R)$ (resp. $S_{\max}(X, R)$) are called *minimal* (resp. *maximal*) S-relations with respect to a set X and a relation R.

Symmetrically, let $R_{\min}(X, S)$ be the set of the relations R such that X is an (R, S)-code, and for all relations $R' \subset R$, X is not an (R', S)-code. Also, let $R_{\max}(X, S))$ be the set of the relations R such that X is an (R, S)-code, and for all relations $R \subset R'$, X is not an (R', S)-code. Relations belonging to $R_{\min}(X, S)$ (resp. $R_{\max}(X, S)$) are called *minimal* (resp. *maximal*) R-relations with respect to a set X and a relation S.

We make a few easy observations.

Theorem 5. *The minimal and maximal relations have the following properties: Let $X \subseteq A^*$.*

(i) X is not a code if and only if, for all word relations R and S, we have $S_{\min}(X, R) = S_{\max}(X, R) = R_{\min}(X, S) = R_{\max}(X, S) = \emptyset$.

(ii) X is not an (R, R)-code if and only if $S_{\min}(X, R) = S_{\max}(X, R) = \emptyset$.

(iii) For all (R, R)-codes X, $S_{\min}(X, R)$ has a unique element.

(iv) For all (R, R)-codes, $S_{\max}(X, R) = \{\Omega\}$.

(v) For all codes, $R_{\min}(X, S) = \{\iota\}$.

(vi) If $S_1 \subset S_2$, then for all $R \in R_{\max}(X, S_1)$ there exists $R' \in R_{\max}(X, S_2)$ such that $R \subseteq R'$.

Proof. (i): By Theorem 1, every (R, S)-code is a code. Thus, for noncodes, there does not exist any maximal and minimal relations. On the other hand, if X is a code, then at least for $R = S = \iota$, the maximal and minimal relations are nonempty.

(ii): By Theorem 2, every (R, S)-code is a weak R-code. Hence, if X is not an (R, R)-code, then no maximal or minimal S-relations with respect to X and R exist. Conversely, if X is an (R, R)-code, then $S_{\min}(X, R)$ and $S_{\max}(X, R)$ are trivially nonempty.

(iii): Let **S** be the set of all relations S such that X is an (R, S)-code. By Theorem 2(ii), the unique minimal S-relation with respect to X and R is the intersection of all $S \in \mathbf{S}$.

(iv): Follows directly from Theorem 2(ii).

(v): Follows directly from Theorem 2(i).

(vi): Let $S_1 \subset S_2$ and let R belong to $R_{\max}(X, S_1)$. By Theorem 2(ii), X is also an (R, S_2)-code. Hence either R is maximal with respect to S_2 or $R \subset R'$ for some maximal R' with respect to S_2. □

Note that there may be several maximal relations belonging to $R_{\max}(X,S)$, though, by Theorem 5(ii) and (iii) $S_{\min}(X,R)$ always is a unique relation. For example, in our Example 2 both relations R_1 and R_2 are maximal. With respect to X these two word relations seem to have symmetric roles and they have the same number of pairs of letters in the corresponding relation on A. This need not be the case in general. A more complicated case can be seen later in Example 3.

The coding properties of an (R,S)-code X can be measured by defining the maximal and minimal relations R and S. Next we will present two algorithms for this purpose.

Algorithm 3. (X restriction) *Let the input be a finite set $X \subseteq A^*$ and a word relation R.*

1. *Set $S = \iota$.*

2. *Find $R_X = \{(x,y) \in X \times X \mid x\,R\,y\}$.*

3. *For all $m \geq 1$ and for each pair of words $x = a_1 \cdots a_m$ and $y = b_1 \cdots b_m$ in R_X set $S \leftarrow S \cup \langle\{(a_i,b_i)\}\rangle$ for $i = 1,2,\ldots,m$.*

The previous algorithm can be used to find the minimal S relation with respect to X and R.

Theorem 6. *Let X be a finite (R,R)-code. The relation S obtained in Algorithm 3 is $S_{\min}(X,R)$.*

Proof. Since X is an (R,R)-code, the unique minimal element S' belonging to $S_{\min}(X,R)$ must be a subset of R. By Theorem 3, we have $R_X \subseteq S'_X$, and thus $R_X = S_X$. (Note that this does not mean that $S' = R$, since in R there may be pairs of letters which never occur in any compatible words of X.)

On the other hand, the algorithm ensures that for all $x,y \in X$ the relation $x\,R\,y$ implies $x\,S\,y$, i.e, $R_X \subseteq S_X$. Also, the relation S is minimal. Indeed, if we omit any pair (a,b) with $a \neq b$ from S, then for some words $x,y \in X$ with $x\,R\,y$, we would have $(x,y) \notin S$. □

Finding the maximal R-relations in $R_{\max}(X,S)$ is a more complicated task. By Theorem 3 there are two properties that restrict the maximal R relations. Namely we must have $R_X \subseteq S_X$, but at the same time X must be a weak R-code. We do not know which one of these conditions is more restrictive.

In order to present the algorithm we define a new total order on word relations. It is based on two orders of words in A^*. Assume that the alphabet A is totally ordered by $\prec$, i.e., for each two letters $a \neq b$ either $a \prec b$ or $b \prec a$. Denote the prefix of a word w of length n by $\mathrm{pref}_n(w)$. The *maximal common prefix* of words u and v is denoted by $u \wedge v$. The total order $\prec$ of A is extended to *lexicographic ordering* $\prec_l$ and *alphabetic ordering* $\prec_a$ of A^* by defining

$$u \prec_l v \iff u^{-1}v \in A^+ \quad \text{or} \quad \mathrm{pref}_1((u \wedge v)^{-1}u) \prec \mathrm{pref}_1((u \wedge v)^{-1}v)$$

and

$$u \prec_a v \iff |u| < |v| \text{ or } |u| = |v| \text{ and } u \prec_l v.$$

We use this ordering to define an ordering of pairs of letters. Define a function $\varphi\colon A \times A \to A^2$ by letting $\varphi((a,b)) = ab$. For two pairs of letters (a,b) and (c,d), we define

$$(a,b) \prec (c,d) \iff \varphi((a,b)) \prec_l \varphi((c,d)).$$

This, in turn, induces a total order on the compatibility relations on the letters. Let R_1 and R_2 be two compatibility relations on A. Let r_1 be the catenation of the words $\varphi((a,b))$ for all pairs $(a,b) \in R_1$ in the lexicographic order. Let r_2 be the corresponding word for the relation R_2. Then

$$R_1 \prec R_2 \iff r_1 \prec_a r_2.$$

Now we are ready to present to desired algorithm.

Algorithm 4. (Maximal R) *Let the input be a finite set $X \subseteq A^+$ and a word relation S.*

1. *Construct a directed graph of relations $G = (V, E)$ such that the set of vertices V is the set of all compatibility relations on A and the set of edges is defined by*

$$E = \{(R_1, R_2) \mid R_1 \subseteq R_2 \textit{ and } |R_2 \setminus R_1| = 1\}.$$

2. *Run through all the vertices in the order $\prec$ of the relations $R \in V$. For each vertex calculate whether X is an (R,S)-code or not using Algorithm 2. If the answer is negative, then modify G by deleting the corresponding vertex R, all the vertices R' such that there is a path from R to R' and all related edges.*

3. *Set $R_{\max}(X,S)$ to be the set of all the vertices R with no edges starting from R.*

Theorem 7. *Let X be a finite subset of A^+ and S a word relation on A. Algorithm 4 finds all the relations R belonging to $R_{\max}(X,S)$.*

Proof. For each compatibility relations on A the algorithm decides whether X is an (R,S)-code or not. This is done either by using Algorithm 2 or deleting the corresponding vertex according to previous calculations. Namely, if X is not an (R,S)-code for a relation R, then X is not an (R',S)-code for all the relations R' with $R \subseteq R'$ by Theorem 2(i). This justifies the modifications of the directed graph in the step 2. The edges describe the order of the vertices and corresponding relations. Thus, after deleting all vertices corresponding to noncodes, the remaining vertices with no outgoing edges must correspond to maximal relations. □

The following example shows how the algorithm works in a four letter alphabet $\{a, b, c, d\}$.

Example 3. Let $X = \{ab, bccb, ca\}$ and $S = \langle\{(a, b), (a, c)\}\rangle$. The directed graph of step 1 is illustrated in Figure 2. It is clear that X is a code, since it is even a prefix code. With all the R relations with one generator the set X is also an (R, S)-code. We notice that two first letters of each of the words in X differ from the letters in the same position in other words. This means that at least two relations are needed in order to achieve two different compatible words. In the case $R = \langle\{(a, b), (a, c)\}\rangle$ the two compatible words ab and ca are valid since $S = R$. Since the pair (b, c) is missing, all the words in X^+ compatible with a word beginning with $bccb$ begin also with the same word. Thus X must be an (R, S)-code. In the other cases with the generator set consisting of two elements we have nontrivial compatibility relations such as $bccb\,R\,ab.ca$ and $bccb\,R\,ca.ab$. Thus Algorithm 2 gives a negative answer and these vertices are deleted. Consequently, also the vertex Ω is deleted. The deleted vertices are marked with a double circle in Figure 2. Hence in step 3 we set $R_{\max}(X, S) = \{\langle(b, c)\rangle, \langle\{(a, b), (a, c)\}\rangle\}$. Note that these two maximal R relations are by no means isomorphic. They do not even have the same size.

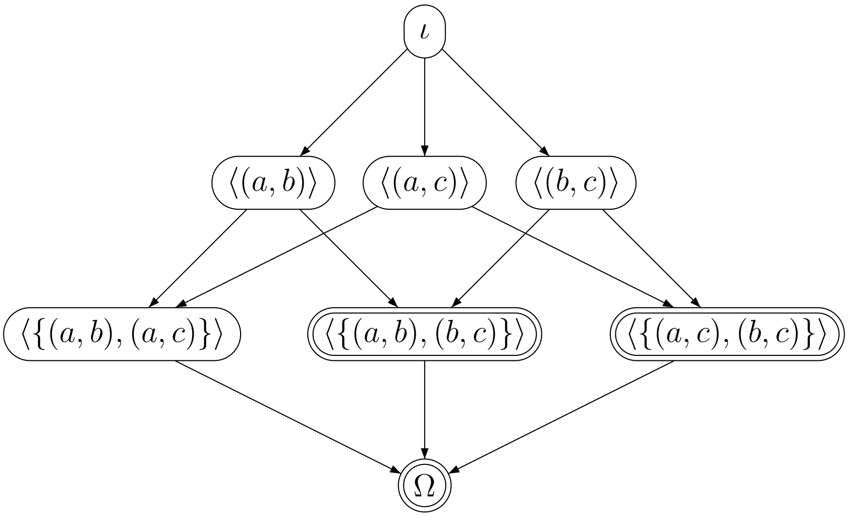

ι
$\langle(a, b)\rangle$
$\langle(a, c)\rangle$
$\langle(b, c)\rangle$
$\langle\{(a, b), (a, c)\}\rangle$
$\langle\{(a, b), (b, c)\}\rangle$
$\langle\{(a, c), (b, c)\}\rangle$
Ω

Figure 2: The graphs G of the example 3

We consider briefly the complexity of Algorithm 4. Let us first suppose that the alphabet is fixed. Then the construction of the graph G takes a fixed number of operations. Similarly running through the graph can be done in a fixed time. Thus the complexity of our algorithm is just a constant times the complexity of Algorithm 2. This is essentially same as the complexity of the Sardinas-Patterson algorithm. It is not clear that Algorithm 1 in the form it was presented terminates in a polynomial time compared to the size of the input, but using a construction in [11] it is easy to show that this test for unique deciphering can be done in time $\mathcal{O}(n^2)$, where n is the sum of the lengths of all words in the input set X. Actually the test can be done even in time $\mathcal{O}(nm)$, where m is the number of words in X; see also [15]. Thus finding all the maximal elements R with respect to a given set X and a given word relation S can be done in polynomial time.

From another viewpoint, i.e., if we allow arbitrary alphabets, the problem of finding maximal R relations is actually very difficult. The corresponding decision problem is namely *NP-complete;* for more on NP-complete problems see [12]. Define the size of a word relation to be the number of pairs in the corresponding compatibility relation of letters. Let us denote this number by sz(R). Define the number $M_R(X,S)$ to be the maximal size of the relations in $R_{\max}(X,S)$, i.e., $M_R(X,S) = \max\{\text{sz}(R) \mid R \in R_{\max}(X,S)\}$. We formulate the following problem:

Problem:	MAXIMAL RELATION
Instance:	A set $X \subseteq A^+$, a relation S on A and a positive integer k
Question:	Is $M_R(X,S) \geq k$?

The problem above is related to the following problem of graphs. Let $G = (V,E)$ be a graph. A set $W \subseteq V$ is a *vertex cover* of G if for each edge $(u,v) \in E$ at least one of u and v belongs to W. The *cover number* c(G) of a graph G is the minimal cardinality of a vertex cover in G.

Problem:	VERTEX COVER
Instance:	A graph $G = (V,E)$ and a positive integer k
Question:	Is c$(G) \leq k$?

This problem is known to be NP-complete. A proof can be found in [12]. Next we will show how to reduce this problem to the problem MAXIMAL RELATION.

Theorem 8. *The problem MAXIMAL RELATION is NP-complete.*

Proof. First we must show that MAXIMAL RELATION $\in \mathbf{NP}$. This is clear since, for a set $X \subseteq A^*$, a positive integer k, a relation S on A and an arbitrary relation R on A with sz$(R) \geq k$, we can verify in polynomial time whether X is an (R,S)-code. If the answer is positive, then clearly $M_R(X,S) \geq k$.

Secondly our aim is to prove that the NP-complete problem VERTEX COVER can be polynomially reduced to the problem MAXIMAL RELATION, i.e., solving the latter problem gives an answer also to the first problem. More formally, it means that any input x of the problem VERTEX COVER can be turned into an input $f(x)$ of MAXIMAL RELATION in polynomial time and $f(x)$ is a "yes" instance of MAXIMAL RELATION if and only if x is a "yes" instance of VERTEX COVER.

Next we define the function f which maps a pair (G,k) to a triplet (X,S,j) in a following way. Assume that the graph $G = (V,E)$ has vertices $V = \{v_1,\ldots,v_n\}$ and edges $E = \{e_1,\ldots,e_m\}$. We may assume that the graph G has no isolated vertices, i.e., vertices of degree zero, since they are not considered in VERTEX COVER problem. For each edge $e_i = (v_{i_1}, v_{i_2})$ we define two words $iv_{i_1}v_{i_2}$ and iaa. Let X consist of all these words for every $i = 1, 2, \ldots, m$. We also choose $S = \iota$ and

$$j = |A|^2 - 2k - (m^2 - m)$$

for the alphabet

$$A = \{1, 2, \ldots, m\} \cup \{a\} \cup \{v_1, \ldots, v_n\}.$$

Thus it has the cardinality $m + n + 1$. Denote by $\|X\|$ the sum of the lengths of all words in X. Clearly $|X| = 2m$ and since all the words are of length 3 we have $\|X\| = 6m$. Thus this construction is polynomial.

Now suppose that W is a vertex cover of G and $|W| \leq k$. We show that there is a relation R with sz$(R) \geq j$ such that X is an (R, ι)-code. First define

$$T = \{(i, j) \mid i, j \in \{1, 2, \ldots, m\}, i \neq j\}$$

and

$$U = \{(a, v) \mid v \in W\} \cup \{(v, a) \mid v \in W\}.$$

Now let us choose

$$R = \Omega \setminus \{T \cup U\}.$$

This relation satisfies sz$(R) = |A|^2 - 2|W| - (m^2 - m) \geq j$ by the assumption $|W| \leq k$. Now consider all possible pairs of words in $X \times X$. All the words beginning with a different letter $i = 1, 2, \ldots, m$ cannot be compatible by the definition of T. Thus we have to compare only words starting with the same letter. For each $i = 1, 2, \ldots, m$ there is only one such pair, namely (iv_1v_2, iaa). Now since W is a vertex cover at least one of v_1 and v_2 belong to W. Thus at least on of the relations (a, v_1) and (a, v_2) is in U. This means that all the words of X are R-compatible only with themselves. Hence, X is an (R, ι)-code with sz$(R) \geq j$.

Conversely, suppose that there is a relation R of size sz$(R) \geq j$ such that X is an (R, ι)-code. Define

$$W = \{v \in V \mid \ (a, v) \text{ is not in } R\}.$$

Clearly we have to deny all the relations in T. Otherwise, $iaa\, R\, jaa$ for two different i and j in $\{1, 2, \ldots, m\}$ and the coding property is not valid. Thus, all the possible compatible pairs of words start with a same letter $i = 1, 2, \ldots, m$. For each i we have a unique pair of words, namely (iv_1v_2, iaa). Since X is an (R, ι)-code we have to have $(iaa, iv_1v_2) \notin R$. Thus at least one of the relations (a, v_1) and (a, v_2) is not in R. This implies that at least one of the vertices v_1 and v_2 is in W and W is really a vertex cover of G. The number of letter pairs not belonging to R is less than or equal to $|A|^2 - \text{sz}(R) = 2k + (m^2 - m)$. By taking into account that W is not based on $m^2 - m$ pairs in $T \subseteq (A \times A) \setminus R$ and R is symmetric we conclude that $|W| \leq k$. □

References

[1] J. Berstel, L. Boasson, Partial words and a theorem of Fine and Wilf, Theoret. Comput. Sci. 218 (1999) 135–141.

[2] J. Berstel, D. Perrin, Theory of Codes, Academic press, New York, 1985.

[3] F. Blanchet-Sadri, A Periodicity Result of Partial Words with One Hole, Comput. Math. Appl. 46 (2003) 813–820.

[4] F. Blanchet-Sadri, Periodicity on partial words, Comput. Math. Appl. 47 (2004) 71–82.

[5] F. Blanchet-Sadri, Codes, orderings, and partial words, Theoret. Comput. Sci. 329 (2004) 177–202.

[6] F. Blanchet-Sadri, Primitive Partial Words, Discrete Appl. Math. 148 (2005) 195–213.

[7] F. Blanchet-Sadri, A. Chriscoe, Local periods and binary partial words: an algorithm, Theoret. Comput. Sci. 314 (2004) 189–216.

[8] F. Blanchet-Sadri, S. Duncan, Partial words and the critical factorization theorem, J. Combin. Theory, Ser. A 109 (2005) 221–245.

[9] F. Blanchet-Sadri, R.A. Hegstrom, Partial words and a theorem of Fine and Wilf revisited, Theoret. Comput. Sci. 270 (2002) 401–419.

[10] F. Blanchet-Sadri, D.K. Luhmann, Conjugacy on partial words, Theoret. Comput. Sci. 289 (2002) 297–312.

[11] M. Crochemore, W. Rytter, Jewels of Stringology, World Scientific Publishing, 2002.

[12] M.R. Garey, D.S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.

[13] T. Head, A. Weber, Deciding multiset decipherability, IEEE Trans. Inform. Theory 41 (1995) 291–297.

[14] P. Leupold, Partial words for DNA coding, Lecture Notes in Comput. Sci. 3384 (2005) 224–234.

[15] M. Rodeh, A fast test for unique decipherability based on suffix trees, IEEE Trans. Inform. Theory, vol. IT-28, no. 4 (1982), 648–651.

[16] A.A. Sardinas, G.W. Patterson, A necessary and sufficient condition for the unique decomposition of coded messages, IRE Internat. Conv. Rec. 8 (1953) 104–108.

TURKU
CENTRE *for*
COMPUTER
SCIENCE

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku

- Department of Information Technology
- Department of Mathematics

Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration

- Institute of Information Systems Sciences

ISBN 952-12-1721-9
ISSN 1239-1891