KYBERNETIKA — VOLUME 41 (2005), NUMBER 4, PAGES 531–537

SELF–REPRODUCING PUSHDOWN TRANSDUCERS

Alexander Meduna and Luboš Lorenc

After a translation of an input string, x, to an output string, y, a self-reproducing pushdown transducer can make a self-reproducing step during which it moves y to its input tape and translates it again. In this self-reproducing way, it can repeat the translation n-times for any $n \geq 1$. This paper demonstrates that every recursively enumerable language can be characterized by the domain of the translation obtained from a self-reproducing pushdown transducer that repeats its translation no more than three times.

Keywords: pushdown transducer, self-reproducing pushdown transduction, recursively enumerable languages

AMS Subject Classification: 68Q45

1. INTRODUCTION

In this paper, we introduce and discuss a self-reproducing pushdown transducer, which represents a natural modified version of an ordinary pushdown transducer. After a translation of an input string, x, to an output string, y, a self-reproducing pushdown transducer can make a self-reproducing step during which it moves y to its input tape and translates it again. In this self-reproducing way, it can repeat the translation n-times, for $n \geq 1$. This paper demonstrates that every recursively enumerable language can be characterized by the domain of the translation obtained from a self-reproducing pushdown transducer that repeats its translation no more than three times.

This characterization is of some interest because it does not hold in terms of ordinary pushdown transducers. Indeed, the domain obtained from any ordinary pushdown transducer is a context-free language (see [1]).

2. PRELIMINARIES

This paper assumes that the reader is familiar with the theory of automata and formal languages (see [3, 6]).

For a set Q, $Card(Q)$ denotes the cardinality of Q. For an alphabet V, V^* represents the free monoid generated by V under the operation of concatenation. The identity of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ denotes the length of w. For every $i \in \{0, 1, \ldots, |w|\}$, *Suffix*$(w, i)$ denotes w's suffix of length i; analogously, *Prefix*(w, i) denotes w's prefix of length i.

A *queue grammar* (see [2]) is a six-tuple, $Q = (V, T, W, F, s, P)$, where V and W are alphabets satisfying $V \cap W = \varnothing$, $T \subseteq V$, $F \subseteq W$, $s \in (V - T)(W - F)$, and $P \subseteq (V \times (W - F)) \times (V^* \times W)$ is a finite relation such that for every $a \in V$ there exists an element $(a, b, x, c) \in P$. If $u, v \in V^*W$ such that $u = arb$; $v = rxc$; $a \in V$; $r, x \in V^*$; $b, c \in W$; and $(a, b, x, c) \in P$, then $u \Rightarrow v$ $[(a, b, x, c)]$ in G or, simply $u \Rightarrow v$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then based on $\Rightarrow^n$ define $\Rightarrow^+$ and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{w \in T^* : s \Rightarrow^* wf$ where $f \in F\}$.

A *left-extended queue grammar* (see [5]) is similar to an ordinary queue grammar except that it records the members of V used when it works. Formally, a *left-extended queue grammar* is a six-tuple, $Q = (V, T, W, F, s, P)$ where V, T, W, F and s have the same meaning as in a queue grammar. $P \subseteq (V \times (W - F)) \times (V^* \times W)$ is a finite relation (as opposed to an ordinary queue grammar, this definition does not require that for every $a \in V$, there exists an element $(a, b, x, c) \in P$). Furthermore, assume that $\# \notin V \cup W$. If $u, v \in V^*\{\#\}V^*W$ so that $u = w\#arb$; $v = wa\#rxc$; $a \in V$; $r, x, w \in V^*$; $b, c \in W$; and $(a, b, x, c) \in P$, then $u \Rightarrow v$ $[(a, b, x, c)]$ in G or, simply $u \Rightarrow v$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{v \in T^* : \#s \Rightarrow^* w\#vf$ for some $w \in V^*$ and $f \in F\}$.

3. DEFINITIONS

A *self-reproducing pushdown transducer* is a 8-tuple $M = (Q, \Gamma, \Sigma, \Omega, R, s, S, O)$, where Q is a finite set of states, Γ is a total alphabet such that $Q \cap \Gamma = \varnothing$, $\Sigma \subseteq \Gamma$ is an input alphabet, $\Omega \subseteq \Gamma$ is an output alphabet, R is a finite set of *translation rules* of the form $u_1qw \rightarrow u_2pv$ with $u_1, u_2, w, v \in \Gamma^*$ and $q, p \in Q$, $s \in Q$ is the *start state*, $S \in \Gamma$ is the *start pushdown symbol*, $O \subseteq Q$ is the set of *self-reproducing states*. A *configuration of* M is any string of the form $\$zqy\$x$, where $x, y, z \in \Gamma^*$, $q \in Q$, and \$ is a special *bounding symbol* ($\$ \notin Q \cup \Gamma$). If $u_1qw \rightarrow u_2pv \in R$, $y = \$hu_1qwz\$t$, and $x = \$hu_2pz\$tv$, where $h, u_1, u_2, w, t, v, z \in \Gamma^*$, $q, p \in Q$, then M makes a *translation step* from y to x in M, symbolically written as $y \,{}_t\!\Rightarrow x$ $[u_1qw \rightarrow u_2pv]$ or, simply $y \,{}_t\!\Rightarrow x$ in M. If $y = \$hq\$t$, and $x = \$hqt\$$, where $t, h \in \Gamma^*$, $q \in O$, then M makes a *self-reproducing step* from y to x in M, symbolically written as $y \,{}_r\!\Rightarrow x$. Write $y \Rightarrow x$ if $y \,{}_t\!\Rightarrow x$ or $y \,{}_r\!\Rightarrow x$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. Let $w, v \in \Gamma^*$; M *translates* w to v if $\$Ssw\$ \Rightarrow^* \$q\$v$ in M. *The translation obtained from* M, $T(M)$, is defined as $T(M) = \{(w, v) : \$Ssw\$ \Rightarrow^* \$q\$v$ with $w \in \Sigma^*$, $v \in \Omega^*$, $q \in Q\}$. Set *Domain*$(T(M)) = \{w : (w, x) \in T(M)\}$ and *Range*$(T(M)) = \{x : (w, x) \in T(M)\}$. Let n be a nonnegative integer; if during every translation M makes no more than n self-reproducing steps, then M is an *n-self-reproducing pushdown transducer.* Two self-reproducing transducers are equivalent if they both define the same translation.

In the literature, there often exists a requirement that a pushdown transducer,

$M = (Q, \Gamma, \Sigma, \Omega, R, s, S, O)$, replaces no more than one symbol on its pushdown and reads no more than one symbol during every move. As stated next, we can always turn any self-reproducing pushdown transducer to an equivalent self-reproducing pushdown transducer that satisfies this requirement.

Theorem 1. Let M be a self-reproducing pushdown transducer. Then, there is an equivalent self-reproducing pushdown transducer, $N = (Q, \Gamma, \Sigma, \Omega, R, s, S, O)$, in which every translation rule, $u_1qw \to u_2pv \in R$, where $u_1, u_2, w, v \in \Gamma^*$ and $q, p \in Q$, satisfies $|u_1| \leq 1$ and $|w| \leq 1$.

Proof. (Sketch) Consider every rule $u_1qw \to u_2pv$ in M with $|u_1| \geq 2$ or $|w| \geq 2$. N simulates a move made according to this rule as follows. First, N leaves q for a new state and makes $|w|$ consecutive moves during which it reads w symbol by symbol so that after these moves, it has w recorded in a new state, $\langle qw \rangle$. From this new state, it makes $|u_1|$ consecutive moves during which it pops u_1 symbol by symbol from the pushdown so that after these moves, it has both u_1 and w recorded in another new state, $\langle u_1qw \rangle$. To complete this simulation, it performs a move according to $\langle u_1qw \rangle \to u_2pv$. Otherwise, N works as M. A detailed version of this proof is left to the reader. □

4. RESULTS

Lemma 1. For every recursively enumerable language, L, there exists a left-extended queue grammar, Q, satisfying $L(Q) = L$.

Proof. Recall that every recursively enumerable language is generated by queue grammar (see [2]). Clearly, for every queue grammar, there exists an equivalent left-extended queue grammar. Thus, this lemma holds. □

Lemma 2. Let Q' be an left-extended queue grammar. Then there exists a left-extended queue grammar, $Q = (V, T, W, F, s, R)$, such that $L(Q') = L(Q), W = X \cup Y \cup \{1\}$, where $X, Y, \{1\}$ are pairwise disjoint, and every $(a, b, x, c) \in R$ satisfies either $a \in V - T,\ b \in X,\ x \in (V-T)^*,\ c \in X \cup \{1\}$ or $a \in V - T,\ b \in Y \cup \{1\},\ x \in T^*,\ c \in Y$. Q generates every $h \in L(Q)$ in this way

$$
\begin{array}{ll}
\#a_0q_0 & \\
\Rightarrow a_0\#x_0q_1 & [(a_0, q_0, z_0, q_1)] \\
\Rightarrow a_0a_1\#x_1q_2 & [(a_1, q_1, z_1, q_2)] \\
\vdots & \\
\Rightarrow a_0a_1 \ldots a_k\#x_kq_{k+1} & [(a_k, q_k, z_k, q_{k+1})] \\
\Rightarrow a_0a_1 \ldots a_ka_{k+1}\#x_{k+1}y_1q_{k+2} & [(a_{k+1}, q_{k+1}, y_1, q_{k+2})] \\
\vdots & \\
\Rightarrow a_0a_1 \ldots a_ka_{k+1} \ldots a_{k+m-1}\#x_{k+m-1}y_1 \ldots y_{m-1}q_{k+m} & \\
 & [(a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})] \\
\Rightarrow a_0a_1 \ldots a_ka_{k+1} \ldots a_{k+m}\#y_1 \ldots y_mq_{k+m+1} & [(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})]
\end{array}
$$

where $k, m \geq 1$, $a_i \in V-T$ for $i=0,\ldots,k+m$, $x_j \in (V-T)^*$ for $j=1,\ldots,k+m-1$, $s = a_0q_0$, $a_jx_j = x_{j-1}z_j$ for $j = 1,\ldots,k$, $a_1\ldots a_kx_k = z_0\ldots z_k$, $a_{k+1}\ldots a_{k+m} = x_k$, $q_0, q_1, \ldots q_{k+m} \in W - F$ and $q_{k+m+1} \in F$, $z_1,\ldots,z_k \in (V-T)^*$, $y_1,\ldots,y_m \in T^*$, $h = y_1y_2\ldots y_{m-1}y_m$, $q_{k+1} = 1$.

Proof. See Lemma 1 in [4]. □

Lemma 3. Let Q be a left-extended queue grammar satisfying the properties given in Lemma 2. Then, there exists a 2-self-reproducing pushdown transducer, M, such that $Domain(T(M)) = L(Q)$ and $Range(T(M)) = \{\varepsilon\}$.

Proof. Let $G = (V, T, W, F, s, P)$ be a left-extended queue grammar satisfying the properties given in Lemma 2. Without any loss of generality, assume that $\{0,1\} \cap (V \cup W) = \varnothing$. For some positive integer, n, define an injection, ι, from P to $(\{0,1\}^n - \{1\}^n)$ so that ι is an injective homomorphism when its domain is extended to $(VW)^*$; after this extension, ι thus represents an injective homomorphism from $(VW)^*$ to $(\{0,1\}^n - \{1\}^n)^*$; a proof that such an injection necessarily exists is simple and left to the reader. Based on ι, define the substitution, ν, from V to $(\{0,1\}^n - \{1\}^n)$ so that for every $a \in V$, $\nu(a) = \{\iota(p) : p \in P,\ p = (a,b,x,c)$ for some $x \in V^*;\ b,c \in W\}$. Extend the domain of ν to V^*. Furthermore, define the substitution, μ, from W to $(\{0,1\}^n - \{1\}^n)$ so that for every $q \in W$, $\mu(q) = \{\iota(p) : p \in P,\ p = (a,b,x,c)$ for some $a \in V,\ x \in V^*;\ b,c \in W\}$. Extend the domain of μ to W^*. □

Construction 1. Construction of M. Introduce the self-reproducing pushdown transducer

$$M = (Q, T \cup \{0,1,S\}, T, \varnothing, R, z, S, O)$$

where $Q = \{o,f,z\} \cup \{\langle p,i\rangle : p \in W \text{ and } i \in \{1,2\}\}$, $O = \{o,f\}$, and R is constructed by performing the following steps 1 through 6.

1. if $a_0q_0 = s$, where $a \in V - T$ and $q \in W - F$, then add $Sz \to uS\langle q_0,1\rangle w$ to R, for all $w \in \mu(q_0)$ and all $u \in \nu(a_0)$;
2. if $(a,q,y,p) \in P$, where $a \in V-T$, $p,q \in W-F$, and $y \in (V-T)^*$, then add $S\langle q,1\rangle \to uS\langle p,1\rangle w$ to R, for all $w \in \mu(p)$ and $u \in \nu(y)$;
3. for every $q \in W - F$, add $S\langle q,1\rangle \to S\langle q,2\rangle$ to R;
4. if $(a,q,y,p) \in P$, where $a \in V-T$, $p,q \in W-F$, and $y \in T^*$, then add $S\langle q,2\rangle y \to S\langle p,2\rangle w$ to R, for all $w \in \mu(p)$;
5. if $(a,q,y,p) \in P$, where $a \in V-T$, $q \in W-F$, $y \in T^*$, and $p \in F$, then add $S\langle q,2\rangle y \to SoS$ to R;
6. add $o0 \to 0o$, $o1 \to 1o$, $oS \to c$, $0c \to c0$, $1c \to c1$, $Sc \to f$, $0f0 \to f$, $1f1 \to f$ to R.

For brevity, the following proofs omits some obvious details, which the reader can easily fill in. The next claim describes how M accepts each string from $L(M)$.

Claim 1. M accepts every $h \in L(M)$ in this way

$\$Szy_1y_2 \ldots y_{m-1}y_m\$$
$\Rightarrow \ \$g_0\langle q_0, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_0$
$\Rightarrow \ \$g_1\langle q_1, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_1$
$\vdots$
$\Rightarrow \ \$g_k\langle q_k, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_k$
$\Rightarrow \ \$g_k\langle q_k, 2\rangle y_1y_2 \ldots y_{m-1}y_m\$t_k$
$\Rightarrow \ \$g_k\langle q_{k+1}, 2\rangle y_1y_2 \ldots y_{m-1}y_m\$t_{k+1}$
$\Rightarrow \ \$g_k\langle q_{k+2}, 2\rangle y_2 \ldots y_{m-1}y_m\$t_{k+2}$
$\vdots$
${}_t\!\Rightarrow \ \$g_k\langle q_{k+m}, 2\rangle y_m\$t_{k+m}$
${}_t\!\Rightarrow \ \$g_kSo\$t_{k+m}S$
${}_r\!\Rightarrow \ \$g_kSot_{k+m}S\$$
${}_t\!\Rightarrow^\iota \ \$g_kSt_{k+m}oS\$$
${}_t\!\Rightarrow \ \$g_kSt_{k+m}c\$$
${}_t\!\Rightarrow^\iota \ \$u_1Sc\$v_1$
${}_t\!\Rightarrow \ \$u_1f\$v_1$
${}_r\!\Rightarrow \ \$u_1fv_1\$$
$\Rightarrow \ \$u_2fv_2\$$
$\vdots$
$\Rightarrow \ \$u_\varpi fv_\varpi\$$
$\Rightarrow \ \$f\$$

in M, where $k, m \geq 1$; $q_0, q_1, \ldots, q_{k+m} \in W - F$; $y_1, \ldots, y_m \in T^*$; $t_i \in \mu(q_0q_1 \ldots q_i)$ for $i = 0, 1, \ldots, k+m$; $g_j \in \nu(d_0d_1 \ldots d_j)$ with $d_1, \ldots, d_j \in (V-T)^*$ for $j = 0, 1, \ldots, k$; $d_0d_1 \ldots d_k = a_0a_1 \ldots a_{k+m}$ where $a_1, \ldots, a_{k+m} \in V - T$, $d_0 = a_0$, and $s = a_0q_0$; $g_k = t_{k+m}$ (notice that $\nu(a_0a_1 \ldots a_{k+m}) = \mu(q_0q_1 \ldots q_{k+m})$); $v_i \in \mathit{Prefix}(\mu(q_0q_1 \ldots q_{k+m}), |\mu(q_0q_1 \ldots q_{k+m})| - i)$ for $i = 1, \ldots, \upsilon$ with $\upsilon = |\mu(q_0q_1 \ldots q_{k+m})|$; $u_j \in \mathit{Suffix}(\nu(a_0a_1 \ldots a_{k+m}), |\nu(a_0a_1 \ldots a_{k+m})| - j)$ for $j = 1, \ldots, \varpi$ with $\varpi = |\nu(a_0a_1 \ldots a_{k+m})|$; $h = y_1y_2 \ldots y_{m-1}y_m$.

Proof of the Claim. Examine steps 1 through 6 of the construction of R. Notice that during every successful computation, M uses the rules introduced in step i before it uses the rules introduced in step $i+1$, for $i = 1, \ldots, 5$. Thus, in greater detail, every successful computation $\$Szh\$ \Rightarrow^* \$f\$$ can be expressed as

$\$Szy_1y_2 \ldots y_{m-1}y_m\$$
$\Rightarrow \ \$g_0\langle q_0, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_0$
$\Rightarrow \ \$g_1\langle q_1, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_1$
$\vdots$
$\Rightarrow \ \$g_k\langle q_k, 1\rangle y_1y_2 \ldots y_{m-1}y_m\$t_k$
$\Rightarrow \ \$g_k\langle q_k, 2\rangle y_1y_2 \ldots y_{m-1}y_m\$t_k$
$\Rightarrow \ \$g_k\langle q_{k+1}, 2\rangle y_1y_2 \ldots y_{m-1}y_m\$t_{k+1}$

$$\begin{aligned}
&\Rightarrow \ \$g_k\langle q_{k+2},2\rangle y_2y_3\dots y_{m-1}y_m\$t_{k+2}\\
&\Rightarrow \ \$g_k\langle q_{k+3},2\rangle y_3y_4\dots y_{m-1}y_m\$t_{k+3}\\
&\quad\vdots\\
&{}_t\Rightarrow \ \$g_k\langle q_{k+m},2\rangle y_m\$t_{k+m}\\
&{}_t\Rightarrow \ \$g_kSo\$t_{k+m}S\\
&\Rightarrow^* \ \$f\$
\end{aligned}$$

where $k,m \geq 1$; $h = y_1y_2\dots y_{m-1}y_m$; $q_0,q_1,\dots,q_{k+m} \in W-F$; $y_1,\dots,y_m \in T^*$; $t_i \in \mu(q_0q_1\dots q_i)$ for $i = 0,1,\dots,k+m$; $g_j \in \nu(d_0d_1\dots d_j)$ with $d_1,\dots,d_j \in (V-T)^*$ for $j = 0,1,\dots,k$; $d_0d_1\dots d_k = a_0a_1\dots a_{k+m}$ where $a_1,\dots,a_{k+m} \in V-T$, $d_0 = a_0$, and $s = a_0q_0$.

During $\$g_kSo\$t_{k+m}S \Rightarrow^* \$f\$$ only the rules of 6 are used. Recall these rules: $o0 \rightarrow 0o$, $o1 \rightarrow 1o$, $oS \rightarrow c$, $0c \rightarrow c0$, $1c \rightarrow c1$, $Sc \rightarrow f$, $0f0 \rightarrow f$, $1f1 \rightarrow f$. Observe that to obtain $\$f\$$ from $\$g_kSo\$t_{k+m}S$ by using these rules, M performs $\$g_kSo\$t_{k+m}S \Rightarrow^* \$f\$$ as follows

$$\begin{aligned}
&\$g_kSo\$t_{k+m}S\\
&{}_r\Rightarrow \ \$g_kSot_{k+m}S\$\\
&{}_t\Rightarrow^\iota \ \$g_kSt_{k+m}oS\$\\
&{}_t\Rightarrow \ \$g_kSt_{k+m}c\$\\
&{}_t\Rightarrow^\iota \ \$u_1Sc\$v_1\\
&{}_t\Rightarrow \ \$u_1f\$v_1\\
&{}_r\Rightarrow \ \$u_1fv_1\$\\
&\Rightarrow \ \$u_2fv_2\$\\
&\quad\vdots\\
&\Rightarrow \ \$u_\varpi fv_\varpi\$\\
&\Rightarrow \ \$f\$
\end{aligned}$$

in M, where $g_k = t_{k+m}$; $v_i \in \mathit{Prefix}(\mu(q_0q_1\dots q_{k+m}), |\mu(q_0q_1\dots q_{k+m})|-i)$ for $i = 1,\dots,\upsilon$ with $\upsilon = |\mu(q_0q_1\dots q_{k+m})|$; $u_j \in \mathit{Suffix}(\nu(a_0a_1\dots a_{k+m}), |\nu(a_0a_1\dots a_{k+m})| -j)$ for $j = 1,\dots,\varpi$ with $\varpi = |\nu(a_0a_1\dots a_{k+m})|$. This computation implies $g_k = t_{k+m}$. As a result, the claim holds. □

Let M accepts $h \in L(M)$ in the way described in the above claim. Examine the construction of R to see that at this point P contains $(a_0,q_0,z_0,q_1),\dots,(a_k,q_k,z_k,q_{k+1}),(a_{k+1},q_{k+1},y_1,q_{k+2}),\dots,(a_{k+m-1},q_{k+m-1},y_{m-1},q_{k+m}),(a_{k+m},q_{k+m},y_m,q_{k+m+1})$, where $z_1,\dots,z_k \in (V-T)^*$, so G makes the generation of h in the way described in Lemma 2. Thus $h \in L(G)$. Consequently, $L(M) \subseteq L(G)$.

Let G generates $h \in L(G)$ in the way described in Lemma 2. Then, M accepts h in the way described in the above claim, so $L(G) \subseteq L(M)$; a detailed proof of this inclusion is left to the reader.

As $L(M) \subseteq L(G)$ and $L(G) \subseteq L(M)$, $L(G) = L(M)$.

From the above Claim, it follows that M is a 2-self-reproducing pushdown transducer. Thus, Lemma 3 holds. □

Theorem 2. For every recursively enumerable language, L, there exists a 2-self-reproducing pushdown transducer, M, such that $Domain(T(M)) = L$ and $Range(T(M)) = \{\varepsilon\}$.

Proof. This theorem follows from Lemmas 1,2 and 3. □

(Received August 27, 2004.)

REFERENCES

[1] M. A. Harrison: Introduction to Formal Language Theory. Addison–Wesley, Reading 1978.
[2] H. C. M. Kleijn and G. Rozenberg: On the generative power of regular pattern grammars. Acta Inform. *20* (1983), 391–411.
[3] A. Meduna: Automata and Languages: Theory and Applications. Springer, London 2000.
[4] A. Meduna: Simultaneously one-turn two-pushdown automata. Internat. Computer Math. *80* (2003), 679–687.
[5] A. Meduna and D. Kolář: Regulated pushdown automata. Acta Cybernet. *14* (2000), 653–664.
[6] G. E. Revesz: Introduction to Formal Languages. McGraw–Hill, New York 1983.

Alexander Meduna and Luboš Lorenc, Brno University of Technology, Faculty of Information Technology, Božetěchova 2, 612 66 Brno. Czech Republic.
e-mails: meduna, lorenc@fit.vutbr.cz