Equivalence Problems for Mappings on Infinite Strings

by

Karel Culik II and Jan Pachl

Research Report CS-80-41
Department of Computer Science
University of Waterloo

September 1980

Faculty of Mathematics

University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1
Equivalence problems for mappings on infinite strings*

by

Karel Culik II and Jan Pachl

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

* This research was supported by the National Sciences and Engineering Council of Canada, under Grant No. A 7403.
Abstract

This paper is concerned with sets of infinite strings (ω-languages) and mappings between them. The main result is that there is an algorithm for testing the (string by string) equality of two homomorphisms on an ω-regular set of infinite strings. As a corollary we show that it is decidable whether two functional finite-state transducers define the same function on infinite strings (are ω-equivalent).
0. Introduction

Infinite strings and sets of them (\(\omega\)-languages) have been extensively studied, see e.g. Eilenberg (1974), Cohen and Gold (1977). Finite transducers on infinite strings were considered by Boasson and Nivat (1979).

Our main interest here is testing the equivalence of finite transducers on infinite strings. We consider finite transducers with accepting states (for precise definition see Section 2). It is known that for finite transducers (rational transductions) the equivalence on finite strings is undecidable in general but decidable for functional transducers (Berstel, 1979). The latter result holds even when the domain is restricted to a context free language (Culik 1979). Our main goal is to prove analogous results for finite transducers working on infinite strings (but we do not consider any domain restriction). The equivalence of two finite transducers on finite strings implies also their \(\omega\)-equivalence, however the converse does not hold since two \(\omega\)-equivalent transducers might produce their outputs at different speeds. Thus \(\omega\)-equivalence is a necessary condition for equivalence but does not easily reduce to it.

In Section 2 we show some auxiliary results on deterministic \(\omega\)-regular languages. In the next section we extend the techniques from Salomaa (1978), Culik and Salomaa (1978) and Culik (1979) and show that the (string by string) equivalence of two homomorphisms on an \(\omega\)-regular set is decidable. Then we show that the \(\omega\)-equivalence problem for functional transducers reduces to it. Finally, we note that the undecidability of
the \( \omega \)-equivalence for nondeterministic finite transducers (or gsm) easily follows from the undecidability of their equivalence.

1. Preliminaries

Our basic terminology is a mixture of (Eilenberg, 1974) and (Cohen and Gold, 1977).

For a finite alphabet \( \Sigma \), let \( \Sigma^* \) be the set of finite strings over \( \Sigma \), \( \Sigma^\omega \) the set of infinite strings over \( \Sigma \), and \( \Sigma^\omega = \Sigma^* \cup \Sigma^\omega \). The empty string is denoted by \( \epsilon \), the length of a string \( w \) in \( \Sigma^* \) by \( |w| \).

We consider two classes of subsets of \( \Sigma^\omega \) (\( \omega \)-languages): \( \omega \)-regular and deterministic \( \omega \)-regular \( \omega \)-languages. We write a (nondeterministic) finite state automaton as \( M = (K, \Sigma, \delta, q_0, F) \), where \( K \) is a finite set of states, \( \Sigma \) is the input alphabet, \( \delta : K \times \Sigma \rightarrow 2^K \) is the transition function, \( q_0 \in K \) is the initial state and \( F \subseteq K \) is the set of final states. For \( \alpha \in \Sigma^\omega \), \( \alpha = a_0a_1a_2\ldots \), and an infinite sequence \( r = p_0p_1p_2\ldots \) of states \( p_i \in K \), we say that \( r \) is a run of \( M \) on \( \alpha \) if \( p_0 = q_0 \) and \( p_i \in \delta(p_{i-1}, a_{i-1}) \) for \( i \geq 1 \). For an infinite sequence of states \( r = p_0p_1p_2\ldots \), the set of states that appear infinitely many times in \( r \) is denoted by \( \text{INS}(r) \).

The \( \omega \)-language accepted by automaton \( M \) is defined as
\[
L^\omega(M) = \{ \alpha \in \Sigma^\omega \mid \text{there is a run } r \text{ on } \alpha \text{ such that } \text{INS}(r) \cap F \neq \emptyset \}.
\]

In the notation of Cohen and Gold (1977), \( L^\omega(M) = T_2(M) \); in the notation of Eilenberg (1974), \( L^\omega(M) = \|M\| \).
An \( \omega \)-language is \( \omega \)-regular (deterministic \( \omega \)-regular) if \( L = L^\omega(M) \) for some finite state (deterministic finite state) automaton \( M \). We refer to Eilenberg (1974) and Cohen and Gold (1977) for the basic properties of these two (distinct) classes of \( \omega \)-languages. A set \( L \subseteq \Sigma^\omega \) is \( \omega \)-regular (deterministic \( \omega \)-regular) if \( L \cap \Sigma^* \) is a regular language and \( L \cap \Sigma^\omega \) is an \( \omega \)-regular (deterministic \( \omega \)-regular) \( \omega \)-language.

A map \( g : \Sigma^\omega \rightarrow \Delta^\omega \) is a proper homomorphism if \( g(\Sigma) \subseteq \Delta^* \), \( g : \Sigma^* \rightarrow \Delta^* \) is a homomorphism and \( g(a_0a_1a_2... \in \Sigma^\omega) = g(a_0)g(a_1)g(a_2)... \) for every \( \omega \)-word \( a_0a_1a_2... \in \Sigma^\omega \). For \( g : \Sigma^\omega \rightarrow \Delta^\omega \), we write \( |g| = \max\{|g(a)| \mid a \in \Sigma\} \).

A proper homomorphism \( g : \Sigma^\omega \rightarrow \Delta^\omega \) is effectively given by listing \( g(a) \) for all \( a \in \Sigma \). An \( \omega \)-regular language \( L \) is effectively given by giving a finite state automaton \( M \) such that \( L = L^\omega(M) \).

When \( \alpha, \gamma \in \Sigma^\omega \), we write \( \alpha \leq \gamma \) to mean that \( \alpha \) is a prefix of \( \gamma \). When \( v \in \Sigma^* \), we denote the string \( vvv... \) by \( v^\omega \).

For basic notions in formal languages see Salomaa (1973).

2. Deterministic \( \omega \)-regular \( \omega \)-languages

Here we state three simple results that will be needed later. The first lemma is probably known but we have found no reference.

**Lemma 1** Deterministic \( \omega \)-regular \( \omega \)-languages are effectively closed under intersection.
Proof: Let \( M_i = (K_i, \Sigma, \delta_i, q_{i0}, F_i) \), \( i = 1, 2 \), be two deterministic finite state automata. Define \( M_3 = (K_3, \Sigma, \delta_3, q_{30}, F_3) \) by

\[
K_3 = \{0, 1, 2\} \times K_1 \times K_2
\]
\[
F_3 = \{2\} \times K_1 \times K_2
\]
\[
q_{30} = (0, q_{10}, q_{20})
\]
\[
\delta_3((j, q_1, q_2), a) = (j', q_1', q_2')
\]

if \( q_i' = \delta_i(q_i, a) \) for \( i = 1, 2 \)

and \( j' = \pi(j, q_1', q_2') \),

where the function \( \pi : \{0, 1, 2\} \times K_1 \times K_2 \to \{0, 1, 2\} \) is defined by

\[
\pi(2, p, q) = 0
\]
\[
\pi(0, p, q) = 0 \quad \text{if} \quad p \notin F_1
\]
\[
\pi(0, p, q) = 1 \quad \text{if} \quad p \in F_1
\]
\[
\pi(1, p, q) = 1 \quad \text{if} \quad q \notin F_2
\]
\[
\pi(1, p, q) = 2 \quad \text{if} \quad q \in F_2
\]

From the definition of \( \pi \) it follows that a run of \( M_3 \) on \( \alpha \in \Sigma^\omega \) enters a state in \( F_3 \) infinitely many times if and only if both the corresponding runs of \( M_1 \) and \( M_2 \) on \( \alpha \) enter final states infinitely many times.

Therefore \( L^\omega(M_3) = L^\omega(M_1) \cap L^\omega(M_2) \).
Lemma 2 Deterministic $\omega$-regular $\omega$-languages are effectively closed under union.


Lemma 3 If $g : \Sigma^\omega \to \Delta^\omega$ is a proper homomorphism then $g^{-1}(\Delta^\omega) \cap \Sigma^\omega$ is a deterministic $\omega$-regular set.

Proof: $g^{-1}(\Delta^\omega) \cap \Sigma^\omega = L^\omega(M)$ for this finite state automaton

$M = (K, \Sigma, \delta, q_0, F) : K = \{q_0, q_1\}, F = \{q_0\},$

$\delta(q_i, a) = q_1 \text{ if } g(a) = \varepsilon, \quad i = 1, 2 ;$

$\delta(q_i, a) = q_0 \text{ if } g(a) \neq \varepsilon, \quad i = 1, 2 .

3. $\omega$-equality Sets

Throughout this section we consider two fixed proper homomorphisms $g, h : \Sigma^\omega \to \Delta^\omega$; for $s$ in $\Sigma^*$ we define the balance of $s$ by

$\beta(s) = |g(s)| - |h(s)|$

where $|w|$ denotes the length of $w \in \Delta^*$. Let

$|\beta| = \max\{|\beta(a)| \mid a \in \Sigma\}$ .

The $\omega$-equality set is denoted by $E^\omega$ (or $E^\omega(g, h)$ if $g, h$ are not understood) and defined as
\[ E^\omega(g, h) = \{ \alpha \in \Sigma^\omega \mid g(\alpha) = h(\alpha) \} . \]

For \( k = 1, 2, 3, \ldots \) let

\[ E_k^\omega(g, h) = E_k^\omega = \left\{ \alpha \in E^\omega \mid |\beta(s)| \leq k \text{ for each } s \in \Sigma^*, s \leq \alpha \right\} \cup \]

\[ \bigcup_{0 < |v| \leq k} \left\{ t^\gamma \in E^\omega \mid t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \text{ for each } s \leq t, \right. \]

\[ \text{and } [g(\gamma) = v^\omega \text{ or } h(\gamma) = v^\omega] \}\right\}.\]

Clearly, \( E_1^\omega \subseteq E_2^\omega \subseteq \ldots \subseteq E^\omega \).

**Theorem 1** Each \( E_k^\omega(g, h) \) is a deterministic \( \omega \)-regular \( \omega \)-language.

Moreover, if \( k, g \) and \( h \) are effectively given then we can effectively give a deterministic finite state automaton \( M \) such that \( E_k^\omega(g, h) = L^\omega(M) \).

**Proof:** For \( v \in \Delta^* \), \( 0 < |v| \leq k \), let \( D(g, h, k, v) \) be the set of the \( \alpha \in \Sigma^\omega \) with these two properties:

1. for each finite \( s \leq \alpha \) we have \( g(s) \leq h(s) \) or \( h(s) \leq g(s) \);

2. either (2a) \( |\beta(s)| \leq k \) for each finite \( s \leq \alpha \)
   or (2b) \( \alpha = t^\gamma, \ t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \) for each \( s \leq t \), and \( g(s_1) \leq v^\omega \), \( h(ts_1) \leq g(t)v^\omega \) for each finite \( s_1 \leq \gamma \).

We shall show that each \( D(g, h, k, v) \) is (effectively) a deterministic \( \omega \)-regular \( \omega \)-language. Then \( E_k^\omega \) is the union of the sets
\[ \{ \alpha \in \Sigma^\omega \mid |\beta(s)| \leq k \text{ for each finite } s \leq \alpha \text{ and } g(s) = h(s) \text{ for infinitely many } s \leq \alpha \} , \]

\[ D(g, h, k, v) \cap g^{-1}(\Delta^\omega) \cap h^{-1}(\Delta^\omega) , \quad 0 < |v| \leq k , \text{ and } \]

\[ D(h, g, k, v) \cap g^{-1}(\Delta^\omega) \cap h^{-1}(\Delta^\omega) , \quad 0 < |v| \leq k . \]

The first set is deterministic \( \omega \)-regular (this follows from (Salomaa, 1978), Theorem 2.4); the remaining sets are deterministic \( \omega \)-regular by Lemmas 1 and 3. Hence the (effective) existence of \( M \) follows from Lemma 2.

The deterministic finite state automaton \( M_1 \) accepting \( D(g, h, k, v) \) works as follows: For an \( \omega \)-word \( \alpha \) on input, \( M_1 \) keeps comparing the values of \( g \) and \( h \) on finite prefixes \( s \) of \( \alpha \) as long as \( |\beta(s)| \leq k \). If and when \( |\beta(s)| \) exceeds \( k \), the tails of both \( g(\alpha) \) and \( h(\alpha) \) are matched against \( v^\omega \).

Formally, \( M_1 \) has five kinds of states:

(i) the state \( S(+, \varepsilon) = S(-, \varepsilon) \),

(ii) states \( S(+, w) \) and \( S(-, w) \) for each \( w \in \Delta^* \), \( 0 < |w| \leq k \),

(iii) states \( S(+, w, u) \) for \( w \in \Delta^* \), \( 0 < |w| \leq k \), \( u \leq v \),

(iv) states \( S(u_g, u_h) \) for \( u_g \leq v \), \( u_h \leq v \).

(v) the state \( S(*) \) (dead state).

All states of \( M_1 \) except \( S(*) \) are final; the initial state is \( S(+, \varepsilon) = S(-, \varepsilon) \).
The transition function $\delta$ is defined by

$\delta(S(+, w), a) = S(+, w')$ if $wg(a) = h(a)w'$

$\delta(S(+, w), a) = S(-, w')$ if $wg(a)w' = h(a)$

$\delta(S(+, w), a) = S(+, w', u)$ if $|wg(a)| > |h(a)| + k$, $g(a) = v^iu$

for some $i \geq 0$, $w = h(a)w'$, $w' \neq \varepsilon$

$\delta(S(+, w), a) = S(u_g, u_h)$ if $|wg(a)| - |h(a)| > k$, $g(a) = v^iu_g$ for some $i \geq 0$, $h(a) = wv^mu_h$ for some $m \geq 0$

$\delta(S(-, w), a) = S(+, w')$ if $wh(a)w' = g(a)$

$\delta(S(-, w), a) = S(-, w')$ if $wh(a) = g(a)w'$

$\delta(S(-, w), a) = S(u_g, u_h)$ if $|wh(a)| - |g(a)| > k$, $g(a) = v^iu_g$ for some $i \geq 0$, $wh(a) = v^mu_h$ for some $m \geq 0$

$\delta(S(+, w, u), a) = S(+, w', u')$ if $ug(a) = v^iu'$ for some $i \geq 0$, $w = h(a)w'$, $w' \neq \varepsilon$

$\delta(S(+, w, u), a) = S(u_g, u_h)$ if $ug(a) = v^iu_g$ for some $i \geq 0$, $h(a) = wv^mu_h$ for some $m \geq 0$

$\delta(S(u_g, u_h), a) = S(u_g', u_h')$ if $ug(a) = v^iu_g'$ for some $i \geq 0$, $u_h(a) = v^mu_h'$ for some $m \geq 0$.

All other values of $\delta$ are $S(*)$.

It follows from the construction that for $x \in \Sigma^*$ we have
\[ \delta(S(+, \varepsilon), x) = S(+, w) \text{ iff } g(x) = h(x)w \text{ and } |\beta(s)| \leq k \text{ for each } s \leq x \]

\[ = S(-, w) \text{ iff } g(x)w = h(x) \text{ and } |\beta(s)| \leq k \text{ for each } s \leq x \]

\[ = S(+, w, u) \text{ iff there is } t < x, t \text{ is the longest prefix of } x \text{ such that } |\beta(s)| \leq k \text{ for each } s \leq t, h(x)w = g(t) \text{ and } g(x) = g(t)v_i^u \text{ for some } i \geq 0 \]

\[ = S(u_g, u_h) \text{ iff there is } t < x, t \text{ is the longest prefix of } x \text{ such that } |\beta(s)| \leq k \text{ for each } s \leq t, g(x) = g(t)v_i^u \text{ for some } i \geq 0, \text{ and } h(x) = g(t)v_m^u \text{ for some } m \geq 0. \]

\[ = S(*) \text{ otherwise} \]

Hence \( D(g, h, k, v) = L^\omega(M_1) \).

\[ \square \]

Observe that, in terms of Cohen and Gold (1977), \( M \) also \( 1' \)-accepts \( D(g, h, k, v) \); hence \( D(g, h, k, v) \) is even a regular adherence set in the sense of Boasson and Nivat (1979).

Theorem 1 together with the following Theorem 2 enable us to deal with \( \omega \)-regular subsets of \( E^\omega_k \) effectively.

**Theorem 2**  If \( R \subseteq \Sigma^\omega \) is \( \omega \)-regular and \( R \subseteq E^\omega_k(g, h) \) then \( R \subseteq E^\omega_k(g, h) \) for some \( k \). More precisely, if \( R = L^\omega(M) \) for a finite state automaton with \( n \) states then \( R \subseteq E^\omega_k(g, h) \) where \( k = n \cdot \max(1, |\beta|, |g|, |h|) \).
Proof: Take $M$ such that $R = L^\omega(M)$ and $M$ has $n$ states; put $k = n \cdot \max(1, |\beta|, |g|, |h|)$. Choose any $\alpha \in R$. If $|\beta(s)| \leq k$ for each finite prefix $s$ of $\alpha$ then $\alpha \in E_k^\omega$. Otherwise, there is $y \leq \alpha$ such that $|\beta(y)| > k$; take the shortest such $y$ and write $\alpha = y\alpha_1$, $\alpha_1 \in \Sigma^\omega$. We have $|\beta| \cdot |y| \geq |\beta(y)| > k \geq n \cdot |\beta|$ and so $|y| > n$. It follows that $y = tuw$ with $0 < |u| \leq n$, $\beta(u) \neq 0$ and $tu^jw\alpha_1 \in R$ for each $j \geq 0$.

First assume $\beta(u) > 0$, i.e. $|g(u)| > |h(u)|$. Let $\gamma = uw\alpha_1$ and $v = g(u)$. If $h(u) = \varepsilon$ then $h(tw\alpha_1) = g(tu^jw\alpha_1)$ for all $j$, hence $g(\gamma) = v^w$ and therefore $\alpha \in E_k^\omega$. If $|h(u)| > 0$ then for every $s$ in $\Sigma^*$, $s \leq \gamma$ there are $j \geq 0$ and $x \in \Sigma^*$ such that $g(tu^{2j}) = h(tu^{2j}s)x$, $|h(t)| \leq j|h(u)|$, $g(t) \leq h(tu^j)$ and $|g(u)|$ divides $j$. Hence $j|h(u)| = i|g(u)|$ for some $i$, and by Lemma 4 below (with $a_1 = h(t)$, $b_1 = h(u^j)$, $c_1 = h(s)x$, $a_2 = g(t)$, $b_2 = g(u^i)$, $c_2 = g(u^{2(j-i)})$) we get $h(ts) \leq g(tu^{2j})$. Therefore $g(t\gamma) = h(t\gamma) \leq g(tu^\omega)$, and $g(\gamma) \leq g(u^\omega)$. Thus either $g(\gamma) = v^\omega$ (hence $\alpha \in E_k^\omega$) or $g(\gamma) \in \Delta^*$. We show that the latter is impossible: if $g(\gamma) \in \Delta^*$ then there are $s \in \Sigma^*$ and $\alpha_2 \in \Sigma^\omega$ such that $\alpha = ts\alpha_2$, $g(ts) = h(ts)$ and $g(\alpha_2) = h(\alpha_2) = \varepsilon$. Since $tus\alpha_2 \in R$, we get $g(tus) = h(tus)$, in contradiction with $|g(tus)| = |g(ts)| + |g(u)| > |h(ts)| + |h(u)| = |h(tus)|$.

Finally, if $\beta(u) < 0$, i.e. $|g(u)| < |h(u)|$, then the same argument shows that $h(uw\alpha_1) = h(u)^\omega$, hence $\alpha \in E_k^\omega$. 

\qed
Lemma 4 If $a_1 b_1 c_1 a_2 b_2 c_2 \in \Sigma^*$, $a_1 \leq a_2 b_2$, $a_2 \leq a_1 b_1$, $a_1 b_1 c_1 = a_2 b_2 c_2$ and $|b_1| = |b_2|$ then $a_1 c_1 = a_2 c_2$.

Proof: Without loss of generality, assume $a_1 \leq a_2$, i.e. $a_2 = a_1 w$, $w \in \Sigma^*$. Since $a_1 w \leq a_1 b_1$, we have $b_1 = wu$, $u \in \Sigma^*$ Next $a_1 b_1 \leq a_2 b_2$, so $b_2 = uv$, and $|v| = |w|$. Now

$$a_1 w w u v u c_1 = a_1 b_1 c_1 = a_2 b_2 c_2 = a_1 w u v b_2 c_2,$$

hence $v = w$. Since $a_1 w w u v c_1 = a_1 w w u w u c_2$, we get $c_1 = w c_2$ and $a_1 c_1 = a_1 w c_2 = a_2 c_2$.

\[\square\]

Corollaries to Theorems 1 and 2:

Corollary 1 If the set $E^\omega(g, h)$ is $\omega$-regular then it is deterministic $\omega$-regular.

\[\square\]

Corollary 2 For given finite state automaton $M$ and two homomorphisms $g$ and $h$ it is decidable whether or not $g(\alpha) = h(\alpha)$ for each $\alpha \in L^\omega(M)$.

Proof: By Theorem 2, $g(\alpha) = h(\alpha)$ for each $\alpha$ in $L^\omega(M)$ if and only if $L^\omega(M) \subseteq E_k^\omega$ where $k = n \cdot \max(1, |\beta|, |g|, |h|)$ and $n$ is the number of states in $M$. By Theorem 1, $E_k^\omega = L^\omega(M_1)$ for some (deterministic) finite state automaton $M_1$. The property $L^\omega(M) \subseteq L^\omega(M_1)$ is effectively testable by (Cohen and Gold, Th. 2.2.5).

\[\square\]
4. Finite State Transducers

In this section we look at the infinite behavior of finite state transducers. We write a (nondeterministic) finite state transducer as \( T = (K, \Sigma, \Delta, \delta, q_0, F) \), where \( K \) is a finite set of states, \( \Sigma \) is the input alphabet, \( \Delta \) is the output alphabet, \( \delta : K \times (\Sigma \cup \{\varepsilon\}) \to 2^{K \times \Delta^*} \) is the transition (and output) function, \( q_0 \in K \) is the initial state and \( F \subseteq K \) is the set of final states. If \( \alpha \in \Sigma^\omega \), \( \alpha = a_0 a_1 a_2 \ldots \), \( a_i \in \Sigma \cup \{\varepsilon\} \) for \( i \geq 0 \), \( \gamma \in \Delta^\omega \), \( \gamma = u_0 u_1 u_2 \ldots \), \( u_i \in \Delta^* \) for \( i \geq 0 \), and \( r = p_0 p_1 p_2 \ldots \) is an infinite sequence of states \( p_i \in K \), we say that \( r \) is a run of \( T \) on \( \alpha \) with output \( \gamma \) if \( p_0 = q_0 \) and \( (p_i, u_{i-1}) \in \delta(p_{i-1}, a_{i-1}) \) for \( i \geq 1 \); in symbols we write \( \alpha \xrightarrow{r} \gamma \).

Now we will introduce two relations \( R^\omega(T) \) and \( R^\omega(T) \) defined by \( T \):

\[
R^\omega(T) = \left\{ (\alpha, \gamma) \in \Sigma^\omega \times \Delta^\omega \mid \alpha \xrightarrow{r} \gamma \text{ for some run } r \text{ such that } \right.
\]
\[
\INS(r) \cap F \neq \emptyset \right\},
\]

\[
R^\omega(T) = R^\omega(T) \cap (\Sigma^\omega \times \Delta^\omega).
\]

The relation \( R^\omega(T) \) describes the behavior of transducer \( T \) on both finite and infinite strings, the relation \( R^\omega(T) \) on infinite strings only. Note, however, that the range of \( R^\omega(T) \) might contain finite strings, because \( T \) might read infinite input and produce empty output.
The analogue of Nivat's factorization theorem (Eilenberg 1974, Theorem IX.2.2) takes this form:

**Theorem 3**  For a set \( A \subseteq \Sigma^\omega \times \Delta^\omega \), two conditions are equivalent:

(i) \( A = R^\omega(T) \) for a finite state transducer \( T \);

(ii) there is an \( \omega \)-regular set \( B \subseteq \Gamma^\omega \) and two proper homomorphisms \( g : \Gamma^\omega \to \Sigma^\omega \) and \( h : \Gamma^\omega \to \Delta^\omega \) such that

\[
A = \{ (g(\gamma), h(\gamma)) \mid \gamma \in B \}.
\]

\( \square \)

The following lemma is easy to verify.

**Lemma 5**  Let \( g : \Sigma^\omega \to \Delta^\omega \) be a proper homomorphism. If \( R \subseteq \Sigma^\omega \) is \( \omega \)-regular then \( g(R) \subseteq \Delta^\omega \) is \( \omega \)-regular. If \( R \subseteq \Delta^\omega \) is \( \omega \)-regular then \( g^{-1}(R) \subseteq \Sigma^\omega \) is \( \omega \)-regular.

\( \square \)

**Theorem 4**  If \( T \) is a finite state transducer, then the domain and the range of \( R^\omega(T) \) are \( \omega \)-regular sets.

**Proof:**  Follows by Theorem 3 and Lemma 5.

\( \square \)

Moreover, the domain and the range of \( R^\omega(T) \) can be effectively given when \( T \) is effectively given (because the constructions in Theorem 3 and Lemma 5 are effective); this fact will be used in the proof of Theorem 5.

A finite state transducer \( T \) is called **functional** if the relation \( R^\omega(T) \subseteq \Sigma^\omega \times \Delta^\omega \) is a partial function from \( \Sigma^\omega \) to \( \Delta^\omega \).
Theorem 5. There is an algorithm to decide, for two given functional finite state transducers $T_1$ and $T_2$, whether $R^\omega(T_1) \subseteq R^\omega(T_2)$.

Proof: Let $T_i = (K_i, \Sigma, \Delta, \delta_i, q_{i0}, F_i)$, $i = 1, 2$. Let $\overline{\Delta}$ be another copy of $\Delta$, disjoint from $\Delta$; the correspondence between $\Delta^*$ and $(\overline{\Delta})^*$ will be written as $x \mapsto \overline{x}$, $x \in \Delta^*$. Define two proper homomorphisms $g, h : (\Delta \cup \overline{\Delta})^\infty \to \Delta^\infty$ by $g(a) = a$, $g(\overline{a}) = h(a) = \varepsilon$ and $h(\overline{a}) = a$ for $a \in \Delta$.

We shall construct a finite state transducer $T_3$ such that $R^\omega(T_1) \subseteq R^\omega(T_2)$ if and only if

(i) the domain of $R^\omega(T_1)$ is contained in the domain of $R^\omega(T_2)$,

and

(ii) $g(\alpha) = h(\alpha)$ for each $\alpha$ in the range of $R^\omega(T_3)$.

This together with Corollary 2 (in Section 3) and Theorem 4 establishes the result.

The transducer $T_3$ simulates the simultaneous operation of $T_1$ and $T_2$ and produces their two outputs intermixed (the outputs are then separated by $g$ and $h$).

Formally, we define $T_3 = (K_3, \Sigma, \Delta \cup \overline{\Delta}, \delta_3, q_{30}, F_3)$ where

- $K_3 = \{0,1,2\} \times K_1 \times K_2$
- $q_{30} = (0, q_{10}, q_{20})$
- $F_3 = \{2\} \times K_1 \times K_2$.

To define $\delta_3$, first define two other functions $\delta_4$ and $\delta_5$; the formula for $\delta_4$ employs the function $\pi : \{0,1,2\} \times K_1 \times K_2 \to \{0,1,2\}$ defined in the proof of Lemma 1.
Let
\[ \delta_4((j, q_1, q_2), a) = \{((\pi(j, q_1', q_2'), q_1', q_2'), x_1 x_2) \mid (q_1', x_1) \in \delta_1(q_1, a) \text{ for } i = 1, 2\} ,\]
a \in \Sigma \cup \{ \epsilon \} ;
\[ \delta_5((j, q_1, q_2), a) = \phi \quad \text{for } a \in \Sigma ;\]
\[ \delta_5((0, q_1, q_2), \epsilon) = \{((j, q_1', q_2), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \epsilon) \text{ and either } [q_1' \notin F_1 \text{ and } j = 0] \text{ or } [q_1' \in F_1 \text{ and } j = 1] \}
\cup \{((0, q_1, q_2'), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \epsilon)\} ;\]
\[ \delta_5((1, q_1, q_2), \epsilon) = \{((1, q_1', q_2), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \epsilon)\}
\cup \{((1, q_1', q_2), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \epsilon)\}
\text{ and either } [q_2' \notin F_2 \text{ and } j = 1] \text{ or } [q_2' \in F_2 \text{ and } j = 2] \};\]
\[ \delta_5((2, q_1, q_2), \epsilon) = \{((0, q_1', q_2), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \epsilon)\}
\cup \{((0, q_1, q_2'), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \epsilon)\} .\]

Finally, let
\[ \delta_3(p, a) = \delta_4(p, a) \cup \delta_5(p, a) \quad \text{for } p \in K_3 , \ a \in \Sigma \cup \{ \epsilon \} .\]
The definition of $\delta_3$ and $F_3$ ensures that if
\[ r = (j_0, q_{10}, q_{20})(j_1, q_{11}, q_{21})(j_2, q_{12}, q_{22}) \ldots \] is a run of $T_3$ on
$\alpha \in \Sigma^\omega$ and $r_1$ and $r_2$ are the corresponding runs of $T_1$ and $T_2$ on
$\alpha$, then $\text{INS}(r) \cap F_3 \neq \emptyset$ iff $\text{INS}(r_1) \cap F_1 \neq \emptyset$ and
$\text{INS}(r_2) \cap F_2 \neq \emptyset$. Therefore $T_3$ has the desired properties and the
proof is completed.

\[ \square \]

**Corollary 3**

Given two functional finite transducers $T_1$ and $T_2$, it
is decidable whether $R^\omega(T_1) = R^\omega(T_2)$.

**Proof:** The decidability of inclusion implies the decidability of
equality.

\[ \square \]

We say that two finite transducers $T_1$ and $T_2$ are
$\omega$-equivalent if $R(\omega)(T_1) = R(\omega)(T_2)$.

**Corollary 4**

The $\omega$-equivalence problem for functional finite
transducers is decidable.

**Proof:** Follows from Corollary 3 and the following lemma, which
says that finite transducers on infinite strings are closed under the
restriction to an $\omega$-regular set.

\[ \square \]

**Lemma 6**

Let $T = (K, \Sigma, \Delta, \delta, q_0, F)$ be a finite transducer and
$L \subseteq \Sigma^\omega$ an $\omega$-regular set. Then we can construct a finite transducer
$T_1$ such that $R^\omega(T_1) = R^\omega(T) \cap (L \times \Delta^\omega)$.
Proof: As in Theorem 3, find \( \Gamma, g : \Gamma^\omega \to \Sigma^\omega, h : \Gamma^\omega \to \Delta^\omega \) and \( B \subseteq \Gamma^\omega \) such that \( R^\omega(T) = \{(g(\gamma), h(\gamma)) \mid \gamma \in B\} \). The set \( g^{-1}(L) \subseteq \Gamma^\omega \) is \( \omega \)-regular by Lemma 5, and \( B \cap g^{-1}(L) \subseteq \Gamma^\omega \) is \( \omega \)-regular because \( \omega \)-regular sets are closed under intersection (Eilenberg, 1974; Chapter XIV). Since \( R^\omega(T) \cap (L \times \Delta^\omega) = \{(g(\gamma), h(\gamma)) \mid \gamma \in B \cap g^{-1}(L)\} \), the result follows by Theorem 3.

\[ \square \]

We conclude this section by showing that the \( \omega \)-equivalence is undecidable for general finite transducers, even for gsm as defined in (Salomaa, 1973).

**Theorem 6** The \( \omega \)-equivalence problem for (nondeterministic) gsm is undecidable.

Proof: Consider any gsm \( M = (K, \Sigma, \Delta, \delta, q_0, F) \). Modify it to \( M' \) so that \( M' = (\mathcal{KU}(f), \Sigma \cup \#, \Delta \cup \#, \delta', q_0, \{f\}) \) where \( f \notin K \), \( \# \notin \Sigma \cup \Delta \), and \( \delta \) is extended to \( \delta' \) by \( (f, \#) \in \delta'(q, \#) \) for each \( q \in F \), \( (f, \#) \in \delta'(f, \#) \).

Clearly \( M_1 \) and \( M_2 \) are equivalent iff \( M'_1 \) and \( M'_2 \), constructed as above, are \( \omega \)-equivalent. That completes the proof since the equivalence for (nondeterministic) gsm is undecidable (Berstel, 1979).

\[ \square \]
References

Berstel, J. (1979), Transductions and context-free languages; B.G. Teubner, Stuttgart.


Culik, K. II (1979), Some decidability results about regular and pushdown translations; Information Processing Letters 8, 5-8.

