Equivalence Problems for Mappings on Infinite Strings*

KAREL CULIK II AND JAN K. PACHL

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

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 (ω-languages) have been extensively studied, see, e.g., Eilenberg (1974) and 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 ω-equivalence, however, the converse does not hold since two ω-equivalent transducers might produce their outputs at different speeds. Thus ω-equivalence is a necessary condition for equivalence but does not easily reduce to it.

In Section 2 we show some auxiliary results on deterministic ω-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 ω-regular set is decidable. Then we show that the ω-equivalence problem for functional

* This research was supported by the Natural Sciences and Engineering Council of Canada, under Grant No. A7403.
transducers reduces to it. Finally, we note that the undecidability of the ω-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 Σ, let Σ* be the set of finite strings over Σ, Σω the set of infinite strings over Σ, and Σω = Σ* ∪ Σω. The empty string is denoted by ε, the length of a string w in Σ* by |w|.

We consider two classes of subsets of Σω (ω-languages): ω-regular and deterministic ω-regular ω-languages. We write a (nondeterministic) finite-state automaton as 
M = (K, Σ, δ, q0, F), where K is a finite set of states, Σ is the input alphabet, δ: K × Σ → 2K is the transition function, q0 ∈ K is the initial state and F ⊆ K is the set of final states. For a ∈ Σω, a = a0a1a2..., and an infinite sequence r = p0p1p2..., of states Pi ∈ K, we say that r is a run of M on a if p0 = q0 and Pi ∈ δ(pi−1, ai−1) for i ≥ 1. For an infinite sequence of states r = p0p1p2..., the set of states that appear infinitely many times in r is denoted by INS(r). The ω-language accepted by automaton M is defined as

Lω(M) = {a ∈ Σω | there is a run r on a such that INS(r) ∩ F ≠ ∅}.

In the notation of Cohen and Gold (1977), Lω(M) = T2(M); in the notation of Eilenberg (1974), Lω(M) = |M|.

An ω-language is ω-regular (deterministic ω-regular) if L = Lω(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 ω-languages. A set L ⊆ Σω is ω-regular (deterministic ω-regular) if L ∩ Σ* is a regular language and L ∩ Σω is an ω-regular (deterministic ω-regular) ω-language.

A map g: Σω → Aω is a proper homomorphism if g(Σ) ⊆ A*, g: Σ* → A* is a homomorphism and g(a0a1a2...) = g(a0)g(a1)g(a2) for every ω-word a0a1a2... ∈ Σω. For g: Σω → Aω, we write |g| = max{||g(a)|| | a ∈ Σ}.

A proper homomorphism g: Σω → Aω is effectively given by listing g(a) for all a ∈ Σ. An ω-regular language L is effectively given by giving a finite-state automaton M such that L = Lω(M).

When α, γ ∈ Σω, we write α ≤ γ to mean that α is a prefix of γ. When v ∈ Σ*, we denote the string vv... by vω.

For basic notions in formal languages see Salomaa (1973).
2. Deterministic ω-Regular ω-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 ω-regular ω-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

\[
\begin{align*}
K_3 &= \{0, 1, 2\} \times K_1 \times K_2, \\
F_3 &= \{2\} \times K_1 \times K_2, \\
q_{30} &= (0, q_{10}, q_{20}), \\
\delta_3((j, q_1, q_2), a) &= (j', q_1', q_2') \quad \text{if} \quad q_i' = \delta_i(q_i, a) \\
&\quad \text{for} \quad i = 1, 2 \quad \text{and} \quad j' = \pi(j, q_1', q_2'),
\end{align*}
\]

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

\[
\begin{align*}
\pi(2, p, q) &= 0, \\
\pi(0, p, q) &= 0 \quad \text{if} \quad p \not\in F_1, \\
&= 1 \quad \text{if} \quad p \in F_1, \\
\pi(1, p, q) &= 1 \quad \text{if} \quad q \not\in F_2, \\
&= 2 \quad \text{if} \quad q \in F_2.
\end{align*}
\]

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 ω-regular ω-languages are effectively closed under union.


**Lemma 3.** If \( \varphi: \Sigma^\omega \to \Delta^\omega \) is a proper homomorphism then \( \varphi^{-1}(\Delta^\omega) \cap \Sigma^\omega \) is a deterministic ω-regular set.
Proof. \( g^{-1}(\alpha) \cap \Sigma^\omega = L^\omega(M) \) for this finite-state automaton \( M = (K, \Sigma, \delta, q_0, F) \): \( K = \{q_0, q_1\}, F = \{q_0\} \),

\[
\delta(q_i, a) = q_1 \quad \text{if} \quad g(a) = \epsilon, \quad i = 0, 1;
\]

\[
\delta(q_i, a) = q_0 \quad \text{if} \quad g(a) \neq \epsilon, \quad i = 0, 1. \quad \blacksquare
\]

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 (of \( g \) and \( h \)) is denoted by \( E^\omega(g, h) \) (or \( E^\omega(g, h) \) if \( g, h \) are not understood) and defined as

\[
E^\omega(g, h) = \{a \in \Sigma^\omega \mid g(a) = h(a)\}.
\]

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

\[
E^\omega_k(g, h) = \left\{ a \in E^\omega \mid |\beta(s)| \leq k \text{ for each } s \in \Sigma^*, s \leq a \right\} \cup \\
\bigcup_{0 < |\nu| \leq k} \{ t \gamma \in E^\omega \mid t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \text{ for each } s \leq t, \text{ and } g(\gamma) = \nu^\omega \text{ or } h(\gamma) = \nu^\omega \}.
\]

Clearly, \( E^\omega_1 \subseteq E^\omega_2 \subseteq \cdots \subseteq E^\omega \).

**Theorem 1.** Each \( E^\omega_k(g, h) \) 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 \( E^\omega_k(g, h) = L^\omega(M) \).

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

1. for each finite \( s \leq a \) we have

\[
g(s) \leq h(s) \quad \text{or} \quad h(s) \leq g(s);
\]

2. either

\[
|\beta(s)| \leq k \text{ for each finite } s \leq a
\]
or

\[(2b) \quad \alpha = t_\gamma, t \in \Sigma^*, \gamma \in \Sigma^\omega, |\beta(s)| \leq k \text{ for each } s \leq t, \text{ and } g(s_1) \leq \nu^\omega, h(ts_1) \leq g(t)\nu^\omega \text{ for each finite } s_1 \leq \gamma.\]

We shall show that each \(D(g, h, k, \nu)\) is (effectively) a deterministic \(\omega\)-regular \(\omega\)-language. Then \(E_k^\omega\) is the union of the sets

\[\{\alpha \in \Sigma^\omega | |\beta(s)| \leq k \text{ for each finite } s \leq \alpha \text{ and } g(s) = h(s) \text{ for infinitely many } s \leq \alpha\},\]

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

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

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 \(D(g, h, k, \nu)\) 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 \(|\mu(s)| \leq k\). If and when \(|\beta(s)|\) exceeds \(k\), the tails of both \(g(\alpha)\) and \(h(\alpha)\) are matched against \(\nu^\omega\).

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

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

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

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

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

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

All states of \(M_1\) except \(S(\ast)\) are final; the initial state is \(S(+, \epsilon) = S(-, \epsilon)\).

When running on an infinite string \(\alpha\), the automaton moves through the states \(S(\pm, w)\) as long as all finite prefixes \(p\) of \(\alpha\) encountered satisfy \(|\beta(p)| \leq k\); the "buffer" \(w\) records the difference between \(g(p)\) and \(h(p)\) (the sign "+" means \(h(p) \leq g(p)\), the sign "-" means \(g(p) \leq h(p)\)).

If there is a prefix \(p\) of \(\alpha\) such that \(|\beta(p)| > k\), then let \(t\) be the longest prefix of \(\alpha\) such that \(|\beta(s)| \leq k\) for all \(s \leq t\). As the automaton reads beyond \(t\), it moves to \(S(+, w, u)\) or \(S(u_g, u_h)\), depending on whether \(h(p) < g(t)\) or \(g(t) \leq h(p)\) for the prefix \(p\) read so far. The states \(S(u_g, u_h)\) trace the progress of \(g(p)\) and \(h(p)\) through the loop \(\nu^\omega\).
The transition function \( \delta \) is defined by

\[
\delta(S(+, w), a) = S(+, w') \quad \text{if} \quad wg(a) = h(a)w';
\]
\[
\delta(S(+, w), a) = S(-, w') \quad \text{if} \quad wg(a)w' = h(a); 
\]
\[
\delta(S(+, w), a) = S(+, w', u) \quad \text{if} \quad |wg(a)| > |h(a)| + k, \quad g(a) = v'u \\
\text{for some } i > 0, \quad w = h(a)w', \quad w' \neq \varepsilon; 
\]
\[
\delta(S(+, w), a) = S(u_g, u_h) \quad \text{if} \quad ||wg(a)| - |h(a)|| > k, \quad g(a) = v'u_g \\\n\text{for some } i > 0, \quad h(a) = wv^mu_h \text{ for some } m > 0; 
\]
\[
\delta(S(-, w), a) = S(+, w') \quad \text{if} \quad wh(a)w' = g(a); 
\]
\[
\delta(S(-, w), a) = S(-, w') \quad \text{if} \quad wh(a) = g(a)w'; 
\]
\[
\delta(S(-, w), a) = S(u_g, u_h) \quad \text{if} \quad ||wh(a)| - |g(a)|| > k, \quad g(a) = v'u_g \\\n\text{for some } i > 0, \quad wh(a) = v^mu_h \text{ for some } m > 0; 
\]
\[
\delta(S(+, w, u), a) = S(+, w', u') \quad \text{if} \quad ug(a) = v'w' \text{ for some } i > 0, \\
\text{for } w = h(a)w', \quad w' \neq \varepsilon; 
\]
\[
\delta(S(+, w, u), a) = S(u_g, u_h) \quad \text{if} \quad ug(a) = v'u_g \text{ for some } i > 0, \\
h(a) = wv^mu_h \text{ for some } m > 0; 
\]
\[
\delta(S(u_g, u_h), a) = S(u'_g, u'_h) \quad \text{if} \quad u_gg(a) = v'u'_g \text{ for some } i > 0, \\
u_hh(a) = v^mu'_h \text{ for some } m > 0; 
\]

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

It follows from the construction that for \( x \in \Sigma^* \) we have

\[
\delta(S(+, \varepsilon), x) = S(+, w) \quad \text{iff} \quad g(x) = h(x)w \text{ and } |\beta(s)| \leq k \\
\text{for each } s \leq x; 
\]
\[
= S(-, w) \quad \text{iff} \quad g(x)w = h(x) \text{ and } |\beta(s)| \leq k \\
\text{for each } s \leq x; 
\]
\[
= S(+, w, u) \quad \text{iff there is } t < x, t \text{ is the longest} \\
\text{prefix of } x \text{ such that } |\beta(s)| \leq k \\
\text{for each } s \leq t, \quad h(x)w = g(t) \text{ and} \\
g(x) = g(t)v'u \text{ for some } i > 0; 
\]
\[
= S(u_g, u_h) \quad \text{iff there is } t < x, t \text{ is the longest} \\
\text{prefix of } x \text{ such that } |\beta(s)| \leq k \\
\text{for each } s \leq t, \\
g(x) = g(t)v'u_g \text{ for some } i > 0, \text{ and} \\
h(x) = g(t)v^mu_h \text{ for some } m > 0; 
\]
\[
= S(*) \quad \text{otherwise.} 
\]

Hence \( D(g, h, k, v) = L^\omega(M_1) \).
Observe that, in the terminology 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).

Now, we show a technical lemma which will be needed in the proof of the next theorem.

**Lemma 4.** If $a_1b_1c_1a_2b_2c_2 \in \Sigma^*$, $a_1 \leq a_2b_2$, $a_2 \leq a_1b_1$, $a_1b_1^2c_1 = a_2b_2^2c_2$ and $|b_1| = |b_2|$ then $a_1c_1 = a_2c_2$.

**Proof.** Without loss of generality, assume $a_1 \leq a_2$, i.e., $a_2 = a_1w$, $w \in \Sigma^*$. Since $a_1w \leq a_1b_1$, we have $b_1 = uw$, $u \in \Sigma^*$. Next $a_1b_1 \leq a_2b_2$, so $b_2 = uv$, and $|v| = |w|$. Now

$$a_1wuwuc_1 = a_1b_1^2c_1 = a_2b_2^2c_2 = a_1wuvb_2c_2,$$

hence $v = w$. Since $a_1wuwuc_1 = a_1wuwuc_2$, we get $c_1 = w$ and $a_1c_1 = a_1wc_2 = a_2c_2$. \[\square\]

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

**Theorem 2.** If $R \subseteq \Sigma^\omega$ is $\omega$-regular and $R \subseteq E^\omega(g, h)$ 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 $M = (K, \Sigma, \delta, q_0, F)$ such that $R = L^\omega(M)$; let $k = n \cdot \max(1, |\beta|, |g|, |h|)$, where $n = \text{card}(K)$. Consider any $a \in R$. If $|\beta(s)| \leq k$ for each finite prefix $s$ of $a$, then $a \in E^\omega_k$; otherwise find the longest finite prefix $t$ of $a$ such that $|\beta(s)| \leq k$ for each $s \leq t$. Write $a = t\gamma$ and let $\alpha$ be the first letter of $\gamma$; clearly $|\beta(t\alpha)| > k$ by the choice of $t$. We shall assume, without loss of generality, that $\beta(ta) > k$ (the case $\beta(ta) < -k$ is symmetrical), and prove that $g(\gamma) = v^\omega$ for a string $v$ such that $0 < |v| \leq k$.

This will prove that $a \in E^\omega_k$.

We have $\beta(t) + |\beta(a)| \geq |\beta(ta)| > k \geq n \cdot |\beta|$ and $|\beta(a)| \leq |\beta|$, hence $\beta(t) > (n - 1) \cdot |\beta|$. Let $r = p_0p_1p_2 \ldots$ be a run of $M$ on $a$ such that $\text{INS}(r) \cap F \neq \emptyset$; then $p_{|t|} \in \delta(q_0, t)$, and $z_\gamma \in R$ whenever $z \in \Sigma^*$ is such that $p_{|t|} \in \delta(q_0, z)$.

Since $\beta(t) > (n - 1) \cdot |\beta|$, we can find $u, w, x \in \Sigma^*$ and $q \in K$ such that $|w| \leq n$, $\beta(w) > 0$, $q \in \delta(q_0, u) \cap \delta(q, w)$ and $p_{|t|} \in \delta(q, x)$. It follows that $uw^jx^j \in R$, and therefore also $uw^jx^j \gamma \in E^\omega(g, h)$, for each $j \geq 0$.

We distinguish two cases:

**Case I.** $h(w) = e$. Then $h(uxy) = g(uwx^j\gamma)$ for all $j$, hence $g(uxy) = g(uwx^j\gamma)$ for all $j$, hence $g(x^j\gamma) = g(w)^\omega$. Thus $g(\gamma) = v^\omega$ for some $v$ with $|v| = |g(w)|$. Moreover, $0 < |v| \leq k$ because $0 < \beta(w) = |g(w)| \leq |g| \cdot |w| \leq k$. 


Case II. $h(w) \neq \varepsilon$. Since $\beta(w) > 0$, for every finite $s \leq \gamma$ there are $j \geq 0$ and $s'$ such that $g(uw^i) = h(uw^i)xs'$, $|h(u)| \leq j|h(w)|$, $g(u) \leq h(uw^i)$ and $|g(w)|$ divides $j$. Thus $j|h(w)| = i|g(w)|$ for some $i$, and we get $h(uxs) \leq g(uw^i)$ by using Lemma 4 with $a_1 = h(u)$, $b_1 = h(w)$, $c_1 = h(xs)s'$, $a_2 = g(u)$, $b_2 = g(w)$, $c_2 = g(w^{j(i-1)})$.

Therefore $g(uxy) = h(uxy) \leq g(uw^i)$, and $g(xy) \leq g(w)$. Thus either $g(y) = \varepsilon$ for some $s \in D^*$, or $g(y) \in D^*$. We show that the latter is impossible: if $g(y) = \varepsilon$ for some $s \in D^*$, then there are $s \in \Sigma^*$ and $s \in \Sigma^*$ such that $y = s\gamma$, $g(uxs) = h(uxs)$ and $g(y') = h(y') = \varepsilon$. Since $uwx = uwx\gamma \in R$, we also get $g(uwxs) = h(uwxs)$, in contradiction with $|g(uwxs)| = |g(uxs)| + |g(w)| > |h(uwxs)|$.

This completes the proof.

Corollaries to Theorems 1 and 2:

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

**Corollary 2.** Given a finite-state automaton $M$ and two homomorphisms $g$ and $h$, it is decidable whether or not $g(a) = h(a)$ for each $a \in L_\omega(M)$.

**Proof.** By Theorem 2, $g(a) = h(a)$ for each $a \in L_\omega(M)$ if and only if $L_\omega(M) \subseteq E_\omega^k$, where $k = n \cdot \max(1, |\beta|, |g|, |h|)$ and $n$ is the number of states in $M$. By Theorem 1, $E_\omega^k = L_\omega(M_1)$ 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, Theorem 2.2.5).

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, $\delta$ is the transition (and output) function from $K \times (\Sigma \cup \{\varepsilon\})$ into the set of finite subsets of $K \times \Delta^*$, $q_0 \in K$ is the initial state and $F \subseteq K$ is the set of final states. If $a \in \Sigma^\omega$, $a = a_0a_1a_2...$, $a_i \in \Sigma \cup \{\varepsilon\}$ for $i \geq 0$, $\gamma \in \Delta^\omega$, $\gamma = u_0u_1u_2...$, $u_i \in \Delta^*$ for $i \geq 0$, and $r = p_0p_1p_2...$ is an infinite sequence of states $p_i \in K$, we say that $r$ is a run of $T$ on $a$ 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 $a \Rightarrow r \gamma$.

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

$$R^\omega(T) = \{(a, \gamma) \in \Sigma^\omega \times \Delta^\omega \mid a \Rightarrow r \gamma \text{ for some run } r \text{ such that } \INS(r) \cap F \neq \emptyset\},$$

$$R^\omega(T) = R^\omega(T) \cap (\Sigma^\omega \times \Delta^\omega).$$
The relation $R^\omega(T)$ describes the behavior of transducer $T$ on both finite and infinite input strings, the relation $R^\omega(T)$ on infinite input 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

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

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.

**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.

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 $\bar{\Delta}$ be another copy of $\Delta$, disjoint from $\Delta$; the correspondence between $\Delta^*$ and $(\bar{\Delta})^*$ will be written as $x \mapsto \bar{x}$, $x \in \Delta^*$. Define two proper homomorphisms $g, h: (\Delta \cup \bar{\Delta})^\omega \to \Delta^\omega$ by $g(a) = a$, $g(\bar{a}) = h(a) = \varepsilon$ and $h(\bar{a}) = a$ 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) $g(a) = h(a)$ for each $a$ 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 outputs intermixed (the outputs are then separated by $g$ and $h$). On input $a \in \Sigma$, $T_3$ outputs $x_1x_2$, where $x_1$ is the output of $T_1$ and $x_2$ is the output of $T_2$. The situation is more complicated with input $\varepsilon$: here it can happen that only $T_1$ or only $T_2$ moves (this possibility is taken care of by the function $\delta_5$ defined below).

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

$$K_3 = \{0, 1, 2\} \times K_1 \times K_2,$$

$$q_{30} = (0, q_{10}, q_{20}),$$

$$F_3 = \{2\} \times K_1 \times K_2.$$

Let $\delta_3(p, a) = \delta_4(p, a) \cup \delta_5(p, a)$ for $p \in K_3$ and $a \in \Sigma \cup \{\varepsilon\}$, where $\delta_4$ and $\delta_5$ are defined as follows (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):

\[
\delta_4((j, q_1, q_2), a) = \{((\pi(j, q_1', q_2'), q_1', q_2'), x_1x_2) \mid (q_i', x_i) \in \delta_i(q_i, a) \text{ for } i = 1, 2\},
\]

\[a \in \Sigma \cup \{\varepsilon\};\]

\[\delta_5((j, q_1, q_2), a) = \emptyset \quad \text{for } a \in \Sigma;\]

\[\delta_5((0, q_1, q_2), \varepsilon) = \{((j, q_1', q_2), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \varepsilon) \text{ and either} \]

\[q_1' \not\in F_1 \text{ and } j = 0 \text{ or } [q_1' \in F_1 \text{ and } j = 1]\]

\[\cup \{((0, q_1', q_2'), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \varepsilon)\};\]

\[\delta_5((1, q_1, q_2), \varepsilon) = \{((1, q_1', q_2), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \varepsilon)\}
\]

\[\cup \{((j, q_1, q_2'), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \varepsilon) \text{ and either} \]

\[q_2' \not\in F_2 \text{ and } j = 1 \text{ or} \]

\[q_2' \in F_2 \text{ and } j = 2\}\];

\[\delta_5((2, q_1, q_2), \varepsilon) = \{((0, q_1, q_2'), x_1) \mid (q_1', x_1) \in \delta_1(q_1, \varepsilon)\}
\]

\[\cup \{((0, q_1', q_2), x_2) \mid (q_2', x_2) \in \delta_2(q_2, \varepsilon)\}.\]

The definition of $\delta_3$ and $F_3$ ensures that if $r = (j_0, q_{10}, q_{20})(j_1, q_{11}, q_{21}) (j_2, q_{12}, q_{22})...$ is a run of $T_3$ on $a \in \Sigma^\infty$ and $r_1$ and $r_2$ are the corresponding runs of $T_1$ and $T_2$ on $a$, 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. \[\blacksquare\]

**Corollary 3.** Given two functional finite transducers $T_1$ and $T_2$, it is decidable whether $R^\infty(T_1) = R^\infty(T_2)$. 
Proof. The decidability of inclusion implies the decidability of equality.  

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 string are closed under the restriction to an $\omega$-regular set.

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 $R^\omega(T) = \{(g(\gamma), h(\gamma)) | \gamma \in B\}$. 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, Chap. XIV). Since 

$$R^\omega(T) \cap (L \times \Delta^\omega) = \{(g(\gamma), h(\gamma)) | \gamma \in B \cap g^{-1}(L)\},$$

the result follows by Theorem 3.

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 

$$M' = (K \cup \{f\}, \Sigma \cup \{\#\}, \Delta \cup \{\#\}, \delta', q_0, \{f\})$$

where $f \in K$, $\# \in \Sigma \cup \Delta$, 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).

Received: September 27, 1980; Revised: April 17, 1981

References

COHEN, R. S. AND GOLD, A. Y. (1977), Theory of $\omega$-languages. I. Characterization of $\omega$-
CULIK, K. II (1979), Some decidability results about regular and pushdown translations,
SALOMAA, A. (1978), Equality sets for homomorphisms of free monoids, Acta Cybernet. 4,
127–139.