THE UNSOLVABILITY OF THE EQUIVALENCE PROBLEM FOR ε-FREE NGSM'S WITH UNARY INPUT (OUTPUT) ALPHABET AND APPLICATIONS*

Oscar H. Ibarra
Computer Science Department
University of Minnesota
Minneapolis, Minnesota 55455

It is shown that the equivalence problem is unsolvable for ε-free nondeterministic generalized sequential machines whose input/output are restricted to unary/binary (binary/unary) alphabets. This strengthens a known result of Griffiths. Applications to some decision problems concerning right-linear grammars and directed graphs are also given.

1. Introduction

The equivalence problem for deterministic generalized sequential machines is decidable. (In fact, the equivalence problem is solvable for deterministic sequential transducers [1,3]). It is also obvious that the equivalence problem for complete nondeterministic generalized sequential machines is decidable. (These are machines which output exactly one symbol per move.) However, the equivalence problem for ε-free (not having the null string ε as output) nondeterministic generalized sequential machines is unsolvable. This result was shown by Griffiths [4] who also observed (as a corollary) that the equivalence problem for c-finite languages [3] is undecidable. In [2], the result was used to show the unsolvability of the equivalence problem for sentential forms of context-free grammars.

In this paper, we strengthen Griffiths's result. Specifically, we show that the equivalence problem for ε-free nondeterministic generalized sequential machines is unsolvable even if we restrict the input/output to unary/binary (respectively, binary/unary) alphabets. This result which is somewhat surprising clearly demonstrates the complexity that nondeterminism can introduce even in very simple computing devices. Related results are also obtained. For example, it is proved that there is no algorithm to determine for 2 right-linear grammars G_1 and G_2 all of whose rules are of the form $A \to xB$ or $A \to x$ (A, B are nonterminals, x is a non-null binary terminal string) whether for each binary string y and $n \geq 1$, y is derivable in G_1 in n steps if and only if y is derivable in G_2 in n steps. In fact, the result holds even if the rules $A \to xB$ and $A \to x$ are restricted so that the length of x is 2, 3, or 6. Another result concerns directed graphs. Let $G = \langle V,E,v_0,f,g\rangle$ be a directed graph, where V is a finite nonempty set of vertices, E is a finite nonempty set of ordered pairs $\langle u,v\rangle$ of distinct vertices called edges, v_0 is a distinguished vertex called the source vertex, and f and g are functions from E into $\{0,1\}$ and $\{1,2,3\}$, respectively. Let $R(G) = \{(x,c) \mid x = a_1 \cdots a_n,\ n \geq 1$, each a_i in $\{0,1\}$, there exist edges $\langle u_1,u_2\rangle,\ldots,\langle u_n,u_{n+1}\rangle$ such that $u_1 = v_0$, $f(\langle u_i,u_{i+1}\rangle) = a_i$ for $1 \leq i \leq n$, and $c = \sum_1^n g(\langle u_i,u_{i+1}\rangle)\}$. It is shown that it is recursively unsolvable to determine for arbitrary directed graphs $G_i = \langle V_i,E_i,v_{0i},f_i,g_i\rangle$, $i = 1,2$, whether $R(G_1) = R(G_2)$.

The proofs are facilitated by considering a more general type of machine which we now define.

* This research was supported by the National Science Foundation under Grant No. DCR75-17090.

Definition. An ε-free nondeterministic generalized sequential machine with accepting states (EFNGSMA) over $\Sigma \times \Delta$ is a 6-tuple $M = \langle K,\Sigma,\Delta,\delta,q_0,F\rangle$, where K, Σ, and Δ are finite nonempty sets called the state set, input alphabet, and output alphabet, respectively. δ is a function from $K \times \Sigma$ into the finite subsets of $K \times \Delta^+$, q_0 in K is the initial state, and $F \subseteq K$ is a set of accepting states. (Δ^+ denotes the set of all non-null finite-length strings of symbols in Δ.)

If $F = K$ (i.e., all states are accepting), M is called simply an EFNGSM. In this case, $F(=K)$ is not included in the specification.

The function δ is extended to $K \times \Sigma^+$ as follows: For q in K, x_1,x_2 in Σ^+, $\delta(q,x_1x_2) = \{(p,y_1y_2) \mid$ for some p', (p',y_1) is in $\delta(q,x_1)$ and (p,y_2) is in $\delta(p',x_2)\}$. For x in Σ^+, let $M(x) = \{y \mid (p,y)$ is in $\delta(q_0,x)$ for some p in $F\}$. Let $R(M) = \{(x,y) \mid x$ in Σ^+, y in $M(x)\}$. A relation $R \subseteq \Sigma^+ \times \Delta^+$ is called an EFNGSMA (respectively, EFNGSM) relation over $\Sigma \times \Delta$ if we can find an EFNGSMA (respectively, EFNGSM) M such that $R(M) = R$.

For convenience, we will sometimes represent an EFNGSMA $M = \langle K,\Sigma,\Delta,\delta,q_0,F\rangle$ by a directed labeled graph where the nodes represent states and the labeled edges represent transitions. If $\delta(q,a)$ contains (p,y), then there is an edge from node q to node p with label a/y. For example, Figure 1 shows an EFNGSMA, where $K = \{q_0,q_1,q_2,q_3\}$, $\Sigma = \{a,b\}$, $\Delta = \{0,1\}$, q_0 is the initial state, and $F = \{q_0,q_2,q_3\}$.

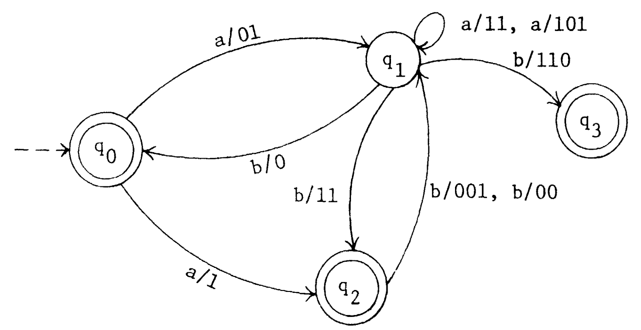


Figure 1. An EFNGSMA

The equivalence problem for EFNGSMA (respectively, EFNGSM) relations over $\Sigma \times \Delta$ is the problem of deciding for arbitrary EFNGSMA's (respectively, EFNGSM's) M_1 and M_2 over $\Sigma \times \Delta$ whether $R(M_1) = R(M_2)$.

2. Unsolvability of the Equivalence Problem for EFNGSM Relations over $\{0,1\} \times \{1\}$

First, we prove the following lemma.

Lemma 1. The following statements are equivalent:

(a) The equivalence problem for EFNGSMA relations over $\Sigma \times \{1\}$ is solvable for any Σ containing at least 2 elements.

(b) The equivalence problem for EFNGSM relations over $\{0,1\} \times \{1\}$ is solvable.

Proof. Clearly, (a) implies (b). To prove the converse, consider 2 EFNGSMA's N_1 and N_2 over $\Sigma = \{a_1, \ldots, a_n\}$, $n \geq 2$ and $\Delta = \{1\}$. By encoding each a_j $(1 \leq j \leq n)$ as string $10^j 1^{n-j} 1$ (of length $n+2$), we can construct 2 new EFNGSMA's M_1 and M_2 such that $R(M_i) = \{(10^{j_1} 1^{n-j_1} 1 10^{j_2} 1^{n-j_2} 1 \ldots 10^{j_k} 1^{n-j_k} 1, 1^{r(n+2)}) \mid (a_{j_1} a_{j_2} \ldots a_{j_k}, 1^r)$ is in $R(N_i)\}$, $i = 1,2$. The construction of M_i from N_i is straightforward. For example, if in N_i there is a transition of the form shown in Figure 2(a), then the "encoded" sequence of transitions in M_i has the form shown in Figure 2(b), where the intermediates states numbered $0,1,\ldots,n$ are new. Clearly, $R(M_1) = R(M_2)$ if and only if $R(N_1) = R(N_2)$. Now let $M_i = \langle K_i, \{0,1\}, \{1\}, \delta_i, q_{0i}, F_i \rangle$, $i = 1,2$. We shall construct 2 EFNGSM's M and M' from M_1 and M_2.

Assume that $K_1 \cap K_2 = \emptyset$, and let $q_0, p_1, \ldots, p_{n+2}$ be new states not in $K_1 \cup K_2$. Let $M = \langle K_1 \cup K_2 \cup \{q_0, p_1, \ldots, p_{n+2}\}, \{0,1\}, \{1\}, \delta, q_0 \rangle$, where δ is defined as follows:

(1) For each a in $\{0,1\}$, let $\delta(q_0, a) = \delta_1(q_{01}, a) \cup \delta_2(q_{02}, a)$.

(2) For each q in $K_1 \cup K_2$ and a in $\{0,1\}$, let $\delta(q,a) = \delta_1(q,a) \cup \delta_2(q,a)$.

(3) For $1 \leq i < n+2$, let $\delta(p_i, 1) = \{(p_{i+1}, 11)\}$.

(4) For each q in F_1, let $\delta(q,1) = \{(p_1, 11)\}$.

(5) For each q in F_2, let $\delta(q,1) = \{(p_{n+2}, 11)\}$.

M' is defined like M except that (4) and (5) are replaced by:

(4') For each q in F_2, let $\delta(q,1) = \{(p_1, 11)\}$.

(5') For each q in F_1, let $\delta(q,1) = \{(p_{n+2}, 11)\}$.

Clearly, $R(M_1) = R(M_2)$ implies $R(M) = R(M')$. Now suppose $R(M) = R(M')$. Consider (x,y) in $R(M_1)$. Then, $(x1^{n+2}, y1^{2(n+2)})$ is in $R(M)$ and, hence, also in $R(M')$. But from the construction of M_1, M_2, M, and M' it is clear that the only way $(x1^{n+2}, y1^{2(n+2)})$ can be in $R(M')$ is for (x,y) to be in $R(M_2)$. Hence $R(M_1) \subseteq R(M_2)$. By symmetry, $R(M_2) \subseteq R(M_1)$. Thus, $R(M) = R(M')$ if and only if $R(M_1) = R(M_2)$, and if and only if $R(N_1) = R(N_2)$. It follows that (b) implies (a). ▨

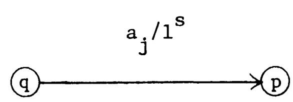
Notation. For any input alphabet Σ, define the one-state EFNGSMA $M_\Sigma = \langle \{q\}, \Sigma, \{1\}, \delta, q, \{q\} \rangle$, where $\delta(q,a) = \{(q, 1^k) \mid k = 1,2,3\}$ for each a in Σ. Clearly, $R(M_\Sigma) = \{(x, 1^r) \mid$ for some x_1, x_2, x_3 in Σ^*, $x = x_1 x_2 x_3 \neq \varepsilon$ and $r = |x_1| + 2|x_2| + 3|x_3|\}$. ($\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$, $|x|$ = length of x.)

Theorem 1. It is recursively unsolvable to determine for arbitrary input alphabet Σ and EFNGSMA M over $\Sigma \times \{1\}$ whether $R(M) = R(M_\Sigma)$.

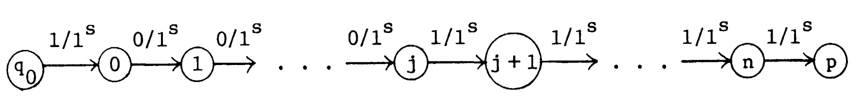
Proof. The proof involves a reduction of the halting problem for single-tape Turing machines to the problem at hand. We show how we can construct for a given single-tape Turing machine Z an EFNGSMA M over $\Sigma \times \{1\}$ (for some Σ) such that $R(M) = R(M_\Sigma)$ if and only if Z does not halt on an initially blank tape. Since the halting problem for Turing machines is unsolvable [5], the result would follow. The construction of M uses some ideas developed in the proof of Theorem 6.3 of [6].

Let Z be a single-tape Turing machine and K be its set of states. Assume without loss of generality that Z's tape alphabet consists of 0, 1 and b (for blank). We may also assume that Z never overwrites a symbol by a blank. Hence, any configuration of Z can be written as $bxqyb$, where x, y are strings of 0's and 1's, and q is in K. The initial configuration is bq_0b, where we assume that q_0, the initial state, is not a halting state. The EFNGSMA M we shall construct has input alphabet $\Sigma = \{0,1,b,\#\} \cup K$, where # is a new symbol.

Let $L_Z = \{x \mid x = \#ID_1\# \ldots \#ID_k\#$, $k \geq 2$, $ID_1, \ldots, ID_k$ are configurations of Z, ID_1 is the initial configuration, and ID_k is a halting configuration$\}$. Clearly, L_Z is a regular set and finite automata [7] N_1 and N_2 can be constructed to accept L_Z and $\Sigma^+ - L_Z$, respectively. The EFNGSMA M is constructed from 4 EFNGSMA's $M_1, \ldots, M_4$ such that $R(M) = R(M_1) \cup \ldots \cup R(M_4)$. Since EFNGSMA relations are obviously effectively closed



(a) Transition in N_i



(b) Equivalent sequence of transitions in M_i

(states $0,1,\ldots,n$ are new)

Figure 2.

under union, we need only describe the construction of $M_1,\ldots,M_4$.

(1) Let $R_1 = \{(x,1^r) \mid (x,1^r) \text{ in } R(M_\Sigma),\ x \text{ in } \Sigma^+ - L_Z\}$. (See notation above.) Clearly, an EFNGSMA M_1 can be constructed from M_Σ and finite automaton N_2 so that $R(M_1) = R_1$.

(2) M_2 and M_3 are shown in Figure 3. It is easy to verify that $R(M_2) = \{(x,1^r) \mid (x,1^r) \text{ in } R(M_\Sigma),\ r > 2|x|\}$ and $R(M_3) = \{(x,1^r) \mid (x,1^r) \text{ in } R(M_\Sigma),\ r < 2|x|\}$.

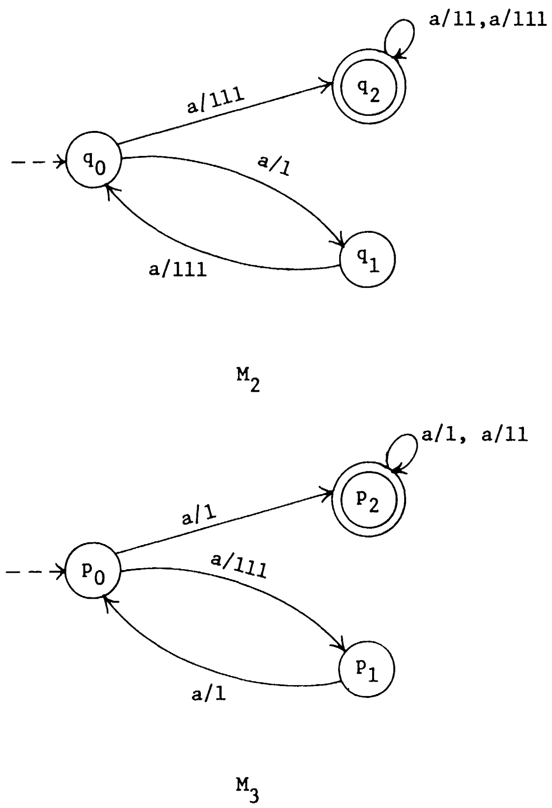


Figure 3. a/x represents several transitions, one for each a in Σ.

(3) Now let $R_4 = \{(x,1^r) \mid (x,1^r) \text{ in } R(M_\Sigma),\ x = \#ID_1\# \ldots \#ID_k\# \text{ in } L_Z$ and either $r \neq 2|x|$ or $r = 2|x|$ and for some ID_i, $1 \leq i < k$, ID_{i+1} is not a proper successor of $ID_i\}$. We shall construct an EFNGSMA M_4 such that $R(M_4) = R_4$. Since the finite automaton N_1 (accepting L_Z) can easily be built into the finite-state control of M_4, we may assume that the inputs to M_4 come from the language L_Z.

M_4 may (nondeterministically) choose to simulate either M_2 or M_3, or perform the following operations on input $x = \#ID_1\# \ldots \#ID_i\#ID_{i+1}\# \ldots \#ID_k\#$ (see Figure 4): M_4 moves right emitting 2 ones/move until it reaches the # immediately to the left of some ID_i, $1 \leq i < k$ (ID_i is chosen nondeterministically.) Then M_4 moves right emitting 1 one/move until it reaches some number $\ell_2 \geq 1$ (chosen nondeterministically) of squares to the right of # and guesses that an "error" occurs in position ℓ_2, ℓ_2+1, or ℓ_2+2 of ID_i and ID_{i+1}. M_4 uses its finite-state control to remember these symbols of ID_i as it moves right (of square ℓ_2) emitting 2 ones/move until it reaches the next #. Then M_4 moves right (of #) emitting 3 ones/move. At some point, M_4 guesses that the number ℓ_4 (≥ 1) of squares it has crossed from the last # is equal to ℓ_2. It then moves right (of square ℓ_4) emitting 2 ones/move and checks whether the symbols at positions ℓ_4, ℓ_4+1, and ℓ_4+2 are appropriate for the successor of ID_i if $\ell_2 = \ell_4$. If they are appropriate (respectively, not appropriate), M_4 enters a nonaccepting state (respectively, accepting state) and remains in this state emitting 2 ones/move as it advances to the right. The formal construction of M_4 from our description of its operation is straightforward but tedious and is therefore omitted.

Now suppose $(x,1^r)$ is in $R(M_4)$. Then for some ℓ_1, ℓ_2, ℓ_3, ℓ_4, ℓ_5, $|x| = \ell_1 + \ell_2 + \ell_3 + \ell_4 + \ell_5$ (see Figure 4) and $r = 2\ell_1 + \ell_2 + 2\ell_3 + 3\ell_4 + 2\ell_5$. Clearly, $r = 2|x|$ if and only if $\ell_2 = \ell_4$. It follows that $R(M_4) = R_4$.

Let M be an EFNGSMA such that $R(M) = R(M_1) \cup \ldots \cup R(M_4)$. Then $R(M) = R(M_\Sigma)$ if and only if the Turing machine Z does not halt. ▨

Corollary 1. There is no algorithm P to construct for a given EFNGSMA $M = \langle K,\Sigma,\{1\},\delta,q_0,F\rangle$ a state-minimal EFNGSMA $M' = \langle K',\Sigma,\{1\},\delta',q_0',F'\rangle$ such that $R(M) = R(M')$.

Proof. Suppose an algorithm P exists. Let Z be a single-tape Turing machine and M be the associated EFNGSMA constructed in the proof of Theorem 1. Using P, construct a state-minimal machine M' equivalent to M, i.e., $R(M) = R(M')$. Now look at the one-state EFNGSMA M_Σ and consider the following cases.

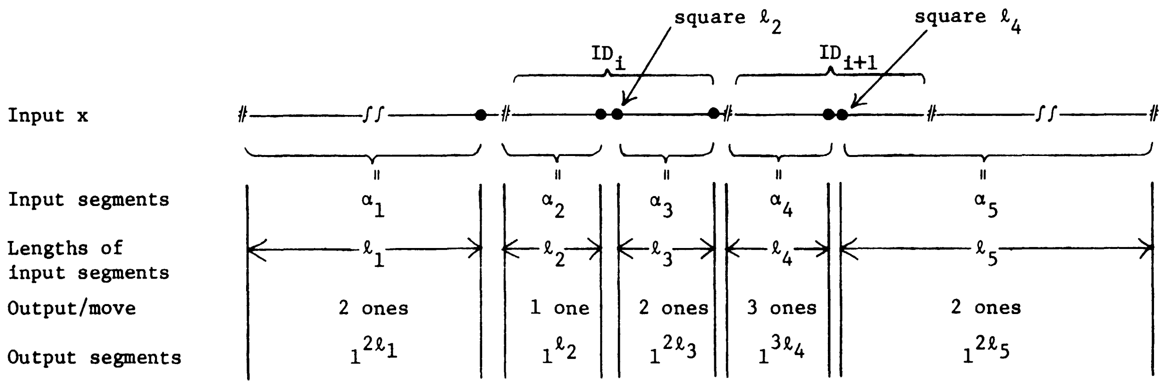
case 1. M' has only 1 state. Then it is obvious that $R(M) = R(M') = R(M_\Sigma)$ if and only if M' and M_Σ are identical which is trivially decidable.

case 2. M' has more than 1 state. Then $R(M) = R(M') \neq R(M_\Sigma)$ since M_Σ has only 1 state.

Cases 1 and 2 show that we can decide if $R(M) = R(M') = R(M_\Sigma)$. But from the proof of Theorem 1, $R(M) = R(M_\Sigma)$ if and only if Z does not halt. The result follows. ▨

From the constructions in Lemma 1 and Theorem 1, we have one of our main results:

Theorem 2. Let $\mathcal{S}_1$ be the class of EFNGSM's $M = \langle K, \{0,1\},\{1\},\delta,q_0\rangle$, where δ satisfies the property that for each q in K and a in $\{0,1\}$, $(p,1^k)$ in $\delta(q,a)$ implies $k = 1,2,3$. Then the equivalence problem for $\mathcal{S}_1$ is unsolvable.



$\alpha_1 = \#ID_1\#\ldots\#ID_{i-1}$, $\alpha_2\alpha_3 = \#ID_i$, $\alpha_4\alpha_5 = \#ID_{i+1}\#\ldots\#ID_k\#$

● represents a symbol

Figure 4.

3. Unsolvability of the Equivalence Problem for EFNGSM Relations over $\{1\}\times\{0,1\}$

Lemma 2. The following statements are equivalent:

(a) The equivalence problem for EFNGSMA relations over $\{1\}\times\Delta$ is solvable for any Δ containing at least 2 elements.

(b) The equivalence problem for EFNGSM relations over $\{1\}\times\{0,1\}$ is solvable.

Proof. The proof is similar to that of Lemma 1. (a) certainly implies (b). For the converse, consider 2 EFNGSMA's N_1 and N_2 over $\{1\}\times\Delta$. Let $\Delta = \{a_1,\ldots,a_n\}$, $n \geq 1$. Let h be a homomorphism defined by: $h(a_j) = 10^j1^{n-j}1$, $1 \leq j \leq n$. We construct EFNGSMA's M_1 and M_2 such that $R(M_i) = \{(1^{r(n+2)},h(x)) \mid (1^r,x) \text{ is in } R(N_i)\}$, $i=1,2$. If in N_i there is a transition of the form shown in Figure 5(a), then the "encoded" sequence of transitions in M_i has the form shown in Figure 5(b). The states numbered $0,1,\ldots,n$ are new.

Then $R(M_1) = R(M_2)$ if and only if $R(N_1) = R(N_2)$. Let $M_i = \langle K_i,\{1\},\{0,1\},\delta_i,q_{0i},F_i\rangle$, $i=1,2$, and assume $K_1\cap K_2 = \emptyset$. Construct an EFNGSM $M = \langle K_1\cup K_2\cup\{q_0,p_1,\ldots,p_{n+2}\},\{1\},\{0,1\},\delta,q_0\rangle$, where $q_0,p_1,\ldots,p_{n+2}$ are new states and δ is defined as follows:

(1) $\delta(q_0,1) = \delta_1(q_{01},1)\cup\delta_2(q_{02},1)$.

(2) For each q in $K_1\cup K_2$, let $\delta(q,1) = \delta_1(q,1)\cup\delta_2(q,1)$.

(3) For $1 \leq i < n+2$, let $\delta(p_i,1) = \{(p_{i+1},11)\}$.

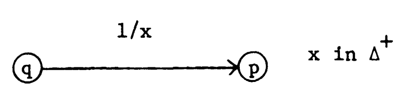
(4) For each q in F_1, let $\delta(q,1) = \{(p_1,11)\}$.

(5) For each q in F_2, let $\delta(q,1) = \{(p_{n+2},11)\}$.

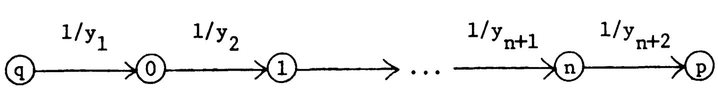
Construct another EFNGSM M' which is just like M except that (4) and (5) are replaced by:

(4') For each q in F_2, let $\delta(q,1) = \{(p_1,11)\}$.

(5') For each q in F_1, let $\delta(q,1) = \{(p_{n+2},11)\}$.



(a) Transition in N_i



$h(x) = y_1y_2 \cdots y_{n+2}$, $|x| = |y_1| = \ldots = |y_{n+2}|$

(b) Equivalent sequence of transitions in M_i (states $0,1,\ldots,n$ are new)

Figure 5.

Then $R(M) = R(M')$ if and only if $R(M_1) = R(M_2)$. Hence, (b) implies (a). ▨

Notation. For any output alphabet Δ, define the one-state EFNGSMA $M_\Delta = \langle\{q\},\{1\},\Delta,\delta,q,\{q\}\rangle$, where $\delta(q,1) = \{(q,y) \mid y \text{ in } \Delta^+,\ |y| = 2,3, \text{ or } 6\}$. Clearly, $R(M_\Delta) = \{(1^r,y) \mid \text{for some integers } r_1,r_2,r_3 \geq 0,\ r = r_1 + r_2 + r_3 \neq 0,\ y \text{ in } \Delta^+ \text{ and } |y| = 2r_1 + 3r_2 + 6r_3\}$.

Theorem 3. It is recursively unsolvable to determine for arbitrary output alphabet Δ and EFNGSMA M over $\{1\} \times \Delta$ whether $R(M) = R(M_\Delta)$.

Proof. The proof is similar to that of Theorem 1. Let Z be a single-tape Turing machine with state set K and tape alphabet consisting of 0, 1, and b. As before, we assume that the initial state q_0 is not a halting state and Z does not overwrite a symbol by a blank. Let $\Delta = \{0,1,b,\#\} \cup K$. We shall construct an EFNGSMA M over $\{1\} \times \Delta$ such that $R(M) = R(M_\Delta)$ if and only if Z does not halt on an initially blank tape.

Let h be a homomorphism on Δ^* defined by $h(a) = aaaaaa$ for each a in Δ. Let $Q_Z = \{h(x) \mid x = \#ID_1\# \ldots \#ID_k\#,\ k \geq 2,\ ID_1,\ldots,ID_k$ are configurations of Z, ID_1 is the initial configuration, and ID_k is a halting configuration$\}$. (Thus, h(x) is just like x except that each symbol is written 6 times.) We can construct finite automata N_1 and N_2 to accept Q_Z and $\Delta^+ - Q_Z$, respectively.

Now define 4 EFNGSMA's $M_1,\ldots,M_4$ over $\{1\} \times \Delta$ as follows:

(1) M_1 is such that $R(M_1) = \{(1^r,y) \mid (1^r,y) \text{ in } R(M_\Delta),\ y \text{ in } \Delta^+ - Q_Z\}$. Clearly, M_1 can be constructed from M_Δ and finite automaton N_2.

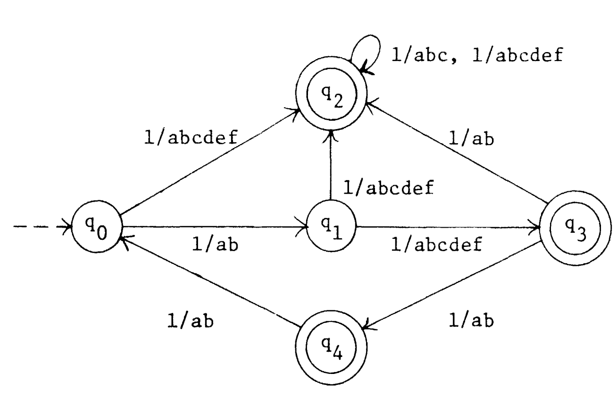
(2) M_2 and M_3 are shown in Figure 6. It is easy to check that $R(M_2) = \{(1^r,y) \mid (1^r,y) \text{ in } R(M_\Delta),\ |y| > 3r\}$ and $R(M_3) = \{(1^r,y) \mid (1^r,y) \text{ in } R(M_\Delta),\ |y| < 3r\}$.

(3) Let $R_4 = \{(1^r,y) \mid (1^r,y) \text{ in } R(M_\Delta),\ y = h(\#ID_1\# \ldots \#ID_k\#) \text{ in } Q_Z$ and either $|y| \neq 3r$ or $|y| = 3r$ and for some ID_i, $1 \leq i < k$, ID_{i+1} is not a proper successor of $ID_i\}$. We shall construct an EFNGSMA M_4 such that $R(M_4) = R_4$.

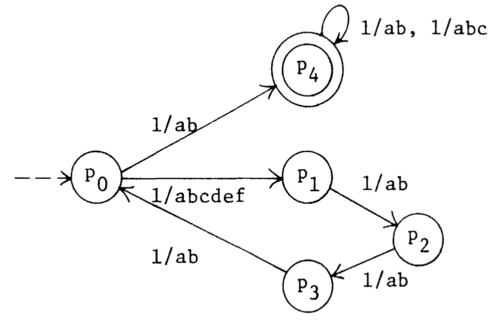
Since Q_Z is a regular set, we may assume that in any successful computation of M_4, the output string generated is in Q_Z. M_4 may (nondeterministically) choose to simulate either M_2 or M_3, or perform the following operations (see Figure 7).

Given input 1^r, M_4 nondeterministically decomposes it into 5 segments, $1^r = 1^{2\ell_1}\,1^{\ell_2}\,1^{2\ell_3}\,1^{3\ell_4}\,1^{2\ell_5}$, and generates the different output segments as follows: M_4 generates $h(\alpha_1)$ at the rate of 3 symbols/move while reading the first $2\ell_1$ ones. Then M_4 scans the next ℓ_2 ones and generates $h(\alpha_2)$ at the rate of 6 symbols/move. The next output segment $h(\alpha_3)$ is generated at the rate of 3 symbols/move while $h(\alpha_4)$ is generated at the rate of 2 symbols/move. Finally, for the last $2\ell_5$ ones, M_4 generates $h(\alpha_5)$ at the rate of 3 symbols/move. As in the proof of Theorem 1, M_4 has to guess that an "error" occurs after generating $h(\alpha_2)$ and checks this condition after generating $h(\alpha_4)$.

Now $r = 2\ell_1 + \ell_2 + 2\ell_3 + 3\ell_4 + 2\ell_5$ and $|y| = 6(\ell_1 + \ell_2 + \ell_3 + \ell_4 + \ell_5)$. Clearly, $|y| = 3r$ if and only if $6\ell_2 = 6\ell_4$. Hence, M_4 can be constructed so that $R(M_4) = R_4$. Construct an EFNGSMA M such that $R(M) = R(M_1) \cup \ldots \cup R(M_4)$. Then $R(M) = R(M_\Delta)$ if and only if Z does not halt, completing the proof. ▨



M_2



M_3

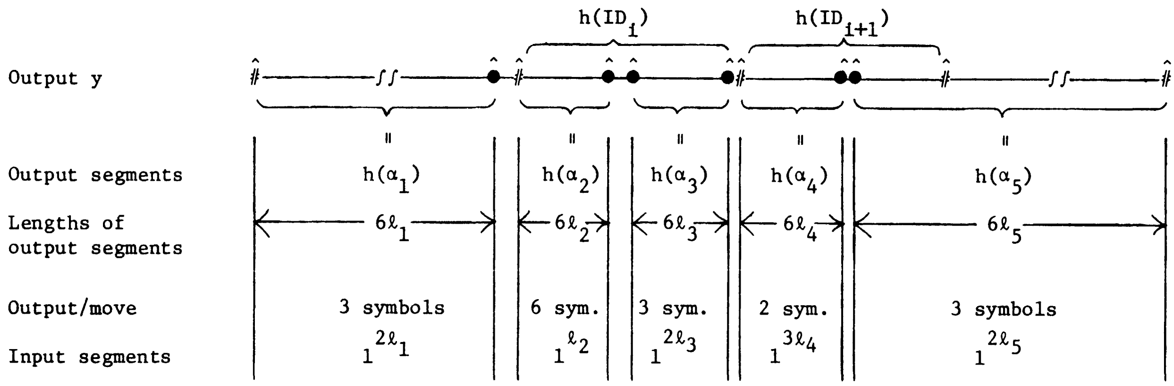
Figure 6. a, b, c, d, e, f represent symbols in Δ. 1/ab, e.g., represents several transitions, one for each choice of a and b in Δ.

Corollary 2. There is no algorithm P to construct for a given EFNGSMA $M = \langle K,\{1\},\Delta,\delta,q_0,F\rangle$ a state-minimal EFNGSMA $M' = \langle K',\{1\},\Delta,\delta',q_0',F'\rangle$ such that $R(M) = R(M')$.

Proof. Similar to that of Corollary 1, this time using Theorem 3. ▨

Theorem 4. Let $\mathcal{B}_2$ be the class of EFNGSM's $M = \langle K,\{1\},\{0,1\},\delta,q_0\rangle$, where δ satisfies the property that for each q in K, (p,y) in $\delta(q,1)$ implies $|y| = 2$, 3, or 6. Then the equivalence problem for $\mathcal{B}_2$ is undecidable.

Proof. A careful study of the constructions in Lemma 2 and Theorem 3. ▨



$h(\alpha_1) = h(\#ID_1\#\ldots\#ID_{i-1})$, $h(\alpha_2)h(\alpha_3) = h(\#ID_i)$

$h(\alpha_4)h(\alpha_5) = h(\#ID_{i+1}\#\ldots\#ID_k\#)$

$\hat{\bullet}$ represents a symbol written 6 times

Figure 7.

Remark. We have shown that the equivalence problem for EFNGSM's with unary input (respectively, output) alphabet is unsolvable. If we require both the input and output to have unary alphabets then the equivalence problem becomes solvable. In fact, it is decidable to determine for arbitrary (not necessarily ε-free) NGSMA's M_1 and M_2 satisfying $R(M_i) \subseteq w_1^* \ldots w_k^* \times z_1^* \ldots z_m^*$ for some k, $m \geq 1$, non-null strings $w_1,\ldots,w_k$, $z_1,\ldots,z_m$, whether $R(M_1) = R(M_2)$. This follows from the decidability of the equivalence problem for bounded context-free languages [3] and the observation that we can effectively construct linear grammars generating languages $L_i = \{x\#y^R \mid y \text{ in } M_i(x)\} \cup \{\#\}$, $i = 1,2$, where $\#$ is a new symbol. (y^R is the reverse of string y.)

4. Applications to Grammar and Graph Problems

Let $M = \langle K,\Sigma,\Delta,\delta,q_0\rangle$ be an EFNGSM. Assume that $K \cap (\Sigma \cup \Delta) = \emptyset$, and let c be a new symbol. We can construct a linear grammar [3] $G = \langle N,\Sigma \cup \Delta \cup \{c\},P,S\rangle$ where $N = K$ is the set of nonterminals, $\Sigma \cup \Delta \cup \{c\}$ is the set of terminals, $S = q_0$, and P is the set of rewriting rules defined as follows: For each q in K and a in Σ, let $q \to apy^R$ be in P if $\delta(q,a)$ contains (p,y). Also let $q \to c$ be in P for each q in K. Clearly, $L(G) \subseteq \Sigma^* c \Delta^*$ and $L(G)$ has the following property which characterizes c-finite languages [3]: (a) for every x in Σ^*, the set $\{y \mid xcy \text{ in } L(G)\}$ is finite, and (b) for every y in Δ^*, the set $\{x \mid xcy \text{ in } L(G)\}$ is finite. G is called a c-finite grammar.

From Theorems 2 and 4, we have the following refinement of Griffiths's result concerning c-finite languages.

Theorem 5. The equivalence problem for c-finite grammars is unsolvable even if the rules are restricted to be of the form $A \to c$ or $A \to aB1^k$, where A and B are nonterminals, a is in $\{0,1\}$, and $k = 1,2,3$. The result also holds for the case when the rules are restricted to be of the form $A \to c$ or $A \to 1Bx$, where x is in $\{0,1\}^+$, and $|x| = 2$, 3, or 6.

The unsolvability results can be applied to two problems concerning right-linear grammars (RLG's). A RLG over Σ is a linear grammar $G = \langle N,\Sigma,P,S\rangle$ where the rules are of the form $A \to xB$ or $A \to x$, A,B in N and x in Σ^+ [5]. (We shall only consider ε-free languages.) If x is in Σ (i.e., x is a single symbol) then G is normalized. It is obvious that for every RLG G_1 we can construct a normalized RLG G_2 equivalent to G_1, i.e., $L(G_1) = L(G_2)$.

Let y be in Σ^+ and $n \geq 1$. If $S \overset{*}{\underset{G}{\Rightarrow}} y$ in an n-step derivation, then write $S \overset{n}{\underset{G}{\Rightarrow}} y$. Call 2 RLG's $G_i = \langle N_i,\Sigma,P_i,S_i\rangle$, $i = 1,2$, time-equivalent if for every y in Σ^+ and $n \geq 1$, $S_1 \overset{n}{\underset{G_1}{\Rightarrow}} y$ if and only if $S_2 \overset{n}{\underset{G_2}{\Rightarrow}} y$. Time-equivalence implies equivalence, but the converse is not true in general. Clearly, time-equivalence of normalized RLG's is decidable. This is not true for arbitrary RLG's:

Theorem 6. The time-equivalence problem for RLG's over $\Sigma = \{0,1\}$ is undecidable. Moreover, the unsolvability holds even if the rules of the form $A \to xB$ or $A \to x$ are such that $|x| = 2$, 3, or 6.

Proof. Let $M = \langle K,\{1\},\{0,1\},\delta,q_0\rangle$ be an EFNGSM, and assume that $K \cap \{0,1\} = \emptyset$. Construct a RLG $G = \langle K,\{0,1\},P,q_0\rangle$, where $P = \{q \to x,\ q \to xp \mid q \text{ in } K,\ \delta(q,1) \text{ contains } (p,x)\}$. Then $(1^n,y)$ is in $R(M)$ if and only if $q_0 \overset{n}{\underset{G}{\Rightarrow}} y$. The result now follows from Theorem 4. ▨

Consider now a normalized RLG $G = \langle N,\Sigma,P,S\rangle$ with labeled rules. Let Γ be the set of labels of rules in P and f be a function (called cost function) from Γ into C, where C is a finite nonempty set of positive integers. Extend f to Γ^+ by defining $f(\alpha_1\alpha_2) = f(\alpha_1) + f(\alpha_2)$ for α_1 and α_2 in Γ^+. Let y be in Σ^+ and $n \geq 1$. If $S \overset{*}{\underset{G}{\Rightarrow}} y$ using a sequence of rules with labels $\ell_1,\ldots,\ell_k$, write $S \overset{n}{\underset{G}{\Rightarrow}} y$ if $n = f(\ell_1 \ldots \ell_k)$. Call 2 normalized RLG's $G_i = \langle N_i,\Sigma,P_i,S_i\rangle$ $i = 1,2$, with cost functions f_1 and f_2 cost-equivalent if for every y in Σ^+ and

$n \geq 1$, $S_1 \overset{n}{\underset{G_1}{\Rightarrow}} y$ if and only if $S_2 \overset{n}{\underset{G_2}{\Rightarrow}} y$. It is easy to show that cost-equivalence is decidable for the class of normalized RLG's where the range of the cost functions is a singleton. However, we have

Theorem 7. The cost-equivalence problem for normalized RLG's over $\Sigma = \{0,1\}$ with cost function range $C = \{1,2,3\}$ is undecidable.
Proof. Let $M = \langle K,\{0,1\},\{1\},\delta,q_0\rangle$ be an EFNGSM and assume that $K \cap \{0,1\} = \emptyset$. Construct a normalized RLG $G = \langle K,\{0,1\},P,q_0\rangle$ where P is defined as follows: Let q be in K and a in $\{0,1\}$. If $\delta(q,a)$ contains $(p,1^k)$, then let $q \rightarrow ap$ and $q \rightarrow a$ be in P, and assign to them labels $[q,a,k,p]$ and $[q,a,k,\$]$, respectively. ($ is a new symbol.) Define the cost function f by: $f([q,a,k,p]) = f([q,a,k,\$]) = k$. Then $(y,1^n)$ is in R(M) if and only if $q_0 \overset{n}{\underset{G}{\Rightarrow}} y$. From Theorem 2, the result follows. ▨

Theorems 2 and 4 can also be applied to problems involving graphs. We shall show, for example, how we can use Theorem 2 to prove the unsolvability of some form of equivalence problem concerning directed graphs. First, we state the following lemma.

Lemma 3. Let $M = \langle K,\Sigma,\Delta,\delta,q_0\rangle$ be an EFNGSM. We can effectively construct an EFNGSM $M' = \langle K',\Sigma,\Delta,\delta',q_0'\rangle$ such that $R(M') = R(M)$ and M' has the following properties:

(1) For each q in K' and a in Σ, if (p,x) is in $\delta'(q,a)$ then $q \neq p$. (M' has no reflexive loops.)

(2) For q, p in K' and a_1, a_2 in Σ, if $\delta'(q,a_1)$ contains (p,x_1) and $\delta'(q,a_2)$ contains (p,x_2), then $a_1 = a_2$ and $x_1 = x_2$. (At most 1 transition exists from state q to state p.)

Proof. M' is constructed from M in 2 stages. We describe the construction briefly. First, we remove all reflexive loops by iterating the following process: A reflexive loop in M of the form shown in Figure 8(a) is replaced by equivalent transitions shown in Figure 8(b). Call the resulting EFNGSM M_1. Next, we construct from M_1 the desired EFNGSM M' by iterating the following transformation: Transitions in M_1 of the form shown in Figure 9(a) are replaced by equivalent transitions shown in Figure 9(b). Clearly, $R(M') = R(M)$, and M' satisfies properties (1) and (2). ▨

We can now prove the following theorem. (Refer to the Introduction for notation.)

Theorem 8. It is recursively unsolvable to determine for arbitrary directed graphs $G_i = \langle V_i,E_i,v_{0i},f_i,g_i\rangle$, $i = 1,2$, whether $R(G_1) = R(G_2)$.
Proof. By Theorem 2, it is sufficient to show how we can construct for a given EFNGSM $M = \langle K,\{0,1\},\{1\},\delta, q_0\rangle$ a graph $G = \langle V,E,v_0,f,g\rangle$ such that $R(G) = \{(x,r) \mid (x,1^r)$ is in $R(M)\}$. We may assume that M satisfies properties (1) and (2) of Lemma 3. Let $V = K$, $v_0 = q_0$ and define E, f, and g as follows: For q and p in K, a in $\{0,1\}$ and x in $\{1\}^+$, if $\delta(q,a)$ contains (p,x), then let $\langle q,p\rangle$ be in E, $f(\langle q,p\rangle) = a$, and $g(\langle q,p\rangle) = |x|$. f and g are well-defined since M satisfies properties (1) and (2) of Lemma 3. Clearly, $R(G) = \{(x,r) \mid (x,1^r)$ is in $R(M)\}$, and the result follows. ▨

Finally, suppose $M = \langle K,\Sigma,\Delta,\delta,q_0,F\rangle$ is an EFNGSMA and λ is a function from $\Sigma \cup \Delta$ into a finite nonempty set, C, of positive integers. Define $\lambda(\varepsilon) = 1$ and $\lambda(a_1 \ldots a_n) = \lambda(a_1) * \ldots * \lambda(a_n)$ for $n \geq 1$, $a_1,\ldots,a_n$ in $\Sigma \cup \Delta$. Let $\lambda(R(M)) = \{(\lambda(x),\lambda(y)) \mid (x,y)$ in $R(M)\}$. We will show that equivalence of $\lambda(R(M))$'s is decidable.

Lemma 4. Let $M = \langle K,\Sigma,\Delta,\delta,q_0,F\rangle$ be an EFNGSMA and λ be a function from $\Sigma \cup \Delta$ into C, where C is a finite nonempty set of positive integers. Let $p_1,\ldots,p_k$ be the prime numbers appearing in the prime decompositions of integers in C. Let $a_1,\ldots,a_k$, $b_1,\ldots,b_k$, # be distinct symbols and $L = \{a_1^{d_1} \ldots a_k^{d_k} \# b_1^{e_1} \ldots b_k^{e_k} \mid (p_1^{d_1} \ldots p_k^{d_k}, p_1^{e_1} \ldots p_k^{e_k})$ in $\lambda(R(M))\}$. We can effectively construct a one-way finite automaton, N, with 2k counters accepting L. Moreover, N has the property that in any accepting computation, each counter makes at most 1 reversal.
Proof. N guesses the input x to M and stores in the first k counters the exponents of the prime numbers $p_1,\ldots,p_k$ appearing in the prime decomposition of $\lambda(x)$. Clearly, the exponents can be computed without the counters making any reversal. The exponents corresponding to output $\lambda(y)$ are stored in the second set of k counters. If (x,y) is in R(M), then N goes through its input and checks that the input is of the form $a_1^{d_1} \ldots a_k^{d_k} \# b_1^{e_1} \ldots b_k^{e_k}$, where $d_1,\ldots,d_k$, $e_1,\ldots,e_k$ are the exponents stored in the 2k counters. The checking requires the counters to make a reversal. ▨

In [6] it was shown that the equivalence problem is solvable for one-way finite automata with bounded-reversal counters accepting only bounded languages (i.e., subsets of $w_1^* \ldots w_k^*$ for some strings $w_1,\ldots, w_k$). Thus, we have

Theorem 9. It is decidable to determine for arbitrary EFNGSMA's $M_i = \langle K_i,\Sigma_i,\Delta_i,\delta_i,q_{0i},F_i\rangle$ and functions λ_i from $\Sigma_i \cup \Delta_i$ into C_i, $i = 1,2$, whether $\lambda_1(R(M_1)) = \lambda_2(R(M_2))$.

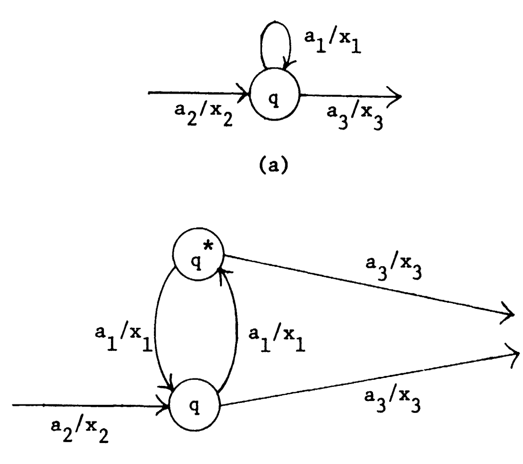
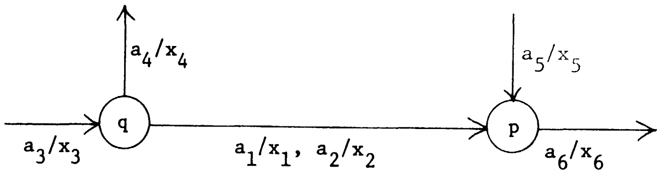
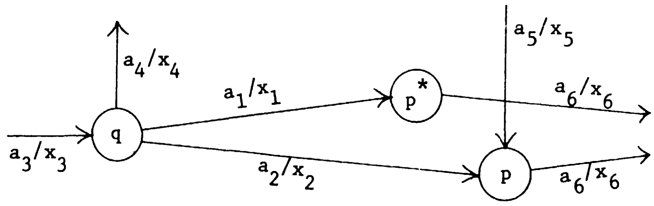


Figure 8. q^* is a new state.



(a)



(b)

Figure 9. p^* is a new state.

References

[1] Bird, M., The equivalence problem for deterministic two-tape automata, Journal of Computer and System Sciences, 7 (1973), 218-236.

[2] Blattner, M., The unsolvability of the equality problem for sentential forms of context-free grammars, Journal of Computer and System Sciences, 7 (1973), 463-468.

[3] Ginsburg, S., "The Mathematical Theory of Context-free languages," McGraw-Hill, New York, 1966.

[4] Griffiths, T. V., The unsolvability of the equivalence problem for ε-free nondeterministic generalized machines, Journal of the Association for Computing Machinery, 15(1968), 409-413.

[5] Hopcroft, J. E. and Ullman, J. D., "Formal Languages and their Relation to Automata," Addison Wesley, Reading, Massachusetts, 1969.

[6] Ibarra, O. H., Reversal-bounded multi-counter machines and their decision problems, to appear in the Journal of the Association for Computing Machinery.

[7] Rabin, M. and Scott, D., Finite automata and their decision problems, IBM Journal of Research and Development, 3(1959), 114-125.