Theoretical Computer Science 601 (2015) 2–14

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Periodicity forcing words ☆,☆☆

Joel D. Day [a,*], Daniel Reidenbach [a,*], Johannes C. Schneider [b,*]

[a] Department of Computer Science, Loughborough University, Loughborough Leicestershire, LE11 3TU, UK
[b] DIaLOGIKa GmbH, Pascalschacht 1, 66125 Saarbrücken, Germany

ARTICLE INFO

Article history:
Available online 28 August 2015

Keywords:
Equality sets
Morphisms
Dual post correspondence problem
Periodicity forcing sets
Periodicity forcing words
Ambiguity of morphisms

ABSTRACT

The Dual Post Correspondence Problem asks, for a given word α, if there exists a non-periodic morphism g and an arbitrary morphism h such that $g(\alpha) = h(\alpha)$. Thus α satisfies the Dual PCP if and only if it belongs to a non-trivial equality set. Words which do not satisfy the Dual PCP are called *periodicity forcing*, and are important to the study of word equations, equality sets and ambiguity of morphisms. In this paper, a 'prime' subset of periodicity forcing words is presented. It is shown that when combined with a particular type of morphism it generates exactly the full set of periodicity forcing words. Furthermore, it is shown that there exist examples of periodicity forcing words which contain any given factor/prefix/suffix. Finally, an alternative class of mechanisms for generating periodicity forcing words is developed, resulting in a class of examples which contrast those known already.

1. Introduction

The Dual Post Correspondence Problem (Dual PCP) is a decidable variation of the famous Post Correspondence Problem (see Post [12]). It was introduced by Culik II and Karhumäki in [1], where the authors make progress towards a characterisation of binary equality sets. A word is said to satisfy the Dual PCP if it belongs to an equality set $\mathrm{E}(g, h)$ for two morphisms g, h where at least one morphism is non-periodic. For example, the word $\mathtt{abba}$ belongs to $\mathrm{E}(g, h)$ where $g, h : \{\mathtt{a}, \mathtt{b}\}^* \to \{\mathtt{a}, \mathtt{b}\}^*$ are the morphisms given by:

$$g(x) := \begin{cases} \mathtt{aba} & \text{if } x = \mathtt{a}, \\ \mathtt{b} & \text{if } x = \mathtt{b}, \end{cases} \quad \text{and} \quad h(x) := \begin{cases} \mathtt{a} & \text{if } x = \mathtt{a}, \\ \mathtt{bab} & \text{if } x = \mathtt{b}. \end{cases}$$

Thus $\mathtt{abba}$ satisfies the Dual PCP; in other words, it is a non-trivial *equality word*. In contrast, the word $\mathtt{abaab}$ does not satisfy the Dual PCP, but this claim is much harder to verify. The latter is called a *periodicity forcing word* since it forces each pair of morphisms which agree on it to be periodic.

Identifying which words belong to non-trivial equality sets and which do not is of immediate significance to the Post Correspondence Problem, which is simply the emptiness problem for equality sets. It is well known that although the PCP is undecidable in general, it is decidable even in polynomial time in the binary case (see Halava and Holub [6]). It is therefore no surprise that, for binary words, the Dual PCP is relatively well understood.

☆ This work was supported by the London Mathematical Society, grant SC7-1112-02.
☆☆ A preliminary version [4] of this work was presented at the conference WORDS 2013.
* Corresponding authors.
E-mail addresses: J.Day@lboro.ac.uk (J.D. Day), D.Reidenbach@lboro.ac.uk (D. Reidenbach), johannes.schneider@dialogika.de (J.C. Schneider).

http://dx.doi.org/10.1016/j.tcs.2015.08.033
0304-3975/© 2015 Published by Elsevier B.V.

This is due to both the original research by Culik II and Karhumäki [1], and from results on equality sets (e.g., Holub [7], Hadravova and Holub [5]) and word equations (e.g., Czeizler et al. [2], Karhumäki and Petre [9]). Much less, however, is known about the Dual PCP for larger alphabets.

One reason for this is that although the Dual PCP is known to be decidable, the proof (given by Culik II and Karhumäki [1]) relies on Makanin's algorithm for solving word equations [11]. While this algorithm demonstrates that the problem is computable in principle, the complexity is extremely high, and it provides little insight into the nature of words which do/do not satisfy the Dual PCP. It is worth noting that the decidability of the PCP for alphabet sizes 3 to 6 is a long-standing open problem, and therefore equality words over these alphabets are of particular interest.

In the present paper, we investigate the Dual PCP in the general case, specifically looking at periodicity forcing words. While examples of equality words are easily found, deciding on whether a word is periodicity forcing can be a particularly intricate task, and becomes even more so as the alphabet size increases. In [3], we overcome this problem by employing the use of morphisms to generate periodicity forcing words over arbitrary alphabets. Since it can be shown that many simple morphisms (such as $\varphi : \{\mathtt{a}, \mathtt{b}\}^* \to \{\mathtt{a}, \mathtt{b}\}^*$ given by $\varphi(\mathtt{a}) := \mathtt{a}$ and $\varphi(\mathtt{b}) := \mathtt{ab}$) preserve the property of being periodicity forcing, it is possible to span large parts of the set of periodicity forcing words (denoted by $\mathrm{DPCP}^{\neg}$) by applying such morphisms to existing examples.

In Section 3 of the present paper, we explore this phenomenon further. Specifically, $\mathrm{DPCP}^{\neg}$ is divided into those words which may be reached by a non-trivial morphism from other elements of the set, and those which cannot. The latter form a 'prime' subset of $\mathrm{DPCP}^{\neg}$ from which all periodicity forcing words may be generated using a specific class of morphisms characterised in [3]. In order to find examples of these prime words – therefore demonstrating that the subset is non-empty – it makes sense to consider the shortest periodicity forcing words. Thus, we also give bounds on the length of the shortest periodicity forcing words for any alphabet.

In Section 4, it is shown that there exist periodicity forcing words with arbitrary factors. This not only further demonstrates the complexity of the Dual PCP, but also provides another large, previously unknown class of periodicity forcing words and with it, further insight into their structure.

Finally, motivated by Section 3, we employ some alternative techniques for finding periodicity forcing words over larger alphabets, yielding insights into the set of 'prime' words.

2. Notation and preliminary results

An alphabet Σ is a set of symbols, or *letters*. A word over Σ is a concatenation of symbols from Σ. The empty word consisting of no symbols is ε. We denote by Σ^* the set of all words over Σ (including ε). Σ^+ is $\Sigma^*\backslash\{\varepsilon\}$. Let Σ be an alphabet. Let $u, v \in \Sigma^*$. Then v is a *factor* of u if there exist $w_1, w_2 \in \Sigma^*$ such that $u = w_1 v w_2$. A word $u \in \Sigma^*$ is *primitive* if $u = v^n$ for some $v \in \Sigma^*$ implies $n = 1$, otherwise u is *imprimitive*. If $u = v^n$ for some $n \in \mathbb{N}$ and v is primitive, then v is a *primitive root* of u; it is unique if and only if $u \neq \varepsilon$. Two words $u, v \in \Sigma^*$ *commute* if $uv = vu$. More generally, a set of words $\{u_1, u_2, \cdots, u_n\}$ commutes if for every i, j, $u_i u_j = u_j u_i$. For a set X, the notation $|X|$ refers to the cardinality of X, and for a word u, $|u|$ stands for the length of u. By $|u|_{\mathtt{a}}$, we denote the number of occurrences of the letter a in the word u. Let $u \in \{\mathtt{a}_1, \mathtt{a}_2, \cdots, \mathtt{a}_n\}^*$ be a word. The *Parikh vector of* u, denoted by $\mathrm{P}(u)$, is the vector $(|u|_{\mathtt{a}_1}, |u|_{\mathtt{a}_2}, \cdots, |u|_{\mathtt{a}_n})$. The result of dividing the Parikh vector by the greatest common divisor of its components is called the *basic Parikh vector*. A word $u \in \Sigma^*$ is *ratio-imprimitive* if there exist $v, w \in \Sigma^*$ such that $u = vw$ and v, w have the same basic Parikh vector. Otherwise u is ratio-primitive.

Let $\mathbb{N} := \{1, 2, \cdots\}$ be the set of natural numbers, and let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. We often use $\mathbb{N}$ as an infinite alphabet of symbols. In order to distinguish between a word over $\mathbb{N}$ and a word over a (possibly finite) alphabet Σ, we call the former a *pattern*. Given a pattern $\alpha \in \mathbb{N}^*$, we call symbols occurring in α *variables* and denote the set of variables in α by $\mathrm{var}(\alpha)$. Hence, $\mathrm{var}(\alpha) \subseteq \mathbb{N}$. Sometimes, for convenience, we will also use $\{x_1, x_2, \cdots\}$ to denote (possibly unknown) variables in $\mathbb{N}$. We use the symbol $\cdot$ to separate the variables in a pattern, so that, for instance, $1 \cdot 1 \cdot 2$ is not confused with $11 \cdot 2$. Given patterns α and α', if α' may be obtained from α by deleting all occurrences of some variables in α, then α' is a *subpattern* of α. If $\mathrm{var}(\alpha) = \{1, 2, \cdots, n\}$ and the leftmost occurrence of each variable $x \in \mathbb{N}$ appears to the left of any variable y with $y > x$, then α is in *canonical form*.

Given arbitrary alphabets $\mathcal{A}$, $\mathcal{B}$, a *morphism* is a mapping $h : \mathcal{A}^* \to \mathcal{B}^*$ that is compatible with concatenation, i.e., for all $v, w \in \mathcal{A}^*$, $h(vw) = h(v)h(w)$. Hence, h is fully defined for all $v \in \mathcal{A}^*$ as soon as it is defined for all symbols in $\mathcal{A}$. A morphism h is called *periodic* if and only if there exists a $v \in \mathcal{B}^*$ such that $h(a) \in \{v\}^*$ for every $a \in \mathcal{A}$. The morphisms $g, h : \mathcal{A}^* \to \mathcal{B}^*$ are *distinct* if and only if there exists an $a \in \mathcal{A}$ such that $g(a) \neq h(a)$. For the *composition* of two morphisms $g, h : \mathcal{A}^* \to \mathcal{A}^*$, we write $g \circ h$, i.e., for every $w \in \mathcal{A}^*$, $g \circ h(w) = g(h(w))$. If $g(v) = h(v)$ for some $v \in \mathcal{A}^+$, then g and h *agree* on v. The set of all words on which g and h agree is called the *equality set* of g and h. A morphism $g : \mathcal{A}^* \to \mathcal{B}^*$ is called a *renaming morphism* if it is injective, and $|g(a)| = 1$ for every $a \in \mathcal{A}$. For words $u, v \in \mathcal{A}^+$, if there exists a renaming morphism g such that $v = g(u)$, then v is simply said to be a *renaming* of u.

Two words $u \in \mathcal{A}^+$, $v \in \mathcal{B}^+$ are *morphically coincident* if there exist morphisms $g : \mathcal{A}^* \to \mathcal{B}^*$ and $h : \mathcal{B}^* \to \mathcal{A}^*$ such that $g(u) = v$ and $h(v) = u$. A pattern $\alpha \in \mathbb{N}^+$ is *morphically imprimitive* if it is morphically coincident to some pattern β with $|\beta| < |\alpha|$. Otherwise α is *morphically primitive*. It is shown in [13] that if two patterns are morphically coincident, then they are either renamings of each other, or at least one is morphically imprimitive.