DOMINOES OVER A FREE MONOID

Karel Culik II
Dept. of Computer Science, Univ. of Waterloo
Waterloo, Canada

Tero Harju
Dept. of Mathematics, Univ. of Turku
Turku, Finland

June, 1980

DOMINOES OVER A FREE MONOID*

Karel Culik II
Dept. of Computer Science, Univ. of Waterloo
Waterloo, Canada

Tero Harju+
Dept of Mathematics, Univ. of Turku
Turku, Finland

Abstract. Dominoes over a free monoid and operations on them are introduced. Related algebraic systems and their applications to decidability problems about morphisms on free monoids are studied. A new simple algorithm for testing DOL sequence equivalence is presented.

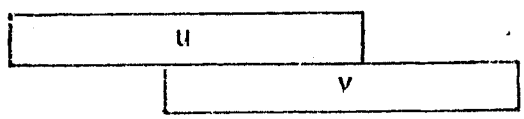
* This work was supported by the Natural Science and Engineering Research Council of Canada under Grant No. A-7403.

+ This research was done during the second author's stay at the University of Waterloo.

1. Introduction

Decidability and other problems concerning morphisms on free monoids have been extensively studied recently, see survey [2]. Here we introduce the notion of a domino which seems to be a very useful tool in this area. It is motivated mainly by the problem of testing the (string by string) equivalence of two morphisms on a given set of words [4]. A number of other problems reduces to such a test, for example, the DOL sequence equivalence problem [3] or equivalence problem for various types of transducers [1].

Intuitively, for any strings u,v in Σ^* we will call the construct



a domino over Σ^*, providing that the overlapping portions of u and v are identical. For a precise definition see section 3. The strings u and v are called the components of the domino. They might not overlap at all or they might overlap completely as in the following example, which indicates the application of dominoes to testing of equality of morphisms on certain strings.

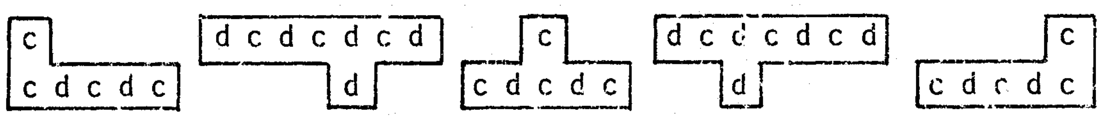
When two morphisms g and h on Σ^* are equal on a string w in Σ^*, i.e. when $g(w) = h(w)$, they might not agree on substrings of w. Consider morphisms g and h on $\{a,b\}^*$ given by

$g(a) = c$, $g(b) = dcdcdcd$, $h(a) = cdcdc$ and $h(b) = d$.

We have $g(ababa) = h(ababa)$ but g and h differ on every substring of ababa. Here one of the dominoes with components g(ababa) and h(ababa) is

c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c
c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c

Such domino is called an equal domino. However, any decomposition of the string ababa in substrings yields a decomposition of the above domino into nonequal dominoes, for example the decomposition of ababa into single letters yields dominoes:



We notice that there are three distinct dominoes with the same components g(a) and h(a) and two distinct dominoes with components g(b) and h(b).

In the following two sections we formally introduce the notion of a domino, some operations on dominoes and corresponding algebraic systems. We will study their properties and also some problems concerning sets of dominoes.

In the last section we present a domino algorithm for testing DOL sequence equivalence. It is far the simplest such algorithm known to us. However, it should be stressed that we are not giving a new proof of the decidability of the DOL sequence equivalence problem. When proving that our algorithm terminates we are using the crucial property of DOL systems, namely that every two sequence equivalent DOL systems have so called bounded balance. This property has been shown for normal systems in [3] and extended to all DOL systems in [5].

2. Preliminaries

We use some elementary notions of formal language theory, we refer the reader to [7]. In the last section the theory of dominoes is applied to testing of the DOL sequence equivalence. We will remind you of the notions from [3] which we use, see also [6].

DOL system is a construct $G = (\Sigma,h,w)$ where Σ is a finite alphabet, h is a morphism $\Sigma^* \to \Sigma^*$ and w is in Σ^*. System G generates the language $L(G) = \{h^n(w) \mid n \geq 0\}$. Two DOL systems $G = (\Sigma,g,u)$ and $G^1 = (\Sigma,h,v)$ are sequence equivalent if $g^n(u) = h^n(v)$ for all $n \geq 0$.

Let g and h be morphisms on Σ^*. The balance of a string w in Σ^* is denoted by $B(w)$ and defined by

$$B(w) = |g(w)| - |h(w)|$$

where $|x|$ denotes the length of string x.

We say that a pair of DOL systems $G = (\Sigma,g,u)$ and $G^1 = (\Sigma,h,v)$ has bounded balance if there is $C > 0$ such that $|B(w)| \leq C$ for all prefixes w of all strings in $L(G)$.

3. Basics on dominoes

Let S be a set with a binary associative operation o and A a subset of S. The semigroup, the monoid and the semigroup with zero generated by A with the operation o are denoted by $S_o(A)$, $S_o^1(A)$ and $S_o^0(A)$, respectively. The corresponding semigroups with defining relations $R_1,R_2,\ldots,R_k$ are denoted by $S_o(A;R_1,\ldots,R_k)$, $S_o^1(A;R_1,\ldots,R_k)$ and $S^0(A;R_1,\ldots,R_k)$. In the case the semigroup operation is concatenation of elements we omit the operation index from the corresponding notations.

For an alphabet Σ we define the <u>quotient alphabet</u> Σ_q as the set

$$\Sigma_q = \{\frac{a}{b} : a,b \in \Sigma \cup \{1\},\ a\cdot b \neq 1\}$$

of abstract symbols. As a convention we shall write $\frac{a}{1}$ as a whenever a is an element of Σ.

Let a, b and c vary over $\Sigma \cup \{1\}$ and define two generating relations E and I as follows:

$$\text{(E)} \qquad a \cdot \frac{b}{c} = \frac{a}{c} \cdot b ,$$

$$\text{(I)} \qquad \frac{a}{a} = 1.$$

<u>Lemma 1</u>. $S^1(\Sigma_q;E,I)$ is a group, with identity element 1, and the inverse element of $\frac{a}{b}$ being $\frac{b}{a}$.

<u>Proof</u>. We shall only verify that $(\frac{a}{b})^{-1} = \frac{b}{a}$, the rest of the claim being obvious. We have

$$\frac{a}{b} \cdot \frac{b}{a} = 1 \cdot \frac{a}{b} \cdot \frac{b}{a} \overset{(E)}{=} \frac{1}{b} \cdot a \cdot 1 \cdot \frac{b}{a} \overset{(E)}{=}$$

$$\frac{1}{b} \cdot a \cdot \frac{1}{a} \cdot b \overset{(E)}{=} \frac{1}{b} \cdot \frac{a}{a} \cdot b \overset{(I)}{=}$$

$$\frac{1}{b} \cdot b \overset{(E)}{=} \frac{b}{b} \overset{(I)}{=} 1. \qquad \square$$

Henceforth we shall write $\frac{b}{a}$ also as $(\frac{a}{b})^{-1}$ and $((\frac{a}{b})^{-1})^{-1}$ as $\frac{a}{b}$ also when these are considered to be elements of the monoid $S^1(\Sigma_q)$. Thus a symbol $\frac{1}{a}$ of Σ_q will be written as a^{-1}. However, it should be remembered that this is only a conventional notation for elements of $S^1(\Sigma_q)$.

Example. A word $x = a \cdot b \cdot \frac{1}{a} \cdot \frac{1}{b} \cdot \frac{1}{c} \cdot \frac{b}{a}$ may be rewritten as

$$a \cdot b \cdot (c \cdot b \cdot a)^{-1} \cdot (\frac{a}{b})^{-1}$$

or

$$a \cdot b \cdot (a \cdot b \cdot c)^{-R} (\frac{a}{b})^{-1} ,$$

where R denotes the minor image of a word, i.e. R is the anti-isomorphism defined by $(xy)^R = y^R x^R$, and x^{-R} is a short-handed notation for $(x^{-1})^R$. The word x is reduced in $S^1(\Sigma_q)$, but in $S^1(\Sigma_q;E,I)$ we would have

$$x = c^{-1} \cdot \frac{b}{a} .$$

□

The rest of the paper will be devoted to special kinds of elements in $S^1(\Sigma_q)$, which we call dominoes.

A word x in $S^1(\Sigma_q)$ is called a <u>domino</u> if it has a presentation

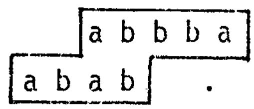
(1) $$x = x_1 \cdot y \cdot x_2 ,$$

where x_1 and x_2 are in $S^1(\Sigma) \cup S^1(\Sigma^{-1})$ and $y = 1$ in $S^1(\Sigma_q;I)$. Here Σ^{-1} denotes the set $\{a^{-1} : a \in \Sigma\}$. The presentation (1) is canonical if x_1 is of maximal length in $S^1(\Sigma) \cup S^1(\Sigma^{-1})$ as a prefix of x. The words $x_1 = \ell(x)$, $y = m(x)$ and $x_2 = r(x)$ are the <u>left</u>, <u>middle</u> and <u>right</u> <u>parts</u> of x, respectively.

As an example let us consider a word

(2) $$x = a^{-1} \cdot b^{-1} \cdot \frac{a}{a} \cdot \frac{b}{b} \cdot b \cdot b \cdot a .$$

By the definition x is a domino, since $\ell(x) = a^{-1}b^{-1}$ is in $S^1(\Sigma^{-1})$, $m(x) = \frac{a}{a} \cdot \frac{b}{b}$ is reducable to the identity by the generating relation I, and $r(x) = bba$ is in $S^1(\Sigma)$. Furthermore (2) is a canonical presentation of x; in fact the only presentation. The domino x may be illustrated graphically as follows:



The concept of a canonical presentation comes into use when in (1) we have $y = 1$ (in $S^1(\Sigma_q)$) and both x_1 and x_2 belong to the same semigroup $S(\Sigma)$ or $S(\Sigma^{-1})$.

Clearly, each domino has a unique canonical presentation.

Let D_Σ denote the set of dominoes in $S^1(\Sigma_q)$. The set D_Σ is not a submonoid of $S^1(\Sigma_q)$. To see this let us consider dominoes $x = a \cdot a^{-1}$ and $z = a$. Their concatenation $x \cdot z = a \cdot a^{-1} \cdot a$ is not a domino and thus D_Σ is not closed under catenation of elements. However,

Lemma 2. D_Σ generates $S^1(\Sigma_q;E)$.

Proof. The claim follows from the observation that by the relation E each word in $S^1(\Sigma_q)$ can be reduced to a domino. □

Now we shall define some technical terms for dominoes to be used later on.

Let h_u and h_ℓ be two morphisms from $S^1(\Sigma_q)$ into $S^1(\Sigma)$ such that if $\frac{a}{b}$ is in Σ_q then

$$h_u(\tfrac{a}{b}) = a \quad \text{and} \quad h_\ell(\tfrac{a}{b}) = b.$$

The words $h_u(x)$ and $h_\ell(x)$ are called the upper and the lower components of the domino x, respectively.

The balance of a domino x is an integer

$$B(x) = \max\{|\ell(x)|, |r(x)|\},$$

(where $|w|$ denotes the length of the word w). A domino x is said to be fine if it is reducable in $S^1(\Sigma_q;I)$ and x is p-fine if $p \cdot B(x) \leq |m(x)|$, where p is a rational number. Furthermore x is called a B-domino, if it is B-fine and $B(x) \leq B$.

Example. Let $x = a \cdot b \cdot a \cdot \frac{b}{b} \cdot \frac{b}{b} \cdot \frac{a}{a} \cdot \frac{a}{a}$. The upper component of x is the word

$$h_u(x) = ababbaa$$

and the lower component of x is

$$h_\ell(x) = bbaa .$$

The balance of x is equal to 3 ($B(x) = \max\{3,0\}$) and x is a fine domino, since it can be reduced by using the generating relations I.

The following lemmas are simple consequences of the definitions above.

Lemma 3. A domino x is fine iff $m(x) \neq 1$. □

Lemma 4. If $x = 1$ in $S^1(\Sigma_q;I)$ then $B(x) = 0$. □

If a domino x satisfies the conclusion, $B(x) = 0$, of the previous lemma we shall say that x is an equal domino. Thus an equal domino has empty left and right parts, and hence $x = m(x)$. By this observation we derive

Lemma 5. Let x be an equal domino. Then x is a B-domino for each B in Q. As a special case of this we have that x is p-fine domino for all rational numbers p. □

Dominoes x and y are called shiftable with respect to each other, $x \sim y$, if $x = y$ in $S^1(\Sigma_q;E)$. The shift of dominoes x

and y such that $x \sim y$, is the integer

$$s(x,y) = s(y,x) = \begin{cases} |\ell(x)|+|\ell(y)|, & \text{if } \ell(x) \in S^1(\Sigma),\ \ell(y) \in S^1(\Sigma^{-1}) \\ ||\ell(x)|-|\ell(y)||, & \text{if } \ell(x),\ell(y) \in S^1(\Sigma) \text{ or } \ell(x),\ell(y) \in S^1(\Sigma^{-1}) . \end{cases}$$

In case x and y are shiftable we also say that y is a <u>shift</u> of x.

<u>Lemma 6</u>.

(i) The relation $\sim$ is a congruence relation among dominoes.

(ii) $x \sim y$ iff $h_u(x) = h_u(y)$ and $h_\ell(x) = h_\ell(y)$.

(iii) $s(x,y) = 0$ iff $x = y$.

The following lemma is frequently used in chapter 5. Its proof follows closely the "shifting argument" as given in [2].

<u>Lemma 7</u>. If $x \sim y$ and $s(x,y) > 0$ then x is of the form

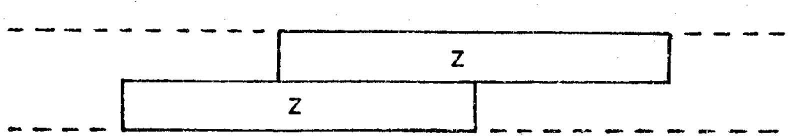
(1) $x = \alpha_1\beta_1\beta^k\alpha_2$,

where β is an equal domino, β_1 a postfix of β, $\alpha_1 = \ell(x)$, $\alpha_2 = r(x)$ and $|\beta| \leqq s(x,y)$. Furthermore we have

(2) $k \geqq \left[\frac{|m(x)|}{s(x,y)}\right]$,

where $[c]$ denotes the greatest integer $\leqq c$.

<u>Proof</u>. Let us consider the middle part of the domino x, i.e. the equal domino $m(x)$. Because $s(x,y) > 0$ this middle part will be also shifted by the amount $s(x,y)$ in the domino y. Thus we have the situation in y:



where $z = h_u(m(x)) = h_\ell(m(x))$, and hence

(3) $z_1 z = z z_2$

for a prefix z_1 of z and for a postfix z_2 of z each of length s(x,y). Thus by (3) z can be written in the form

$$z = z_0 z_2^k$$

for a postfix z_0 of z_2 and some integer k. By choosing β to be the domino with $h_u(\beta) = h_\ell(\beta) = z_2$ and β_1 to be the domino with $h_u(\beta_1) = h_\ell(\beta_1) = z_0$, the result (1) follows. The inequality (2) is an immediate corollary of (1) and the fact that $|\beta| \leq s(x,y)$. □

A domino x of the form (1) will be called <u>periodic</u>.

A domino x is called <u>standard</u> if there is no other domino y such that $x \sim y$ and either

(i) $|\ell(y)| < |\ell(x)|$

or

(ii) $|\ell(y)| = |\ell(x)|$ and $\ell(y) \in S^1(\Sigma)$.

Thus we require that the left part of a standard domino is of minimal length and if there are two such dominoes then we are to take the one which has the left part in $S^1(\Sigma)$. As an example let us consider a domino

$$x = a^{-1} \cdot b^{-1} \cdot \frac{b}{b} \cdot \frac{a}{a} \cdot \frac{a}{a} \cdot \frac{b}{b} \cdot \frac{b}{b} \ .$$

This is not a standard domino, since

$$y = b \cdot a \cdot \frac{a}{a} \cdot \frac{b}{b} \cdot \frac{b}{b} \cdot a \cdot a \cdot b \cdot b$$

fulfils the condition (ii) in the definition of a standard domino, and $x \sim y$.

In order to operate with dominoes we now introduce binary operations for dominoes. We say that a domino z is a <u>matching</u> of dominoes x and y if z = xy in $S^1(\Sigma_q;E)$. We immediately derive from this definition that z is a matching of x and y if and only if $h_u(z) = h_u(x)\cdot h_u(y)$ and $h_\ell(z) = h_\ell(x)\cdot h_\ell(y)$.

A matching is called standard if it is a standard domino. From this definition we conclude the following lemma.

Lemma 8. Each pair of dominoes (x,y) has a unique standard matching, which is denoted by $x{*}y$. The dominoes form a semigroup, $S_*(D_\Sigma)$, with this operation. □

We note that $S_*(D_\Sigma)$ does not contain any identity element. The empty word serves as a centre element for the semigroup: $1{*}x = x{*}1$ for each domino x. However, if x is not a standard domino then $x \neq 1{*}x$. But we have the relation $x \sim 1{*}x$ for all dominoes x and thus the following lemma.

Lemma 9. The set of standard dominoes, DS^1_Σ, is a monoid with the operation $*$. This monoid is isomorphic to the monoid $S^1_*(D\ )/\sim$, the factor monoid of $S^1_*(D_\Sigma)$ modulo the congruence $\sim$. □

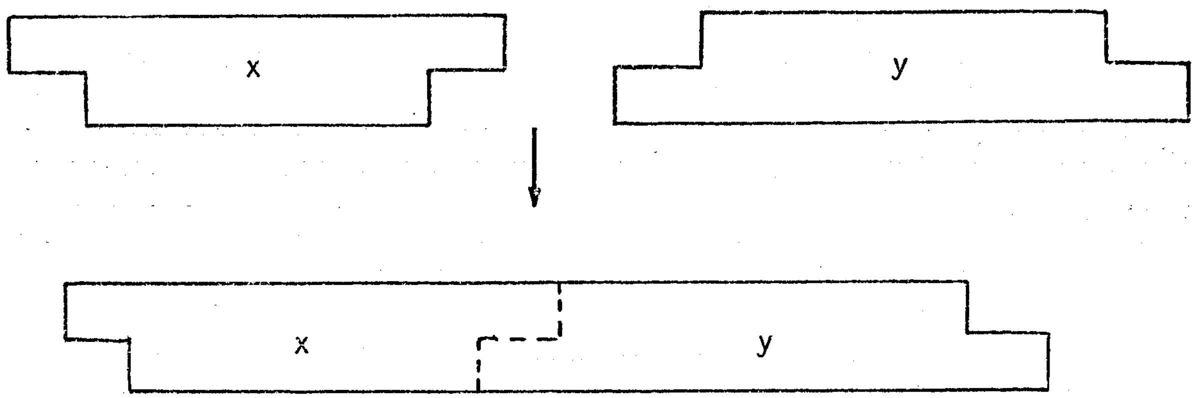
The operation of standard matching does not preserve the structures of the component dominoes because of the possible shiftings made. We shall define now a new operation which preserves the structures more faithfully.

A matching z of dominoes x and y is called faithful if $\ell(z) = \ell(x)$ and $r(z) = r(y)$, i.e. if the left part of x and the right part of y retain the same in the matching. We immediately note that there are dominoes, which have no faithful matchings. One such example is: $x = a\frac{a}{a}$, $y = a\frac{b}{b}a$.

Lemma 10.

(i) If the faithful matching of x and y is defined then it is unique and is denoted by $x \times y$.

(ii) $x \times y$ is defined iff $r(x)^R = \ell(y)^{-1}$.



Proof. The lemma follows from the conditions $\ell(z) = \ell(x)$ and $r(z) = r(y)$, since these guarantee that no shifting can occur when dominoes are matched faithfully. □

The operation of faithful matching is cancellable, i.e. if $x \times y = x \times z$ or $y \times x = z \times x$ then $y = z$ when these matchings are defined. The operation of standard matching does not have this property, but we have a weaker condition: $x * y = x * z$ and $y * x = z * x$ both imply that $y \sim z$. The relationship between standard and faithful matchings is given below.

Lemma 11. For each x and y there are dominoes x_1 and y_1 such that $x \sim x_1$, $y \sim y_1$ and $x * y = x_1 \times y_1$.

We shall now define a new operation $\circledast$ in order to have faithful matchings everywhere defined. Let 0 be a new symbol and let

$$x \circledast y = \begin{cases} x \times y, & \text{if defined} \\ 0, & \text{otherwise}. \end{cases}$$

With this new operation the dominoes form a semigroup with a zero element, 0. This semigroup, denoted by $S^0_{\circledast}(D_\Sigma)$, is not a monoid as is seen from the conditions for faithful matchings.

Lemma 12. If $1 \odot x \neq 0$ then x has left part equal to 1. Furthermore if $1 \odot x \odot 1 \neq 0$ then x is an equal domino.

We shall now define and study the concept of domino equivalence, which will be an important tool in chapter 5.

Dominoes x and y are equivalent, $x \equiv y$, if $\ell(x) \equiv \ell(y)$ and $r(x) \equiv r(y)$, i.e. if their left and right parts are equal words, respectively. The next lemma is a direct corollary to this and previous definitions.

Lemma 13.

(i) The relation $\equiv$ is an equivalence relation in D_Σ.

(ii) The relation $\equiv$ is a congruence relation in $S_{\odot}^{O}(D_\Sigma)$.

(iii) $x \equiv 1$ iff x is an equal domino.

(iv) If x and y are nonfine and $x \equiv y$ then $x = y$. □

4. Domino languages

We shall now study some decidability problems for families of sets of dominoes. Let $F = \{A_i : i \in J\}$ be a set of domino sets indexed over J such that each of the elements A_i is a subset of D_Σ for a fixed alphabet Σ.

The equality problem for F is the following decidability problem: Is is decidable if $A = \{1\}$ in $S^1(\Sigma_q;I)$ for elements A of F, i.e. is it decidable whether or not a domino set A in F consists of equal dominoes only?

The unity problem for F is stated as: Is it decidable if $A \cap \{1\}$ is nonempty in $S^1(\Sigma_q;I)$ for elements A of F? This problem may be restated in the form: Is it decidable whether or not a domino set A in F contains an equal domino?

The finiteness problem for F asks if it is decidable whether or not an element A of F is finite?

Our first result deals with families F the elements of which are finite sets of dominoes. Since the claim of the theorem is obvious we shall omit the proof.

Theorem 1. The equality, unity and finiteness problems are decidable for families F consisting of finite sets of dominoes. □

A more interesting case arises when we allow the elements of F to be infinite. In the next theorem we let the domino sets vary through the semigroups $S_*(A)$ and $S^\circ_\circledast(A)$, where A is a finite set of generators.

Theorem 2. Let $F = \{A_i : i \in J\}$ be the family of finite domino sets. Then the following hold true.

(i) The equality and finiteness problems are decidable for

$F_1 = \{S^{O}_{\circledast}(A_i) : i \in J\}$ and $F_2 = \{S_*(A_i) : i \in J\}$.

(ii) The unity problem is undecidable for F_2 but decidable for F_1.

Proof.

(i) The semigroups $S^{O}_{\circledast}(A)$ and $S_*(A)$ consist of equal dominoes if and only if the set A consists of equal dominoes. The latter case is clearly decidable since A is a finite set. Thus the equality problem is decidable for both F_1 and F_2.

For the finiteness problem let us first consider the family F_2. If $A = \{1\}$ then we shall have also $S_*(A) = \{1\}$. On the other hand suppose A has a nonempty word x. In this case $S_*(A)$ contains the dominoes

$$x_k = x * x * \ldots * x ,$$

where the product has k factors x, for each $k \geq 1$. By the definition of standard matching the length sequence $\{|x_k|\}_{k=1}^{\infty}$ must tend to infinity. Hence $S_*(A)$ is infinite if and only if A is nonempty and different from $\{1\}$.

For the finiteness conditions for $S^{O}_{\circledast}(A)$ let us assume that $\#A = k$, i.e. the cardinality of A equals k. If $k = 0$ then $S^{O}_{\circledast}(A)$ is finite. Suppose that $k \geq 1$, and that $S^{O}_{\circledast}(A)$ contains a word x such that

(1) $x = x_1 \times x_2 \times \ldots \times x_{k+1}, \quad x_i \in A$.

Since there are more factors in (1) than the cardinality of A two of these factors must be equal, say x_i and x_j. Then

$$x = x_1 \times \ldots \times x_i \times (x_{i+1} \times \ldots \times x_j) \times x_{j+1} \times \ldots \times x_{k+1}$$

and by the definition of faithful matching we have that $\ell(x_{i+1})^R = r(x_i)^{-1}$. Now $x_i = x_j$ and thus the subword $x_{i+1} \times \ldots \times x_j$ may be repeated indefinitely, i.e. the words

$$x_{i+1} \times \ldots \times x_j,\ x_{i+1} \times \ldots \times x_j \times x_{i+1} \times \ldots \times x_j,\ \ldots$$

are all different from 0. Thus the condition (1) implies that $S^{O}_{\circledast}(A)$ is infinite. On the other hand if there are no words x satisfying the condition (1) then $S^{O}_{\circledast}(A)$ is finite. Therefore the finiteness problem is decidable for F_1, too.

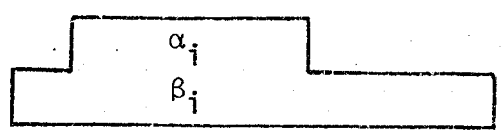
(ii) The proof of the decidability of the unity problem for F_1 follows closely the above argumentation for finiteness and is omitted here.

In order to prove that the unity problem is undecidable for F_2 we shall consider instances of the Post correspondence problem, PCP in short. Let

(2) $(\alpha_1,\alpha_2,\ldots,\alpha_k;\ \beta_1,\beta_2,\ldots,\beta_k)$

be an instance of the PCP. Define a finite set A of dominoes by setting an element

$$x_i = \alpha_i * \beta_i^{-R},$$



into A for each $i = 1,2,\ldots,k$. By this construction it immediately follows that $S_*(A)$ contains an equal domino $x = x_{i_1}x_{i_2}\ldots x_{i_n}$ if and only if $i_1 i_2 \ldots i_n$ is a solution to the problem (2). Because the PCP is an undecidable problem so must be the unity problem for F_2. □

From the proof of the previous theorem we already see that the semigroups $S^{O}_{\circledast}(A)$ possess some regularity properties. We shall formalize this in the following way.

Theorem 3. Let A be a finite set of dominoes and ρ an isomorphism from an alphabet Δ onto the set A. Then the set

$$B_\rho = \{w \mid w \in S(\Delta),\ \rho(w) \in S^{\bigcirc}_{\circledast}(A)\backslash\{0\}\}$$

is a regular set.

Proof. Let $A = \{x_1, x_2, \ldots, x_k\}$ and $\Delta = \{b_1, b_2, \ldots, b_k\}$. We shall design a right linear grammar for B_ρ. This grammar has starting letter S, nonterminals $Y_1, Y_2, \ldots, Y_k$, and productions:

$$S \to Y_i \qquad (i = 1,2,\ldots,n),$$
$$Y_i \to b_i Y_j$$

for each i and j in $\{1,2,\ldots,n\}$ such that x_i and x_j are faithfully matchable, and

$$Y_i \to b_i \qquad (i = 1,2,\ldots,n) .$$

From this construction the claim follows immediately. □

A direct corollary to this result states that the upper and lower components of dominoes in $S^{\bigcirc}_{\circledast}(A)$ have regular properties.

Corollary 4. Let A be a finite set of dominoes. Then $h_u(S^{\bigcirc}_{\circledast}(A))$ and $h_\ell(S^{\bigcirc}_{\circledast}(A))$ are regular sets. □

The above considerations can be generalized to wider classes of dominoes. Let Δ be an alphabet and ϕ a morphism from $S(\Delta)$ into $S_*(D_\Sigma)$. For a subset L of $S(\Delta)$ we define the set $\phi(L)$ to be a domino language of L defined by ϕ. Similarily a faithful domino language of L defined by a morphism $\psi : S(\Delta) \to S^{\bigcirc}_{\circledast}(D_\Sigma)$ is the set $\psi(L)$ of dominoes.

Theorem 5. If A is a subset of a finitely generated semigroup $S_*(A_1)$, then A is a domino language (of some L).

Proof. First of all we note that $S_*(A_1)$ is a domino language of $S(\Delta)$ defined by an isomorphism $\phi : \Delta \to A_1$. Now A is a domino language of L defined by ϕ, where L is a set

$$L = \{b_1b_2\dots b_n \mid b_i \in \Delta,\ \phi(b_1b_2\dots b_n) \in A\}. \qquad \square$$

In quite the same way we can prove

Theorem 6. If A is a subset of a finitely generated semigroup $S^{\circ}_{\circledast}(A_1)$, then A is a faithful domino language (of some L). $\square$

By these two theorems we may consider domino languages and faithful domino languages instead of sets of dominoes.

Next we shall reduce the domino problems (equality and unity) to problems concerning ordinary formal languages.

A morphic equality problem for a language L is the problem: Given two morphisms h and g do they coincide on L, i.e. does $h(w) = g(w)$ hold true for each w in L?

Theorem 7. The morphic equality problem for L is equivalent to the equality problem for domino languages of L.

Proof. Let L be a subset of $S^1(\Delta)$ and h,g two morphisms from $S^1(\Delta)$ into $S^1(\Sigma)$. Define a morphism $I_q : S^1(\Sigma)$ into $S^1(\Sigma_q)$ by setting $I_q(a) = \frac{a}{1}$ and let

$$\phi(w) = I_q h(w) * (I_q g(w))^{-R}$$

be a morphism from $S(\Delta)$ into $S_*(D_\Sigma)$. Then by the definition of ϕ we have that $h(w) = g(w)$ if and only if $\phi(w)$ is an equal domino.

On the other hand let ϕ be given for L. We shall now define two morphisms h and g by

$$h(w) = h_u\phi(w)$$

and

$$g(w) = h_\ell\phi(w)\ .$$

Now $h(w) = g(w)$ if and only if $\phi(w)$ is an equal domino. $\square$

5. DOLs and dominoes

In this chapter we apply dominoes to the DOL sequence equivalence problem in order to obtain for it a simple new algorithm.

Given two morphisms h and g on an alphabet Σ and a word w_0 in $S^1(\Sigma)$, we ask whether $h^n(w_0) = g^n(w_0)$ for all integers $n = 0,1,2,\ldots$. This problem can be restated using the following lemma.

<u>Lemma 14</u>. $h^n(w_0) = g^n(w_0)$ for all n if and only if $h^{n+1}(w_0) = gh^n(w_0)$ for all n.

<u>Proof</u>. Obvious. □

Hence, by Theorem 7 the DOL sequence equivalence problem is equivalent to the equality problem for a domino language of $L = \{h^n(w_0) : n \geq 0\}$ defined by a morphism ϕ such that

$$\phi(w) = I_q h(w) * (I_q g(w))^{-R} .$$

We shall now fix the morphisms h and g as well as the starting string w_0. We begin our considerations with some definitions.

Let x_1 be a given domino. Then x_1 will result a sequence of new dominoes $x_2, x_3, \ldots$ defined by iterative using of the morphisms h and g. Suppose that $h_u(x_1) = w_1$ and $h_\ell(x_1) = w_2$. At the first step x_1 results a domino

$$x_2 = h(w_1) * (g(w_1))^{-R} .$$

In general we have

$$x_{i+1} = h(h_u(x_i)) * (g(h_u(x_i))^{-R} ,$$

which can be written as

$$x_{i+1} = \phi h^i(w_1) = \phi h^i h_u(x_1) .$$

We shall say that a domino x is <u>repetitive modulo</u> t $(t>1)$ if $x \equiv y$ for some y such that $y \sim \phi h^t h_u(x)$. Furthermore x is a <u>(B,t)-domino</u> if it is a B-domino and is repetitive modulo t.

In the following lemmas we study the properties of (B,t)-dominoes in the framework of the system (h,g,w_0).

<u>Lemma 15</u>. Suppose x_1 and x_2 are (B,t)-dominoes such that $x_1 \sim x_2$. Then either $x_1 = x_2$ or x is repetitive modulo t whenever $x \sim x_1$.

<u>Proof</u>. Assume that $x_1 \neq x_2$ in which case $s(x_1,x_2) \neq 0$ and $s(x_1,x_2) \leq 2 \cdot B$. Let $x_{i,t}$ be a domino such that $x_{i,t} \sim \phi h^t h_u(x_i)$ for $i = 1,2$. Since $x_1 \sim x_2$ we have also that $x_{1,t} \sim x_{2,t}$. Moreover, $x_i \equiv x_{i,t}$ implies that $s(x_{1,t},x_{2,t}) = s(x_1,x_2)$, where $i = 1,2$.

By Lemma 7 the following hold

(1) $x_1 = \alpha_1 \beta^k \beta_1 \alpha_2$,

(2) $x_{1,t} = \alpha_1 \beta^r \beta_2 \alpha_2$

for some equal domino β, β_1 and β_2 being prefixes of β, $k \geq [\frac{1}{2} \frac{|m(x)|}{B(x)}]$, and $\alpha_1 = \ell(x_1)$, $\alpha_2 = r(x_1)$. Here the period β may be assumed to be of minimal length.

<u>Claim</u>. $\beta_1 = \beta_2$.

Assume the contrary, i.e. $\beta_1 \neq \beta_2$.

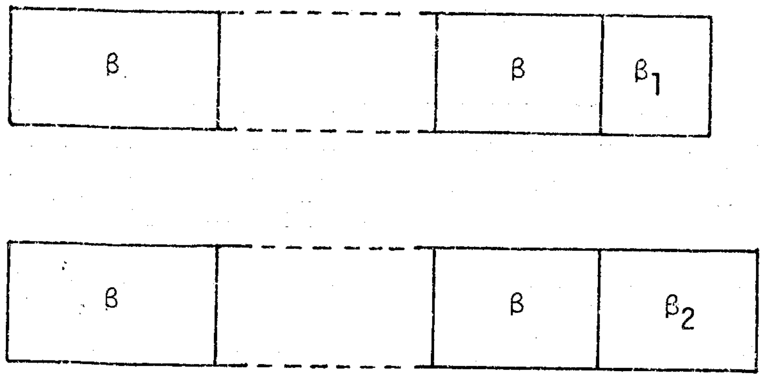
The dominoes x_i and $x_{i,t}$ $(i = 1,2)$ can be written as

$$x_i = \alpha_i * z_i * \alpha_2$$

and

$$x_{i,t} = \alpha_1 * z_{i,t} * \alpha_2 .$$

The equivalences $x_i \equiv x_{i,t}$ imply that also $z_i \equiv z_{i,t}$ for $i = 1,2$. Now the dominoes z_1 and $z_{1,t}$ look like



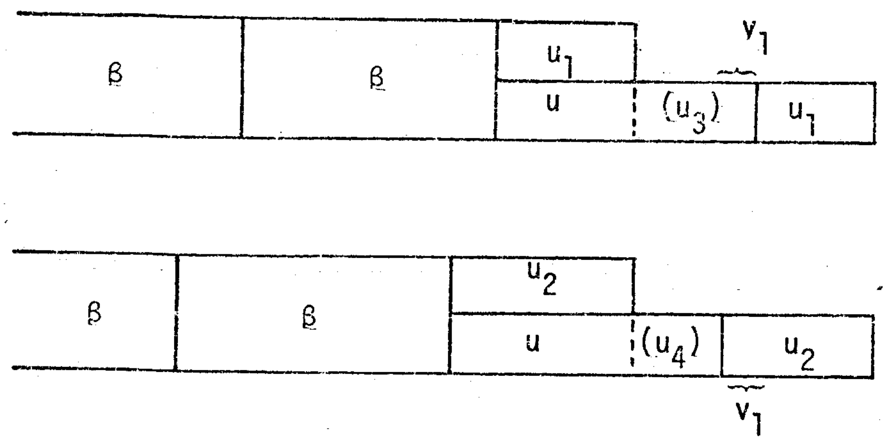
The dominoes z_2 and $z_{2,t}$ are equivalent and thus their right parts are equal to each other. Because of $x_1 \sim x_2$ and $x_{1,t} \sim x_{2,t}$ this may happen only if the upper component of x_1 is shifted some β blocks to the right or to the left. These two shifting directions are clearly symmetrical when we consider the dominoes z_i and $z_{i,t}$ and thus the same proof applies to both of them.

This shifting and the equivalence $z_2 = z_{2,t}$ imply that

(3) $r(z_2) = u_3 u^v u_1 = u_4 u^v u_2 = r(z_{2,t})$,

where $u = h_u(\beta)$, $v \geq 0$, $u_i = h_u(\beta_i)$ for $i = 1,2$, and

(4) $u = u_1 u_3 = u_2 u_4$.



By assumption $\beta_1 \neq \beta_2$ we have that $u_1 \neq u_2$ and thus also $u_3 \neq u_4$. Let us suppose that $|u_2| > |u_1|$. The reverse case is clearly symmetrical.

We have

$$(5)\quad \begin{cases} u_2 = v_1u_1 , \\ u_3 = u_4v_1 \end{cases}$$

for some nonempty word v_1. In all we obtain from (4) and (5) that

$$(6)\quad u = u_1u_4v_1 = v_1u_1u_4 .$$

Here u_1u_4 and v_1 are nonempty words which commute in u. Thus u is a periodic word of the form v^k form some $k \geq 2$. Hence the domino $\beta = u \mathbin{\because} u^{-R}$ would be periodic contradicting our assumption of the minimality of $|\beta|$ in (1). The proof of the claim is completed by this contradiction.

The claim now implies that x_1 and $x_{1,t}$ are always shiftable by the same amount because in the equations (1) and (2) we have that $\beta_1 = \beta_2$. The results of these shiftings remain equivalent, and thus the lemma follows. □

Lemma 16. Let $x_1 \times y_1 \times z_1$ be a faithful matching of (B,t)-dominoes x_1, y_1 and z_1. Suppose $x_2 \times y_2 \times z_2$ is a matching of (B,t)-dominoes $x_2 \sim x_1$ and $z_2 \sim z_1$, and a B-domino $y_2 \sim y_1$. Then y_2 is a (B,t)-domino.

Proof. Suppose that $y_2 \neq y_1$ in which case also $x_1 \neq x_2$ and $z_1 \neq z_2$. Now the dominoes $x_1 \times y_1 \times z_1$ and $x_2 \times y_2 \times z_2$ are shiftable with respect to each other and thus

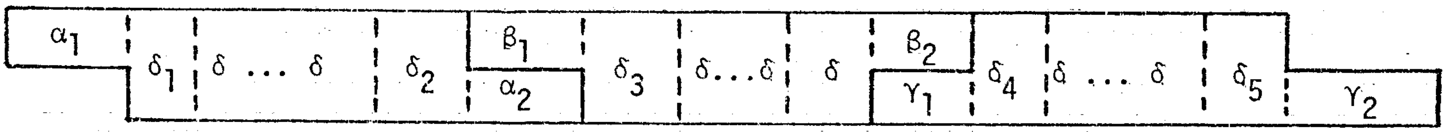
$$(1)\quad x_1 \times y_1 \times z_1 = \alpha_1\delta_1\delta^k\gamma_2 ,$$

$$(2)\quad x_1 = \alpha_1\delta_1\delta^{k_1}\delta_2\alpha_2 ,$$

$$(3)\quad y_1 = \beta_1\delta_3\delta^{k_2}\beta_2 ,$$

$$(4)\quad z_1 = \gamma_1\delta_4\delta^{k_3}\delta_5\gamma_2 ,$$

where δ is a minimal period and $\delta_2 \times (\alpha_2 * \beta_1) \times \delta_3 = \delta^{v_1}$, $(\beta_2 * \gamma_1) \times \delta_4 = \delta^{v_2}$ for some integers v_1 and v_2. Note that $\alpha_2 = \beta_1^{-R}$ and $\beta_2 = \gamma_1^{-R}$.



The dominoes x_1 and x_2 are both (B,t)-dominoes such that $x_1 \sim x_2$. By the proof of Lemma 15 we obtain that

(5) $x_{1,t} = \alpha_1 \delta_1 \delta^{r_1} \delta_2 \alpha_2$,

for some integer r_1 (Here $x_1 \equiv x_{1,t}$ modulo t). The same argument applies to $z_{1,t} \equiv z_1$

(6) $z_{1,t} = \gamma_1 \delta_4 \delta^{r_3} \delta_5 \gamma_2$,

for some integer r_3.

Moreover $y_1 \equiv y_{1,t}$ and the domino $y_{1,t}$ is now sandwiched between $x_{1,t}$ and $z_{1,t}$ in the domino $x_{1,t} \times y_{1,t} \times z_{1,t}$. This condition with the periodicity property in (1) implies that

(7) $y_{1,t} = \beta_1 \delta_3 \delta^{r_2} \beta_2$

for some integer r_2. By (3) and (7) the result follows. $\square$

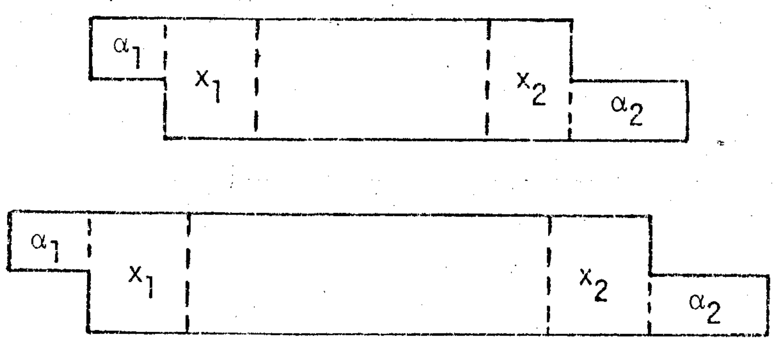
We shall call a B-domino x <u>unique</u> (with respect to B and t) if x has exactly one shifting y such that y is a (B,t)-domino. By Lemma 15 if x is not unique then it has either no shiftings, which are (B,t)-dominoes or all its shiftings which are B-dominoes are also (B,t)-dominoes.

With this terminology Lemma 16 can be reformulated as follows:

<u>Lemma 17</u>. Let $x = x_1 \times y_1 \times z_1$ be a faithful matching of (B,t)-dominoes x_1, y_1 and z_1, where x_1 and z_1 are nonunique If x has

a proper shift which is a B-domino then y_1 is nonunique. □

In order to simplify the conditions for the equality problem we shall now introduce a stronger version of the equivalence relation for dominoes. The dominoes x and y are B-equivalent, $x \equiv_B y$ if they are equivalent and $m(x), m(y)$ have common prefix and postfix of length B.



Clearly, B-equivalence implies equivalence. Analogously, we call a domino x strongly repetitive modulo $t \geq 1$ if $x \equiv_B y$ for some y such that $y \sim \phi h^t h_u(x)$. Moreover x is a strong (B,t)-domino if it is a B-domino and is strongly repetitive modulo t.

We now prove that a faithful matching of unique strong dominoes is unique.

Lemma 18. Let $x_1 \times y_1$ be a faithful matching of unique strong (B,t)-dominoes x_1 and y_1. If $y = x_2 \times y_2$ is a (B,t)-domino, where $x_2 \sim x_1$ and $y_2 \sim y_1$ are B-dominoes then $x = y$.

Proof. Suppose that $x \neq y$. Let x_t, y_t, $x_{1,t}$ and $y_{1,t}$ be the corresponding repetitions mod t of x, y, x_1 and y_1. We shall write $y_t = x_{2,t} \times y_{2,t}$, where $x_{1,t} \sim x_{2,t}$ and $y_{1,t} \sim y_{2,t}$.

From the assumption $x \neq y$ and from the uniqueness of x_1 and y_1 we conclude that $x_2 \not\equiv x_{2,t}$ and $y_2 \not\equiv y_{2,t}$. This means that $r(x_2) \neq r(x_{2,t})$. However, $\ell(x_2) = \ell(x_{2,t})$ since $y \equiv y_t$ and thus

$|r(x_2)| = |r(x_{2,t})|$.

The dominoes x and y are shiftable and hence periodic with a minimal period β. This holds also for x_t and y_t since the shift must be the same for both of these pairs and because of the equivalences $x \equiv x_t$ and $y \equiv y_t$ the minimal period is β for all of these four dominoes.

Now x_1 will be of the form

$$x_1 = \alpha_1\beta_1\beta^k\alpha_2$$

and $x_{1,t}$ of the form

$$x_{1,t} = \alpha_1\delta_1\delta^k\alpha_2 ,$$

for some δ, which is a cyclic conjugate of β, i.e. $\beta = \beta_2\beta_3$ and $\delta = \beta_3\beta_2$ for some β_2 and β_3. But since we assumed that x_1 is strongly repetitive, we must have that $\beta = \delta$. By the same reason we have also that $\beta_1 = \delta_1$. These conditions would imply that x_2 is a (B,t)-domino by the proof of Lemma 15. This contradicts the uniqueness assumption for x_1. Hence $x_2 = x_1$ and in all $x = y$. □

In the next lemma we consider sequences of matchable dominoes.

Lemma 19. Let $x = x_0 \times x_1 \times \ldots \times x_k$ be a faithful matching of B-dominoes $x_0, x_1, \ldots, x_k$ such that x_0 is a (B,t)-domino and

(1) each pair (x_i, x_{i+1}) has a matching $z_i = y_i \times u_i$, where y_i and u_i are strong (B,t)-dominoes and $y_i \sim x_i$, $u_i \sim x_{i+1}$ for each $i = 0,1,\ldots,k-1$.

Then either all of the dominoes x_i are (B,t)-dominoes or there is an integer j such that $x_0, x_1, \ldots, x_{j-1}$ are (B,t)-dominoes and $x_j, \ldots, x_k$ are unique, but not (B,t)-dominoes.

Proof. Assume that not all of the dominoes x_i, $i = 1,2,\ldots,k$, are (B,t)-dominoes and let x_j be the first domino in x which is not a

(B,t)-domino. Then $j \geqq 1$ and x_j must be unique by Lemma 15. The domino x_{j-1} is nonunique because it is a (B,t)-domino and can be shifted to another (B,t)-domino by Condition (1) and by the fact that x_j is a unique domino in a wrong position, i.e. $x_j \neq u_{j-1}$.

Suppose now that x_{j+r} is a (B,t)-domino for some minimal $r \geqq 1$. Then x_{j+r} is nonunique by the same arguments as for x_{j-1}. Thus the dominoes $x_j,\ldots,x_{j+r-1}$ are all unique but not (B,t)-dominoes. Let

$$y = y_j \times y_{j+1} \times \cdots \times y_{j+r-1}$$

be a domino obtained from Condition (1). Here y_{j+i} is a unique strong (B,t)-domino for each $i = 0,1,\ldots,r-1$. By Condition (1) also x_{j-1} and x_{j+r} may be shiftably matched to y yielding a domino

(2) $y_{j-1} \times y_j \times \cdots \times y_{j+r-1} \times u_{j+r-1}$

all the factors of which are strong (B,t)-dominoes.

Applying Lemma 18 to the domino y we obtain that y is a unique domino itself. But from (2) we have a domino $z = y_{j-1} \times y \times u_{j+r-1}$, where y_{j-1} and u_{j+r-1} are nonunique. Furthermore, the domino z is properly shiftable to the domino $x_{j-1} \times x_j \times \cdots \times x_{j+r}$ and thus y must be nonunique by Lemma 17. This contradicts our previous result for the uniqueness of y. In all the counter assumption fails and hence the claim follows. □

Now, we shall construct the necessary and sufficient conditions for a solution to the domino equality problem.

Let us denote the DOL language $\{h^n(w_0) : n \geqq 0\}$ by L. We define the set of adjacent symbols in L as

$$\mathrm{Init}(L) = \{w : w_1 w w_2 \in L \text{ for some } w_1 \text{ and } w_2\} .$$

Furthermore, two words w_1 and w_2 are said to be <u>adjacent</u> in L if

w_1w_2 is in Init(L). Dominoes $x = \phi h^n(w_1)$ and $y = \phi h^n(w_2)$ are <u>adjacent</u> in $\phi(L)$ if w_1 and w_2 are adjacent in L.

Let A be a set of dominoes and B,n positive integers. The set A is called a <u>base</u> w.r.t. B and n (i.e., with respect to B and n), if A consists of dominoes x such that either (1)

$$x = \phi h^n(a) \text{ and } |x| \geq B^2$$

for a letter a in Σ, or

$$x = \phi h^n(w) \text{ and } B^2 \leq |x| \leq 2 \cdot B^2$$

for a word w in Init(L), or (2)

$$x = x_1 \times x_2 \times x_3 ,$$

where x_2 is from the case (1) and $x_i \sim \phi h^n(w_i)$, $|x_i| < B^2$ such that w_i is Init(L) for $i = 1,3$.

Thus a base (w.r.t. B and n) consists of dominoes of the form $\phi h^n(w)$ which are sufficiently large with respect to B. A base is always a finite set.

We shall now state the necessary and sufficient conditions for a domino language $\phi(L)$ to be a set of equal dominoes.

<u>Condition 1</u>. There are positive integers B, n and t, and a <u>base</u> A of $\phi(L)$ w.r.t. B and n such that

1.1. If elements x and y of A are adjacent in $\phi(L)$ then they have faithfully matchable (B,t)-dominoes x_1 and y_1 as shifts. Furthermore, if $x_{1,t} \sim \phi h^t(x)$, $y_{1,t} \sim \phi h^t(y)$ and $x_{1,t} \equiv x_1$, $y_{1,t} \equiv y$, then the factors of $x_{1,t}$ and $y_{1,t}$ from A are B-dominoes.

1.2. If x is in A and begins (ends, resp.) a word in $\phi(L)$ then x has a (B,t)-domino x_1 as a shift such that $|\ell(x_1)| = 0$ ($|r(x_1)| = 0$, resp.).

1.3. $\phi h^j(w_0)$ is an equal domino for $j \leq n + t$, the factors (in A) of which are B-dominoes.

The next lemma shows that this condition is sufficient to guarantee that $\phi h^j(w_0)$ is equal for each $j \geq 0$.

Lemma 20. Condition 1 implies that $\phi h^j(w_0)$ is an equal domino for all $j \geq 0$.

Proof. From Condition 1.3. we know that $\phi h^j(w_0)$ is an equal domino for $j \leq n + t$.

Let us consider the equal domino $\phi h^{n+1}(w_0)$. We have that

(1) $\phi h^{n+1}(w_0) = x_0 \times x_1 \times \ldots \times x_r$

for some B-dominoes x_i in A by assumption. Moreover the domino x_0 is a (B,t)-domino by Condition 1.2. Hence by Lemma 19 either all the dominoes x_i are (B,t)-dominoes (for $i = 0,1,\ldots,r$) or there is an integer j such that $x_j,\ldots,x_r$ are unique dominoes, which are not (B,t)-dominoes. The second case does not hold true since by Condition 1.2. the domino x_r has a shift, which is a (B,t)-domino with right part equal to the empty word, but x_r itself has $|r(x_r)| = 0$ and thus it must be a (B,t)-domino.

From this reasoning we conclude that the dominoes $x_0, x_1, \ldots, x_r$ are (B,t)-dominoes. This implies that $\phi h^{n+t+1}(w_0) \equiv \phi h^{n+1}(w_0)$ and thus $\phi h^{n+t+1}(w_0)$ is an equal domino. By Condition 1.1. the equal domino $\phi h^{n+t+1}(w_0)$ is expressible as a faithful matching of B-dominoes from A.

By proceeding inductively we obtain the result of the lemma. □

The necessity of Condition 1 follows from the crucial property of DOL systems, namely that every two equivalent DOL systems have a bounded balance. This property has been shown in [3] for "normal" DOL systems as essential step in proving the decidability of the DOL sequence equivalence problem. It has been extended to all DOL systems in [5].

Lemma 21. If $\phi h^j(w_0)$ is an equal domino for each $j \geq 0$, then Condition 1 holds.

Proof. Let us suppose that $L = \{h^j(w_0) : j \geq 0\}$ is infinite. Otherwise the claim in trivial.

The dominoes $\phi h^j(w_0)$ are equal dominoes for all $j \geq 0$ if and only if $h^{j+1}(w_0) = gh^j(w_0)$ holds for all $j \geq 0$. In particular this implies that if $\phi(L) = \{\phi h^j(w_0) : j \geq 0\}$ is a set of equal dominoes then h and g have a bounded balance, i.e.

$$||h(w)| - |g(w)|| \leq B$$

for all words w, which are prefixes of words in L.

Let a be a letter in Σ and consider a word

(1) $h^j(w_0) = w_1 a w_2$,

where $j \geq 0$ and w_1, w_2 are words in Init(L). Applying the morphism h repetitively we obtain

$$h^{j+k}(w_0) = h^k(w_1)h^k(a)h^k(w_2)$$

and

$$\phi h^{j+k}(w_0) = u_k \times z_k \times v_k$$

for $k \geq 0$, where $u_k = \phi h^k(w_1)$, $z_k \sim \phi h^k(a)$ and $v_k \sim \phi h^k(w_2)$ are all dominoes with balance at most B.

The infinite sequence $\{z_i\}$ produced by an occurrence of a, in (1), contains only dominoes which have balance bounded by B. Thus in this sequence there are only finitely many equivalence classes with respect to the relation of equivalence of dominoes. Furthermore, each of the dominoes z_{i+1}, $i = 1,2,\ldots$, is produced from z_i by using the morphisms h and g. This means that the equivalent dominoes in the sequence $\{z_i\}$ occur periodically, i.e. there are integers s and r such that $z_i \equiv z_{i+r}$ whenever $i \geq s$.

Different occurrences of a may produce different sequences, but since the elements of each of these sequences have balance bounded by B, there are only a finite number of different ones. Thus we may select integers n_a and t_a such that every sequence $\{f_i\}$ produced by a is periodic, that is

$$f_i \equiv f_i + t_a$$

whenever $i \geqq n_a$.

Let $n_0 = \max\{n_a : a \in \Sigma\}$ and $t_0 = [t_a : a \in \Sigma]$, the least common multiple of the integers t_a, $a \in \Sigma$. By this choice if $\{p_i\}$ is a sequence produced by some occurrence of a letter then

$$(2) \quad p_i \equiv p_i + t_0$$

for all $i \geqq n_0$.

Let $n_1 \geqq m_0$ be an integer such that either $|h^{n_1+i}(a)| \geqq B^2$ for each $i \geqq 0$ or the set $\{h^j(a) : j \geqq 0\}$ is finite, where a runs through the letters of the alphabet Σ. Furthermore, let A_0 be a base at B and n_1 for $\phi(L)$.

The elements of A_0 are all B-dominoes and each of them has a shift, which is a (B,t_0)-domino by the above arguments. Now we shall repeat the above considerations for elements of A_0 to obtain the result.

Claim. There are integers n and t such that if an occurrence of an element in A_0 produces an infinite sequence $\{p_i\}$ then

$$p_i \equiv_B p_{i+t} ,$$

whenever $i \geqq n$.

Since the proof of this fact follows closely the proof of (2) we omit it here.

The lemma follows immediately from this claim when we select A as a base w.r.t. B and n for $\phi(L)$. □

We remind here that L is a DOL language of the form $\{h^n(w_0) : n \geq 0\}$ and that the morphism ϕ is defined using the given morphisms h and g as indicated in the beginning of the chapter. For clarity we shall write also ϕ as $\phi_{h,g}$ in order to specify the morphisms h and g which define ϕ.

From the previous two lemmas we deduce

<u>Theorem 8</u>. Condition 1 holds for the domino language $\phi_{h,g}(L)$ if and only if $\phi_{h,g}(L)$ is a set of equal dominoes. □

A direct corollary to this theorem states

<u>Corollary 9</u>. Let $L_1 = \{h^n(w_0) : n \geq 0\}$ and $L_2 = \{g_n(w_0) : n \geq 0\}$ be two DOL languages. Condition 1 holds for the domino language $\phi_{h,g}(L_1)$ if and only if $g^n(w_0) = h^n(w_0)$ for all $n \geq 0$.

Thus Condition 1 provides a necessary and sufficient condition for two DOL sequences to be equal.

By Corollary 9 an algorithm for testing Condition 1 will be also an algorithm for the DOL sequence equivalence problem. We shall now proceed to give such an algorithm for Condition 1.

Let $G_1 = (\Sigma, h, w_0)$ and $G_2 = (\Sigma, g, w_0)$ be two DOL systems and let

$$\Sigma = \{a_1, a_2, \ldots, a_r\} .$$

We shall denote by Σ' the set of all letters of Σ such that the morphism h is growing on these, i.e.

$$\Sigma' = \{a : a \in \Sigma \text{ and } \{|h^n(a)| : n = 0,1,\ldots\} \text{ is an infinite set}\} .$$

The algorithm advances in stages n, the initial stage being $n = 1$.

Algorithm for testing Condition 1

Stage n.

(i) Let

$$B_n = \max\{B(x) : x \text{ is a factor of } \phi h^n(w_0) \text{ of the form } x \sim \phi h^n(a_i)\} .$$

(ii) Let B be the maximal integer such that $B \geq B_n$ and $|\phi h^n(a_i)| \geq B^2$ for each a_i in Σ'.

(iii) Construct the base A w.r.t. B and n.

(iv) Evolve the sequences B_{n+j}, $\phi h^{n+j}(w_0)$ and $\phi h^{n+j}(u)$ from $j = 1$ onwards, for each element u of A, until either

(iv 1). $\phi h^{n+j}(w_0)$ is nonequal in which case Condition 1 is not true, or

(iv 2). $B_{n+j} > B$ in which case the procedure will continue to stage $n + j$, or

(iv 3). all dominoes u_1 and u_2 from A which are adjacent in $\phi(L)$ have faithfully matchable strong (B,j)-dominoes as shifts. In this case the algorithm stops and Condition 1 is true. □

We are now to prove that the above algorithm is an effective method for testing Condition 1.

Theorem 10. The above algorithm is a test algorithm for Condition 1.

Proof. Clearly parts (i) and (ii) are effective. Part (iii) is effective since a base is always a finite set which can be constructed by first constructing the finite set

$$\mathrm{Init}_M(L) = \{w : w \in L \text{ and } |w| \leq M\} ,$$

where

$$M = \max\{|h^n(a_i)| : a_i \text{ is in } \Sigma\}.$$

It remains to prove that the part (iv) of the algorithm will be finitely processed during each stage n and that there is a stage n which ends up to cases (iv 1) or (iv 2).

The first of these claims follows immediately from the proof of Lemma 21 since otherwise the condition (iv 1) implies that the dominoes $\phi h^{n+j}(w_0)$ are equal for all $j \geqq 0$, the condition (iv 2) implies that the factors of each $\phi h^{n+j}(w_0)$ have bounded balance and these two results would contradict the part (iv 3) and the claim presented in the proof of Lemma 21.

If Condition 1 does not hold then there is an integer n such that $\phi h^{n}(w_0)$ is nonequal, in which case the algorithm would stop at stage n and would reveal this nonequal domino in part (iv 1).

On the other hand if Condition 1 holds true for $\phi(L)$ then eventually a correct base is found for $\phi(L)$ w.r.t. some B and n. This happens at stage n and the algorithm will stop in part (iv 3) when an integer $j = t$ is reached which fulfils the requirements of Condition 1. □

The above algorithm serves as an effective procedure for testing DOL sequence equivalence problem.

Theorem 8 has also another direct corollary.

<u>Corollary 11</u>. Let h and g be two morphisms and $L = \{h^n(w_0) : n \geqq 0\}$. If the domino language $\phi_{h,g}(L)$ is a set of equal dominoes then it is included in a finitely generated semigroup $S^{\circ}_{\circledcirc}(A)$, for some set A of dominoes. □

References

[1] K. Culik II, Some decidability results about regular and push down translations, Information Processing Letters vol. 8, (1979) 5-8.

[2] K. Culik II, Homomorphisms: Decidability, Equality and Test Sets, Proc. Conf. on Formal Languages in Santa Barbara, December 1979, Academic Press, to appear, also University of Waterloo Research Report CS-80-02 (January 1980).

[3] K. Culik II and J. Fris, The decidability of the equivalence problem for DOL-systems, Inf. and Control, vol. 35 (1977) 20-39.

[4] K. Culik II and A. Salomaa, On the decidability of homomorphism equivalence for languages, J. Comp. System Sci., vol. 17 (1978) 1[illegible]3-175.

[5] A. Ehrenfeucht and G. Rozenberg, Every two equivalent DOL systems have a regular true envelope, Theoretical Computer Science, to appear.

[6] G. Rozenberg and A. Salomaa, The mathematical theory of *L* systems (Academic Press 1980).

[7] A. Salomaa, Formal Languages (Academic Press, New York 1973).

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF WATERLOO
TECHNICAL REPORTS 1979

Report No.	Author	Title
CS-79-01*	E.A. Ashcroft W.W. Wadge	Generality Considered Harmful - A Critique of Descriptive Semantics
CS-79-02*	T.S.E. Maibaum	Abstract Data Types and a Semantics for the ANSI/SPARC Architecture
CS-79-03*	D.R. McIntyre	A Maximum Column Partition for Sparse Positive Definite Linear Systems Ordered by the Minimum Degree Ordering Algorithm
CS-79-04*	K. Culik II A. Salomaa	Test Sets and Checking Words for Homomorphism Equivalence
CS-79-05*	T.S.E. Maibaum	The Semantics of Sharing in Parallel Processing
CS-79-06*	C.J. Colbourn K.S. Booth	Linear Time Automorphism Algorithms for Trees, Interval Graphs, and Planar Graphs
CS-79-07*	K. Culik, II N.D. Diamond	A Homomorphic Characterization of Time and Space Complexity Classes of Languages
CS-79-08*	M.R. Levy T.S.E. Maibaum	Continuous Data Types
CS-79-09	K.O. Geddes	Non-Truncated Power Series Solution of Linear ODE's in ALTRAN
CS-79-10*	D.J. Taylor J.P. Black D.E. Morgan	Robust Implementations of Compound Data Structures
CS-79-11*	G.H. Gonnet	Open Addressing Hashing with Unequal-Probability Keys
CS-79-12	M.O. Afolabi	The Design and Implementation of a Package for Symbolic Series Solution of Ordinary Differential Equations
CS-79-13*	W.M. Chan J.A. George	A Linear Time Implementation of the Reverse Cuthill-McKee Algorithm
CS-79-14	D.E. Morgan	Analysis of Closed Queueing Networks with Periodic Servers
CS-79-15*	M.H. van Emden G.J. de Lucena	Predicate Logic as a Language for Parallel Programming
CS-79-16*	J. Karhumäki I. Simon	A Note on Elementary Homorphisms and the Regularity of Equality Sets
CS-79-17*	K. Culik II J. Karhumäki	On the Equality Sets for Homomorphisms on Free Monoids with two Generators
CS-79-18*	F.E. Fich	Languages of R-Trivial and Related Monoids

* Out of print - contact author

CS-79-19*	D.R. Cheriton	Multi-Process Structuring and the Thoth Operating System
CS-79-20*	E.A. Ashcroft W.W. Wadge	A Logical Programming Language
CS-79-21*	E.A. Ashcroft W.W. Wadge	Structured LUCID
CS-79-22	G.B. Bonkowski W.M. Gentleman M.A. Malcolm	Porting the Zed Compiler
CS-79-23*	K.L. Clark M.H. van Emden	Consequence Verification of Flow-charts
CS-79-24*	D. Dobkin J.I. Munro	Optimal Time Minimal Space Selection Algorithms
CS-79-25*	P.R.F. Cunha C.J. Lucena T.S.E. Maibaum	On the Design and Specification of Message Oriented Programs
CS-79-26*	T.S.E. Maibaum	Non-Termination, Implicit Definitions and Abstract Data Types
CS-79-27*	D. Dobkin J.I. Munro	Determining the Mode
CS-79-28	T.A. Cargill	A View of Source Text for Diversely Configurable Software
CS-79-29*	R.J. Ramirez F.W. Tompa J.I. Munro	Optimum Reorganization Points for Arbitrary Database Costs
CS-79-30	A. Pereda R.L. Carvalho C.J. Lucena T.S.E. Maibaum	Data Specification Methods
CS-79-31*	J.I. Munro H. Suwanda	Implicit Data Structures for Fast Search and Update
CS-79-32*	D. Rotem J. Urrutia	Circular Permutation Graphs
CS-79-33*	M.S. Brader	PHOTON/532/Set - A Text Formatter
CS-79-34*	D.J. Taylor D.E. Morgan J.P. Black	Redundancy in Data Structures: Improving Software Fault Tolerance
CS-79-35	D.J. Taylor D.E. Morgan J.P. Black	Redundancy in Data Structures: Some Theoretical Results
CS-79-36	J.C. Beatty	On the Relationship between the LL(1) and LR(1) Grammars
CS-79-37	E.A. Ashcroft W.W. Wadge	R_x for Semantics

* Out of print - contact author

CS-79-38	E.A. Ashcroft W.W. Wadge	Some Common Misconceptions about LUCID
CS-79-39	J. Albert K. Culik II	Test Sets for Homomorphism Equivalence on Context Free Languages
CS-79-40	F.W. Tompa R.J. Ramirez	Selection of Efficient Storage Structures
CS-79-41*	P.T. Cox T. Pietrzykowski	Deduction Plans: A Basis for Intelligent Backtracking
CS-79-42	R.C. Read D. Rotem J. Urrutia	Orientations of Circle Graphs

* Out of print - contact author

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF WATERLOO

RESEARCH REPORTS 1980

Report No.	Author	Title
CS-80-01	P.T. Cox T. Pietrzykowski	On Reverse Skolemization
CS-80-02	K. Culik II	Homomorphisms: Decidability, Equality and Test Sets
CS-80-03	J. Brzozowski	Open Problems About Regular Languages
CS-80-04	H. Suwanda	Implicit Data Structures for the Dictionary Problem
CS-80-05	M.H. van Emden	Chess-Endgame Advice: A Case Study in Computer Utilization of Knowledge
CS-80-06	Y. Kobuchi K. Culik II	Simulation Relation of Dynamical Systems
CS-80-07	G.H. Gonnet J.I. Munro H. Suwanda	Exegesis of Self-Organizing Linear Search
CS-80-08	J.P. Black D.J. Taylor D.E. Morgan	An Introduction to Robust Data Structures
CS-80-09	J.Ll. Morris	The Extrapolation of First Order Methods for Parabolic Partial Differential Equations II
CS-80-10*	N. Santoro H. Suwanda	Entropy of the Self-Organizing Linear Lists
CS-80-11	T.S.E. Maibaum C.S. dos Santos A.L. Furtado	A Uniform Logical Treatment of Queries and Updates
CS-80-12	K.R. Apt M.H. van Emden	Contributions to the Theory of Logic Programming
CS-80-13	J.A. George M.T. Heath	Solution of Sparse Linear Least Squares Problems Using Givens Rotations
CS-80-14	T.S.E. Maibaum	Data Base Instances, Abstract Data Types and Data Base Specification
CS-80-15	J.P. Black D.J. Taylor D.E. Morgan	A Robust B-Tree Implementation
CS-80-16	K.O. Geddes	Block Structure in the Chebyshev-Padé Table
CS-80-17	P. Calamai A.R. Conn	A Stable Algorithm for Solving the Multi-facility Location Problem Involving Euclidean Distances

* In preparation

CS-80-18	R.J. Ramirez	Efficient Algorithms for Selecting Efficient Data Storage Structures
CS-80-19	D. Therien	Classification of Regular Languages by Congruences
CS-80-20	J. Buccino	A Reliable Typesetting System for Waterloo
CS-80-21	N. Santoro	Efficient Abstract Implementations for Relational Data Structures
CS-80-22	R.L. de Carvalho T.S.E. Maibaum T.H.C. Pequeno A.A. Pereda P.A.S. Veloso	A Model Theoretic Approach to the Theory of Abstract Data Types and Data Structures
CS-80-23	G.H. Gonnet	A Handbook on Algorithms and Data Structures
CS-80-24	J.P. Black D.J. Taylor D.E. Morgan	A Case Study in Fault Tolerant Software
CS-80-25	N. Santoro	Four 0(n**2) Multiplication Methods for Sparse and Dense Boolean Matrices
CS-80-26	J.A. Brzozowski	Development in the Theory of Regular Languages
CS-80-27	J. Bradford T. Pietrzykowski	The Eta Interface
CS-80-28	P. Cunha T.S.E. Maibaum	Resource = Abstract Data Type Data + Synchronization ...
CS-80-29	K. Culik II Arto Salomaa	On Infinite Words Obtained by Interating Morphisms
CS-80-30	T.F. Coleman A.R. Conn	Nonlinear Programming via an Exact Penalty Function: Asymptotic Analysis
CS-80-31	T.F. Coleman A.R. Conn	Nonlinear Programming via an Exact Penalty Function: Global Analysis
CS-80-32	P.R.F. Cunha C.J. Lucena T.S.E. Maibaum	Message Oriented Programming - A Resource Based Methodology
CS-80-33	Karel Culik II Tero Harju	Dominoes Over A Free Monoid