INFORMATION AND CONTROL **11**, 323–340 (1967)

A Regularity Test for Pushdown Machines

R. E. STEARNS

General Electric Research and Development Center, P.O. Box 8, Schenectady, New York 12301

It is possible to test a deterministic pushdown machine to determine if the language it recognizes is regular.

The object of this paper is to show that, given a deterministic pushdown recognition machine, it is possible to determine if the set of input strings it recognizes is regular. In particular, we will show that if the set is regular, then the number of states in the reduced state machine which recognizes the set may be bounded by an expression of the order

$$t^{q^{q}}$$

(when q, $t > 1$) where q is the number of control states of the pushdown machine and t is the size of the pushdown tape alphabet. Therefore, one solution to our problem is to test all finite state machines of that size or less to see if one of them recognizes the same set as the pushdown machine.

The method of proof is to take the pushdown machine and extract a finite state machine which is equivalent to the pushdown machine whenever it recognizes a regular set. An alternate solution to the problem is to construct this candidate machine and test it. This improved method is also unsatisfactory as a practical algorithm, so we omit proof that this machine can be obtained constructively; the first solution being sufficient to establish our objective.

We spare the reader and the writer considerable hardship by defining the pushdown machine and proving the basic self-evident lemmas on a slightly informal basis. The symbol Λ will be used to represent a null sequence. ε

DEFINITION 1. A *general* (deterministic on-line) *pushdown machine* is a finite state control with the capability of reading inputs and storing an arbitrary string of symbols from finite tape alphabet X. When this string is non-null, the leftmost symbol is referred to as the *top symbol*; otherwise we call Λ the top symbol. The string is called a *tape word* as it may be pictured as being stored on a vertical Turing machine tape, the top symbol being under the reading head, and the remaining symbols stored below. A machine configuration c is represented as an ordered pair (s, ω) where s is from the set S of control states and $\omega = x_n \cdots x_1$ is the tape word from X^*, the set of strings over X. The machine changes from configuration to configuration under machine operations determined by the control state, top tape symbol, and sometimes an input symbol.

There are three kinds of pushdown machine *operations*; the pushdown operation, the write operation, and the pop-up operation. A *pushdown operation* consists of adding a new tape symbol to the left (top) of the stored tape word and changing the control state. A *write operation* consists either of replacing the non-null top symbol with a new tape symbol and changing control state or else changing control state without altering the (possibly null) tape word. A *pop-up operation* consists of deleting the leftmost symbol of a non-null tape word and changing control state.

With certain (*stable*) combinations of control state and top tape symbol, an input symbol is read and the next machine operation determined by the combination of input symbol, control state, and tape symbol. The remaining (*unstable*) combinations of state and tape symbol determine the next operation without reading an input. These latter operations are commonly called *ε-moves*. If input a in A is read and configuration c_1 changes to configuration c_2 under the resulting operation, we write

$$c_1 \xrightarrow{a} c_2 .$$

If c_1 changes to c_2 under an ϵ-move or if $c_1 = c_2$, we write

$$c_1 \xrightarrow{\Lambda} c_2 .$$

This notation extends inductively to sequences of inputs under the following rule:

$$c_1 \xrightarrow{\alpha_1} c_2 \quad \text{and} \quad c_2 \xrightarrow{\alpha_2} c_3 \quad \text{implies} \quad c_1 \xrightarrow{\alpha_1\alpha_2} c_3 ,$$

where c_1, c_2, and c_3 are configurations, α_1 and α_2 are input strings, and $\alpha_1\alpha_2$ is the concatenation of α_1 and α_2.

DEFINITION 2. A *pushdown recognition machine* is a general pushdown machine with a designated *starting configuration* c_0 with null tape word and a designated subset of the stable combinations in $S \times (X \cup \{\Lambda\})$

called *accepting combinations.* Those configurations which have an accepting combination of control state and top tape symbol are called *accepting configurations.* A sequence of inputs α is said to be *accepted* or *recognized* by the machine if and only if

$$c_0 \xrightarrow{\alpha} c_1 ,$$

for some accepting configuration c_1. The set of all α accepted by the machine is called the set *recognized* by the machine.

Pushdown machines are sometimes defined to allow slightly more general operations such as pushing down a string of tape symbols or writing and pushing in a single operation. These variations are easily simulated on our type pushdown machine, so no generality is lost. Similarly, the case of a starting configuration with a non-null tape word is no problem either.

The essential notation introduced above may be summarized as follows:

	Set	Element	String	Set Size
Input	A	a	α	—
State	S	s	—	q
Tape	X	x	ω	t

Configuration: $c = (s, \omega)$ or $c = (s_n , x_n \cdots x_1)$
Starting configuration: c_0
Null string: Λ

A NON-REGULARITY CONDITION

In this section, we give a condition for non-regularity that we plan to exploit in the main proof. First, we must define an equivalence relation on A^*, the set of all input strings.

L-equivalence

DEFINITION 3. For a given language L over alphabet A, we write $\alpha_1 \approx \alpha_2$ for α_1 and α_2 in A^* if and only if α_1 and α_2 are either both in L or both not in L. We write $\alpha_1 \not\approx \alpha_2$ otherwise.

THEOREM 1. *A language L over alphabet A is non-regular if, for some α_1, α_2, α_3, α_4, and α_5 in A^*, the following two conditions hold:*

(i) for all $i, j, k \geqq 0$,
$\alpha_1\alpha_2^{\,i}\alpha_3\alpha_4^{\,j}\alpha_5 \approx \alpha_1\alpha_2^{\,i+k}\alpha_3\alpha_4^{\,j+k}\alpha_5$;

(ii) there exists an ℓ such that for all $i \geqq \ell$,
$\alpha_1\alpha_2^{\,i}\alpha_3\alpha_5 \not\approx \alpha_1\alpha_3\alpha_5$.

Proof. Suppose that there is a finite state machine M that recognizes L. For each integer n, let s_n be the state of M that results from input

Bounded languages suffice to witness non-regularity of context free languages.

sequence $\alpha_1\alpha_2^{n\ell}$ where α_1, α_2, and ℓ are as in condition (ii). If $n_1 < n_2$, then states s_{n_1} and s_{n_2} can be distinguished by the sequence $\alpha_3\alpha_4^{n_1\ell}\ \alpha_5$ because

$$(\alpha_1\alpha_2^{n_1\ell})(\alpha_3\alpha_4^{n_1\ell}\ \alpha_5) \approx \alpha_1\alpha_3\alpha_5$$

and

$$(\alpha_1\alpha_2^{n_2\ell})(\alpha_3\alpha_4^{n_1\ell}\alpha_5) \approx \alpha_1\alpha_2^{(n_2-n_1)\ell}\alpha_3\alpha_5,$$

by condition (i) and

$$\alpha_1\alpha_3\alpha_3 \not\approx \alpha_1\alpha_2^{(n_2-n_1)\ell}\alpha_3\alpha_5,$$

by condition (ii). But this means that M has an infinite number of states, contrary to our hypothesis. Q.E.D.

The effect of our proof will be to show that Theorem 1 becomes an "if and only if" result when L is a set recognized by a pushdown machine. Thus a non-regular pushdown language has a non-regular context-free subset which is bounded in the sense of Ginsburg and Spanier (1964).

BASIC RELATIONS

The primary purpose of this section is to define two relations $\downarrow(\alpha)$ and $\uparrow(\alpha)$ and derive some of their basic properties. These relations are both special cases of the relation $\xrightarrow{\alpha}$, the first being a generalized pushdown and the other a generalized pop-up.

DEFINITION 4. If α is an input sequence and c and c' are configurations, we write

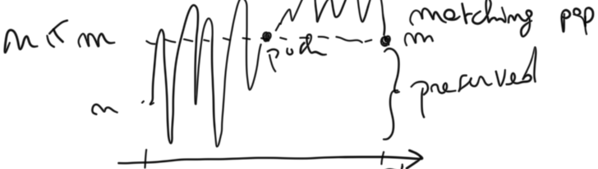
$$\boxed{c \downarrow(\alpha)\ c',}$$

if and only if there is a sequence of configurations $c_1 \cdots c_r$ and corresponding a_i in $A \cup \{\Lambda\}$ for $1 \leqq i < r$ such that $c_1 = c$, $c_r = c'$, each c_j for $r \geqq j > 1$ has a longer tape word than c and results from c_{j-1} by a single operation with input a_{j-1}, and α is the concatenation of the a_k (i.e. $\alpha = a_1 \cdots a_{r-1}$ if $r > 1$ and $\alpha = \Lambda$ if $r = 1$).

DEFINITION 5. If α is an input sequence and c and c' are configurations, we write

$$\boxed{c \uparrow(\alpha)\ c',}$$

if and only if there is a sequence of configurations $c_1 \cdots c_r$ and corresponding a_i in $A \cup \{\Lambda\}$ for $1 \leqq i < r$ such that $c_1 = c$, $c_r = c'$, each c_j for $r > j \geqq 1$ has a longer tape word than c', each c_j for $r \geqq j > 1$ results



from c_{j-1} by a single operation with input a_{j-1}, and α is the concatenation of the a_k.

Note that $c \uparrow(\Lambda)\ c$ and $c \downarrow(\Lambda)\ c$ always hold since we can take $c = c_1$ and $r = 1$.

The first lemma relates the new relations to the previously defined relation $\xrightarrow{\alpha}$.

LEMMA 1. *If* $c = (s, x_n \cdots x_1)$ *and* $c' = (s', x_m' \cdots x_1')$ *are configurations and* α *is an input sequence such that* $c \xrightarrow{\alpha} c'$, *then*

(i) $n \leqq m$ *implies there exists an* s_n *in* Q *and* α_1 *and* α_2 *in* A^* *such that* $c \xrightarrow{\alpha_1} (s_n, \boxed{x_n' \cdots x_1'}) \downarrow(\alpha_2)\ c'$ *and* $\alpha = \alpha_1\alpha_2$; latest conf. of size m

(ii) $n \geqq m$ *implies there exist a unique* s_m *in* Q *and unique* α_1 *and* α_2 *in* A^* *such that* $c \uparrow(\alpha_1)\ (s_m, x_m \cdots x_1) \xrightarrow{\alpha_2} c'$.

Proof. The relation $c \xrightarrow{\alpha} c$ implies that there is some sequence of configurations $c_1 \cdots c_r$ and corresponding a_i in $A \cup \{\Lambda\}$ for $1 \leqq i < r$ such that $c_1 = c$, $c_r = c'$, and $a_1 \cdots a_r = \alpha$.

In case (i), we choose c_k to be the last configuration of this series with tape word of length n. We let s_n be the state of c_k, $\alpha_1 = a_1 \cdots a_{k-1}$, and $\alpha_2 = a_k \cdots a_{r-1}$. In going from c_k to c_r, there was no opportunity for the tape symbols of c_k to be altered and so the tape word of c_k must be precisely $x_n' \cdots x_1'$. The sequence $c_k \cdots c_r$ satisfies Definition 4 and so (i) is proved.

In case (ii), we choose c_k to be the first configuration of the series with tape word of length m, let s_m be the state of c_k, and let $\alpha_1 = a_1 \cdots a_{k-1}$ and $\alpha_2 = a_k \cdots a_{r-1}$. There is no opportunity for changing symbols of c_k between c_1 and c_k and so the tape word of c_k is precisely $x_m \cdots x_1$. Because the machine is deterministic and because c_k occurs prior to the first occurrence of c' in the sequence $c_1 \cdots c_r$, α and c_1 determine s_m and α_1 uniquely. Q.E.D.

Except for certain subcases of the case where c' goes into itself under a non-trivial sequence of ϵ-moves, the s_n, α_1, and α_2 of part (i) are also unique, although we have no application of this fact here. There are never more than two possible values for s_n and α_1.

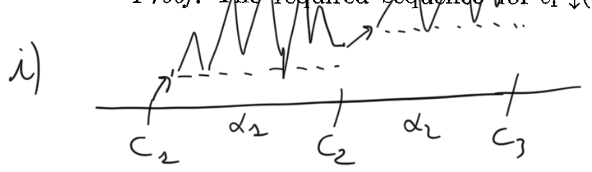
The next lemma shows that the defining property of the relation $\xrightarrow{\alpha}$ also holds for the stronger relations $\downarrow(\alpha)$ and $\uparrow(\alpha)$.

LEMMA 2. (*Concatenation property*) *For all configurations* c_1, c_2, *and* c_3 *and all* α_1 *and* α_2 *in* A^*,

(i) $c_1 \downarrow(\alpha_1)\ c_2$ *and* $c_2 \downarrow(\alpha_2)\ c_3$ *implies* $c_1 \downarrow(\alpha_1\alpha_2)\ c_3$;

(ii) $c_1 \uparrow(\alpha_1)\ c_2$ *and* $c_2 \uparrow(\alpha_2)\ c_3$ *implies* $c_1 \uparrow(\alpha_1\alpha_2)\ c_3$.

Proof. The required sequence for $c_1 \downarrow(\alpha_1\alpha_2)\ c_3$ is obtained by taking



the sequence for $c_1 \downarrow(\alpha_1)\ c_2$ and extending it with the sequence for $c_2 \downarrow(\alpha_2)\ c_3$. The proof of (ii) is similar. Q.E.D.

LEMMA 3. (*Independence property*) *For all control states s and s', all tape words $\omega_1, \omega_2, \omega_3$, and all input words α,*

(i) *$(s, \omega_2) \downarrow(\alpha)\ (s', \omega_1\omega_2)$ implies $(s, \omega_3) \downarrow(\alpha)\ (s', \omega_1\omega_3)$ whenever ω_2 and ω_3 have the same first symbol;*

(ii) *$(s, \omega_1\omega_2) \uparrow(\alpha)\ (s', \omega_2)$ implies $(s, \omega_1\omega_3) \uparrow(\alpha)\ (s', \omega_3)$.*

Proof. The sequence of operations required by Definition 4 are completely determined by the top symbol of ω_2 as ω_2 is simply pushed down and not looked at again. Thus the machine will do the same with any ω_3 that has the same top symbol as ω_2. In part (ii), ω_2 does not affect the intermediate operations at all and thus any substitute for ω_2 would cause the identical sequence of operations and result in the corresponding configuration. Q.E.D.

LEMMA 4. (*Factor property*) *Let c and c' be configurations with tape words of length n and m respectively, let $x_n \cdots x_1$ be the tape word of c, and let α be an input sequence.*

(i) *If $c' \downarrow(\alpha)\ c$, then $n \geqq m$ and there exist control states s_i for $m \leqq i \leqq n$ and input sequences α_{ij} for $m \leqq i \leqq j \leqq n$ such that for $m \leqq i \leqq j \leqq k \leqq n$*

(a) $c = (s_n, x_n \cdots x_1)$,

(b) $c' = (s_m, x_m \cdots x_1)$,

(c) $\alpha = \alpha_{mn}$,

(d) $(s_i, x_i \cdots x_1) \downarrow(\alpha_{ij})\ (s_j, x_j \cdots x_1)$,

(e) $\alpha_{ij}\alpha_{jk} = \alpha_{ik}$.

(ii) *If $c \uparrow(\alpha)\ c'$, then $n \geqq m$ and there exist unique control states s_i for $m \leqq i \leqq n$ and unique input sequences α_{ij} for $n \geqq i \geqq j \geqq m$ such that for $n \geqq i \geqq j \geqq k \geqq m$*

(a) $c = (s_n, x_n \cdots x_1)$,

(b) $c' = (s_m, x_m \cdots x_1)$,

(c) $\alpha = \alpha_{nm}$,

(d) $(s_i, x_i \cdots x_1) \uparrow(\alpha_{ij})\ (s_j, x_j \cdots x_1)$,

(e) $\alpha_{ij}\alpha_{jk} = \alpha_{ik}$.

Proof. The relation $c' \downarrow(\alpha)\ c$ implies at once that $n \geqq m$. Letting $c_1 \cdots c_r$ be a sequence of configurations and $a_1 \cdots \alpha_{r-1}$ a sequence of inputs which satisfy Definition 4 in justification of the relation $c' \downarrow(\alpha)\ c$, let $\sigma(i)$ be the index of the last configuration in the series with tape length i. Let s_i for $m \leqq i \leqq n$ be the control state of configuration $c_{\sigma(i)}$. The fact that $c_k = c$ has tape length n insures that $\sigma(n) = r$. Therefore,

s_n is the control state of $c = c_{\sigma(n)}$ and equation (a) is established. None of the inputs $a_{\sigma(i)}$ to a_{r-1} can change the i tape symbols of $c_{\sigma(i)}$ since these inputs result in configurations with more than i tape symbols. This means that the i tape symbols at $c_{\sigma(i)}$ are just the last i tape symbols of c, i.e.,

$$c_{\sigma(i)} = (s_i, x_i \cdots x_i),$$

for $m \leqq i \leqq n$. Since c_1 is the last configuration in the series with tape length m, $\sigma(m) = 1$ and $c_{\sigma(m)} = c'$. Equation b is a statement of this fact. Now for $m \leqq i \leqq m$, we define $\alpha_{ii} = \Lambda$ and for $m \leqq i < j \leqq n$ we define

$$\alpha_{ij} = a_{\sigma(i)} \cdots a_{\sigma(j)-1} .$$

This definition is valid since $i < j$ implies $\sigma(i) < \sigma(j)$. Equation (e) is immediate from this definition and (c) follows from the fact that $\sigma(m) = 1$ and $\sigma(n) = r$. Relation (d) says that $c_{\sigma(i)} \downarrow(\alpha_{ij})\ c_{\sigma(j)}$ and this is true because $c_{\sigma(i)} \cdots c_{\sigma(j)}$ and corresponding a_t satisfy Definition 4.

For part (ii) take the configuration series $c_1 \cdots c_r$ of Definition 5 and let $\sigma(i)$ be the index of the first configuration in the series with tape length i. Now define s_i and α_{ij} as above using this new series and the equations follow as before. As in the proof of Lemma 1, the deterministic nature of the machine insures that the s_i and α_{ij} are unique.

Q.E.D.

The uniqueness of the configuration sequence associated with $c \uparrow(\alpha)\ c'$ implies some further special properties. Analogous results hold for the non-pathological pushdown cases, but they are not needed.

LEMMA 5. *For configurations c_1, c_2, c_3, input words α, α_1, α_2, and integer n;*

(i) *$c_1 \uparrow(\alpha_1)\ c_2$ and $c_1 \uparrow(\alpha_1\alpha_2)\ c_2$ implies $\alpha_2 = \Lambda$;*

(ii) *$c_1 \uparrow(\alpha)\ c_2$ and $c_1 \uparrow(\alpha)\ c_3$ implies that $c_2 \uparrow(\Lambda)\ c_3$ or $c_3 \uparrow(\Lambda)\ c_2$;*

(iii) *there is at most one configuration c' with tape word of length n such that $c_1 \uparrow(\alpha)\ c'$.*

Proof. No continuation of the configuration sequence for $c_1 \uparrow(\alpha_1)\ c_2$ can be used to justify $c_1 \uparrow(\alpha_1\alpha_2)\ c_2$ as c_2 has the same tape length as itself. Therefore, α_2 must be Λ. The configuration sequence associated with $c_1 \uparrow(\alpha)\ c_2$ must be a prefix of that sequence associated with $c_1 \uparrow (\alpha)c_3$ in which case $c_2 \uparrow(\Lambda)\ c_3$, or the reverse must hold in which case $c_3 \uparrow(\Lambda)\ c_2$. Configuration c' must be the first configuration of length n (if any) resulting from c_1 under input word α. Q.E.D.

Finally, we relate the pop-up relation to distinguishing sequences. Input sequence α is said to *distinguish* configurations c_1 and c_2 if α carries exactly one of the configurations into an accepting configuration.

LEMMA 6. *If α distinguishes between configurations $c_1 = (s, \omega\omega_1)$ and $c_2 = (s, \omega\omega_2)$, then ~~there are input sequences α_1 and α_2~~ and state s' such that* there is a prefix α_1 of α s.t.

(a) $c_1 \uparrow(\alpha_1)\ (s', \omega_1)$;
(b) $c_2 \uparrow(\alpha_1)\ (s', \omega_2)$;
~~(c) $\alpha_1\alpha_2 = \alpha$.~~

Proof. Because α must distinguish between c_1 and c_2, it must cause both configurations to pop up enough tape symbols to reach ω_1 and ω_2 respectively. Letting α_1 be the substring which causes c_1 to do this and letting (s', ω_1) be the first configuration with tape length equal to the length of ω_1, we have relation (a) immediately. Relation (b) follows from the independence property and (c) follows when we let α_2 be the remainder of α. Q.E.D.

NULL TRANSPARENT WORDS

We now consider a special type of tape word which goes into the central proof.

DEFINITION 6. A word ω in X^* is called null transparent if and only if for all s and s' in S,

$$(s, \omega) \uparrow(\Lambda)\ (s', \Lambda) \quad \text{implies} \quad (s', \omega) \uparrow(\Lambda)\ (s', \Lambda).$$

The key property of null transparent words is that if such a word is popped up by a series of ϵ-moves, any additional copies of the word will be eliminated by additional ϵ-moves and the control state entered will be independent of the number of copies eliminated. Thus all the information as to the number of additional copies is wiped out. In short, if one copy is popped with ϵ-moves, all are popped. This property may be stated more usefully as follows.

THEOREM 2. *Suppose $c = (s, \omega\omega_1)$ is a configuration and ω is null transparent. For each α in A^*, there is an integer ℓ such that α cannot distinguish between*

$$(s, \omega^i\omega_1) \quad \textit{and} \quad (s, \omega^j\omega_1),$$

for all $i, j \geqq \ell$.

Proof. Choose ℓ to be one greater than the length of α. Assume that $i \geqq j \geqq \ell$ and that α does distinguish between $c_i = (s, \omega^i\omega_1)$ and $c_j =$

for DPDA without ε-moves all $w \neq \varepsilon$ are null-transparent because $(s, w) \xrightarrow{\varepsilon}{}^* (s', w')$ implies $s = s'$ and $w = w' \neq \varepsilon$

$(s, \omega^j\omega_1)$. Because

$$c_i = (s, \omega^j(\omega^{i-j}\omega_1)),$$

it follows from Lemma 6 that there must be α_1, α_2, and s_0 such that

$$(s, \omega^j\omega_1) \uparrow (\alpha_1) \ (s_0, \omega_1)$$

and $\alpha_1\alpha_2 = \alpha$. It follows from the factor property (Lemma 4) that there exist α'_k and s_k for $j \geqq k \geqq 1$ such that α_1 may be written uniquely as

$$\alpha_1 = \alpha'_j \cdots \alpha'_1,$$

where

$$(s_k, \omega^k\omega_1) \uparrow (\alpha'_k) \ (s_{k-1}, \omega^{k-1}\omega_1),$$

for $j \geqq k \geqq 1$. Since the number of symbols in α is less than j, one of the α'_i must be null, say α'_m. Applying the concatenation property (Lemma 2) to Definition 6,

$$(s_m, \omega^{i-j}) \uparrow (\Lambda) \ (s_{m-1}, \Lambda).$$

Applying the independence property (Lemma 3) gives

$$(s_m, \omega^{i-j+m}\omega_1) \uparrow (\Lambda) \ (s_{m-1}, \omega^{m-1}\omega_1).$$

Also

$$(s, \omega^i\omega_1) \uparrow (\alpha'_j \cdots \alpha'_m) \ (s_m, \omega^{i-j+m}\omega_1),$$

which together with

$$(s_{m-1}, \omega^{m-1}\omega_1) \uparrow (\alpha'_{m-1} \cdots \alpha'_1) \ (s_0, \omega_1), \qquad (\text{if } m > 1),$$

yields

$$c_i \uparrow (\alpha_1) \ (s_0, \omega_1),$$

by concatenation. No proper prefix of α_1 can distinguish c_i and c_j because then a proper prefix α'_1 of α_2 would satisfy

$$c_i \uparrow (\alpha'_1) \ (s_0, \omega_1),$$

in violation of Lemma 5(i). Since α_1 carries both c_i and c_j into (s_0, ω_1), no continuation of α_1 can distinguish c_i from c_j. Thus $\alpha = \alpha_1\alpha_2$ cannot distinguish c_i from c_j, contrary to our assumption. Q.E.D.

A second important property of null transparent words is that they may be found embedded in any tape word of sufficient length. This may be stated more generally as follows:

THEOREM 3. *If* $x_n \cdots x_1$ *is a tape word and* N *is a set of at least* $q! + 1$ *distinct integers less than* n, *then there exist* e *and* f *in* N, $e > f$, *such that* $x_e \cdots x_{f+1}$ *is null transparent.*

Proof. We will say that state s has property P with respect to N if and only if

(a) $(s, x_i \cdots x_1) \uparrow(\Lambda)\ (s, x_j \cdots x_1)$,

for all i and j in N such that $i > j$.

For purposes of induction, we consider case m, $i \leqq m \leqq q$, where the set of integers N_m has at least $m! + 1$ elements and at most m states of Q do not have property P. The case $m = q$ is just a statement of the theorem. We will show that in those cases where the max and min of N_m are not suitable e and f, the problem may be reduced to solving the case $m - 1$ for a subset of N_m. The max and min of N_1 will be shown to be always suitable and the theorem will therefore be true by induction.

Let e and f be the maximum and minimum of N_m. Because N_m has at least two members, $e > f$. If $x_e \cdots x_{f+1}$ is not null transparent, let s_e and s_f be the states such that

(b) $(s_e, x_e \cdots x_{f+1}) \uparrow(\Lambda)\ (s_f, \Lambda)$,

but not

(c) $(s_f, x_e \cdots x_{f+1}) \uparrow(\Lambda)\ (s_f, \Lambda)$.

State s_f cannot have property P because relation (a) with $i = e, j = f$, and $s = s_f$ implies relation (c) by the independence property.

Relation (b) implies, by independence, that

$$(s_e, x_e \cdots x_1) \uparrow(\Lambda)\ (s_f, x_f \cdots x_1).$$

Factoring this relation according to Lemma 4, we consider some s_i for i in N_m. Because $\alpha_{if} = \Lambda$, state s_i cannot have property P, as this would imply $s_f = s_i$ by relation (a) and Lemma 5iii and we have already shown that s_f does not have property P. In case $m = 1$, all these s_i must be the same state, namely the state without property P, and s_e must equal s_f making relations (b) and (c) identical. This is contrary to the assumption that (b) is true and (c) is false and we conclude that e and f do satisfy the theorem for case $m = 1$. In case $m > 1$, divide N_m into m-equivalence classes according to the relationship

$$i \equiv j \quad \text{if and only if} \quad s_i = s_j\,.$$

One of these classes must have at least $(m - 1)! + 1$ elements (since

N_m has more than $m(m - 1)!$ elements) and we call this set N_{m-1}. The m states which had property P with respect to N_m also have property P with respect to subset N_{m-1} and the state s which determined the equivalence class N_{m-1} also has property P since

$$(s, x_i \cdots x_{f+1}) \uparrow(\Lambda) \ (s, x_j \cdots x_{f+1}),$$

implies (a) by the independence property. Therefore case m has been reduced to case $m - 1$ and the theorem is proven. Q.E.D.

COROLLARY 3.1. *For pushdown machines without ϵ-moves, Theorem 3 holds whenever N has 2 elements.*

Proof. In this case, all words satisfy Definition 6.

ℓ-INVISIBILITY

We now seek a way of finding certain segments in the tape word of a large configuration such that the presence of such a segment cannot be detected by the machine without using non-null input words at least ℓ times to pop up the tape symbols above the segment. Stated formally, we are interested in the following property:

DEFINITION 7. A segment $x_e \cdots x_{f+1}$ is said to be *ℓ-invisible* in the configuration

$$c = (s_n, x_n \cdots x_e \cdots x_f \cdots x_1),$$

if and only if, for each α and s' such that

$$c \uparrow(\alpha) \ (s', x_e \cdots x_1),$$

either

$$c \uparrow(\alpha) \ (s', x_f \cdots x_1),$$

or there are at least ℓ integers i, $n \geqq i > \ell$ such that the $\alpha_{i,i-1}$ of Lemma 4 (factor property) applied to the relation

$$c \uparrow(\alpha) \ (s', x_e \cdots x_1),$$

satisfy $\alpha_{i,i-1} \neq \Lambda$.

The existence of ℓ-invisible segments in large configurations is assured by the following:

THEOREM 4. *For given integer ℓ, there exists a bound $B(\ell)$ of order $(q^q)^\ell$ (for $q > 1$) such that, if $c = (s, x_n \cdots x_1)$ is a configuration and N is a set of at least $B(\ell)$ distinct integers i, $1 \leqq i \leqq n$, then there exist e and f in N such that $e > f$ and $x_e \cdots x_{f+1}$ is ℓ-invisible in c. This $B(\ell)$*

may be defined by the expression

$$[(q^{\ell+1} - 1)/(q - 1) + 1]\, q^{q} + 1,$$

for $q > 1$ *and by* $\ell + 3$ *if* $q = 1$.

Proof. For given state s and integer $i \leqq n$, we define $f(s, i)$ to be the smallest j such that

$$(s, x_i \cdots x_1) \uparrow(\Lambda)\ (s_j, x_j \cdots x_1),$$

for some state s_j. Since this relation holds for $j = i$ and $s_j = s$, $f(s, i)$ is well defined and $f(s, i) \leqq i$.

Now define I_k for $k \geqq 0$ inductively by the following:

$$I_0 = \{f(s_n, n)\}$$

$$I_{k+1} = \{m \mid m = f(s, i - 1) \text{ for some } s \text{ in } Q \text{ and } i \text{ in } I_k\}.$$

Since each element of set I_k determines at most q additions to I_{k+1} (i.e. one for each s in Q) and since I_0 has one element, I_k certainly has no more than q^k elements. Let

$$\mathcal{J} = \bigcup_{0 \leqq k \leqq \ell} I_k .$$

Because $\mathcal{J}$ has at most $z = (q^{\ell+1} - 1)/(q - 1)$ elements (or $z = \ell + 1$ if $q = 1$) it follows that if N has at least $(z + 1)q^q + 1$ elements, and there must be some i_0 and j_0 such that the set

$$\bar{N} = \{k \mid k \text{ in } N \text{ and } j_0 \leqq k < i_0\},$$

has at least $q^q + 1$ elements and k is not in $\mathcal{J}$ for $j_0 < k < i_0$. For each i in $\bar{N}$, let Q_i be the set of states s_i such that either

$$c \uparrow(\Lambda)\ (s_i, x_i \cdots x_1)$$

or

$$(s_{i'}, x_{i'}, \cdots x_1) \uparrow(\Lambda)\ (s_i, x_i \cdots x_1),$$

for some $s_{i'}$ in Q and $i' + 1$ in $\mathcal{J}$. By choice of $\bar{N}$, there must be a $j' \leqq j_0$ and $s_{j'}$ such that

$$c \uparrow(\Lambda)\ (s_{j'}, x_{j'} \cdots x_1)$$

or

$$(s_{i'}, x_{i'} \cdots x_1) \uparrow(\Lambda)\ (s_{j'}, x_{j'} \cdots x_1).$$

Because the elements of $\bar{N}$ are between i' and j' and because s_i is an arbitrary element of Q_i, the factor property and Lemma 5iii imply that for all i in $\bar{N}$, s_i in Q_i, and j in $\bar{N}$ such that $j < i$, there exists an s_j in Q_j such that

$$(s_i, x_i \cdots x_1) \uparrow(\Lambda) \ (s_j, x_j \cdots x_1).$$

Let m be the max of $\bar{N}$ and, for each i in $\bar{N}$ and s_m in Q_m, let $g(s_m, i)$ be the s_i in Q_i such that

$$(s_m, x_m \cdots x_1) \uparrow(\Lambda) \ (s_i, x_i \cdots x_1).$$

Function g is unique by Lemma 5iii. Because $\bar{N}$ has $q^q + 1$ elements, there must be e and f in N such that $e > f$ and

$$g(s_m, e) = g(s_m, f),$$

for all s_m in Q_m. We now wish to show that $x_e \cdots x_{f+1}$ is the desired segment.

The important property of e and f is that for all s_e in Q_e,

(a) $(s_e, x_e \cdots x_1) \uparrow(\Lambda) \ (s_e, x_f \cdots x_1)$.

To see this, recall that for s_e in Q_e, there are i' and j' defined above such that i' is in $\mathcal{I}$, $i' \geqq m \geqq e > f \geqq j'$,

$$c \uparrow(\Lambda) \ (s_{j'}, x_{j'} \cdots x_1)$$

or

$$(s_{i'}, x_{i'} \cdots x_1) \uparrow(\Lambda) \ (s_{j'}, x_{j'} \cdots x_1)$$

and the s_e in the factorization of this relation is the given s_e. It follows from $\alpha_{me} = \Lambda$ and $\alpha_{mf} = \Lambda$ that $s_e = g(s_m, e) = g(s_m, f) = s_f$ and since $\alpha_{ef} = \Lambda$, the desired relation is established.

Consider some α such that

(b) $c \uparrow(\alpha) \ (s_e, x_e \cdots x_1)$

for some s_e in S and let the α_{ij} be defined as in Lemma 4 (factor property) and let r be the number of non-null $\alpha_{i,i-1}$ for $n \geqq i > e$. If $r > \ell$, then α automatically satisfies Definition 7. If $r = 0$, then

$$c \uparrow(\Lambda) \ (s_e, x_e \cdots x_1),$$

for some s_e, s_e is in Q_e by definition, and so

$$c \uparrow(\Lambda) \ (s_e, x_f \cdots x_1),$$

by concatenating (a) and (b), and Definition 7 is again satisfied. Now suppose that $0 < r \leqq \ell$ and for each k, $0 \leqq k \leqq r$, let i_k be the integer such that α_{i_k, i_k-1} is the $(k + 1)$th non-null input word in the series

$$\alpha_{n,n-1} \cdots \alpha_{i,i-1} \cdots \alpha_{e+1,e} .$$

The key property of the i_k is that i_k is in I_k. This follows inductively from the relations

$$f(s_n, n) = i_0 \quad \text{and} \quad f(s_{i_k-1}, i_k - 1) = i_{k+1}$$

which are derived from the relations

$$\alpha_{n,i_0} = \alpha_{i_k-1, i_{k+1}} = \Lambda$$

and from Lemma 5i. Now observe that

$$\alpha_{i_r-1,e} = \Lambda$$

and so s_e is in Q_e. Again,

$$c \uparrow (\alpha)\ (s_e, x_f \cdots x_1),$$

by concatenating relations (a) and (b). Thus Definition 7 is established for all r and the theorem is proved. Q.E.D.

COROLLARY 4.1. *If the pushdown machine has no ϵ-moves, then Theorem 4 is true for $B(\ell) = \ell + 2$.*

Proof. All the $\alpha_{i,i-1}$ are non-null.

MAIN RESULTS

The key to all our solvability results is contained in the following theorem. Two configurations are called *equivalent* if there are no input sequences which distinguish them.

THEOREM 5. *If a pushdown machine recognizes a regular set, one can calculate a bound M of order q^{q^q} such that if $c_0 \xrightarrow{\alpha} c$, there is a configuration c' equivalent to c such that c' has less than M tape word symbols. Bound M may be given by*

$$M = tqB(q!(q^2t) + 1) + 1,$$

where B is given in Theorem 4.

Proof. Assume that $c_0 \xrightarrow{\alpha} c$ where

$$c = (s_n, x_n \cdots x_1)$$

is a configuration with $n \geqq M$. It is sufficient to show that there is a configuration c' equivalent to c which has a shorter tape word than c.

By Lemma 1, there exist input sequences α' and β and state s_0 such that $\alpha = \alpha'\beta$ and

$$c_0 \xrightarrow{\alpha'} (s_0, \Lambda) \downarrow(\beta)\ c.$$

We factor this relation according to Lemma 4 using β_{ij} to represent the input strings and s_i the states. For each x in X and s in S, let

$$N(x, s) = \{i \mid 1 \leqq i \leqq n,\ x_i = x, \quad \text{and} \quad s_i = s\}.$$

Because of the size of M, there is some $\bar{x}$ and $\bar{s}$ such that $N(\bar{x}, \bar{s})$ has $B(q!(q^2t) + 1) + 1$ elements. Therefore, according to Theorem 4, there are e and f in $N(\bar{x}, \bar{s})$ such that $x_e \cdots x_{f+1}$ is $(q!(q^2t) + 1)$-invisible in c. We claim that

$$c' = (s_n, x_n \cdots x_{e+1}x_f \cdots x_1)$$

is the desired equivalent configuration.

Defining $\beta' = \beta_{1f}\beta_{en}$, observe that

$$(s_0, \Lambda) \downarrow(\beta')\ c' \quad \text{and} \quad c_0 \xrightarrow{\alpha'\beta'} c',$$

because

$$(s_f, x_f \cdots x_1) \downarrow(\beta_{en})\ c',$$

by the independence property and because of the concatenation property.

Assume, to the contrary, that c and c' are not equivalent. Let γ be the shortest input sequence that distinguishes c and c'. Note that γ is therefore the shortest sequence such that $\alpha'\beta\gamma \not\approx \alpha'\beta'\gamma$. By Lemma 6, γ may be written $\gamma = \Delta\gamma'$ where

$$c \uparrow(\Delta)\ (s'_e, x_e \cdots x_1)$$

and

$$c' \uparrow(\Delta)\ (s'_f, x_f \cdots x_1).$$

We factor this first relation using Lemma 4 where Δ_{ij} is used to represent one of the input sequences and s'_i to represent one of the states. Since segment $x_e \cdots x_{f+1}$ is $(q^2t(q!) + 1)$-invisible, the set

$$N = \{i \mid \Delta_{i,i-1} \neq \Lambda \quad \text{and} \quad n \geqq i > e\}$$

has at least $q^2t(q!) + 1$ elements, for otherwise

$$c \uparrow(\Delta)\ (s'_f, x_f \cdots x_1),$$

by Definition 7 and Lemma 5iii, which would imply that Δ carries c and c' into the same configuration contrary to the fact that $\Delta\gamma'$ distinguishes c and c'.

Because of the size of N, there must be s and s' in S and x in X such that

$$N(s, s', x) = \{i \text{ in } N \mid s_i = s, s'_i = s', \quad \text{and} \quad x_i = x\}$$

has at least $q! + 1$ elements. By Theorem 3, there is an e' and f' in $N(s, s', x)$ such that $x_e \cdots x_{f+1}$ is null transparent.

In order to consolidate notation, we define

$$\begin{aligned} \theta_1 &= \alpha'\beta_{1f'} \\ \theta'_1 &= \alpha'\beta_{1f}\beta_{ef'} \\ \theta_2 &= \beta_{f'c'} \\ \theta_3 &= \beta_{e'n}\Delta_{ne'} \\ \theta_4 &= \Delta_{e'f'} \\ \theta_5 &= \Delta_{f'e}\gamma'. \end{aligned}$$

By straightforward application of the independence and concatenation properties

$$\text{(a)} \quad \theta_1\theta_2^{\,i}\theta_3\theta_4^{\,j}\theta_o \approx \theta_1\theta_2^{i+k}\theta_3\theta_4^{j+k}\theta_5$$

for all i, j and k since both input sequences lead to the same configuration as each θ_4 effectively cancels a θ_2. Similarly one can verify

$$\text{(b)} \quad \theta'_1\theta_2^{\,i}\theta_3\theta_4^{\,j}\theta_5 \approx \theta_1\theta_2^{i+k}\theta_3\theta_4^{j+k}\theta_5$$

for all i, j and k.

Because γ distinguishes c and c',

$$\text{(c)} \quad \theta_1\theta_2\theta_3\theta_4\theta_5 \not\approx \theta'_1\theta_2\theta_3\theta_4\theta_5\,,$$

(this is a restatement of the relation $\alpha'\beta\gamma \not\approx \alpha'\beta'\gamma$) and since $\Delta_{ne'}\Delta_{f'e}\gamma'$ is shorter than γ (recall $\Delta_{e',e'-1} \not\approx \Lambda$) and cannot distinguish c and c', it follows that

$$\text{(d)} \quad \theta_1\theta_2\theta_3\theta_5 \approx \theta'_1\theta_2\theta_3\theta_5\,.$$

By independence and concatenation,

$$c_0 \xrightarrow{\theta_1\theta_2{}^i\theta_3} (s'_{e'}, (x_{e'} \cdots x_{f'+1})^i x_{f'} \cdots x_1)$$

and

$$c_0 \xrightarrow{\theta'_1\theta_2{}^i\theta_3} (s'_{e'}, (x_{e'} \cdots x_{f'+1})^i x_{f'} \cdots x_{e+1}\, x_f \cdots x_1)$$

and because θ_4 is null transparent, Theorem 2 implies

$$\text{(e)}\quad \theta_1\theta_2^{\,i}\theta_3\theta_5 \approx \theta_1\theta_2^{\,j}\theta_3\theta_5$$

and

$$\text{(f)}\quad \theta'_1\theta_2^{\,i}\theta_3\theta_5 \approx \theta'_1\theta_2^{\,j}\theta_3\theta_5\,,$$

for all i and j greater than some ℓ.

Relations (c), (d), (e), and (f) imply that one of the following must be true for all $i \geqq \ell$:

$$\begin{aligned}
&\text{(g)}\quad \theta_1\theta_2^{\,i}\theta_3\theta_5 \not\approx \theta_1\theta_2\theta_3\theta_5\,,\\
&\text{(h)}\quad \theta_1\theta_2^{\,i}\theta_3\theta_5 \not\approx \theta_1\theta_2\theta_3\theta_4\theta_5\,,\\
&\text{(i)}\quad \theta'_1\theta_2^{\,i}\theta_3\theta_5 \not\approx \theta'_1\theta_2\theta_3\theta_5\,,\\
&\text{(j)}\quad \theta'_1\theta_2^{\,i}\theta_3\theta_5 \not\approx \theta'_1\theta_2\theta_3\theta_4\theta_5\,.
\end{aligned}$$

If relation (g) holds, relations (a) and (g) satisfy Theorem 1 with $\alpha_1 = \theta_1\theta_2$, $\alpha_2 = \theta_2$, $\alpha_3 = \theta_3$, $\alpha_4 = \theta_4$, and $\alpha_5 = \theta_5$. If relation (h) holds, relation (a) implies

$$\theta_1\theta_2^{\,i+1}\theta_3\theta_4\theta_5 \not\approx \theta_1\theta_2\theta_3\theta_4\theta_5\,,$$

and Theorem 1 holds with $\alpha_1 = \theta_1\theta_2$, $\alpha_2 = \theta_2$, $\alpha_3 = \theta_3$, $\alpha_4 = \theta_4$, $\alpha_5 = \theta_4\theta_5$, and $\ell = \ell + 1$. Similarly, (b) and (i) or (b) and (j) also satisfy Theorem 1. In any case, Theorem 1 says that the set recognized is not regular, contrary to our assumption, and the theorem is proved. Q.E.D.

Corollary 5.1. *If the pushdown machine has only one state, M may be taken to be $t^2 + 4t + 1$.*

Proof. This is true by direct substitution into the expression for B.

Corollary 5.2. *If the pushdown machine has no ϵ-moves, then M may be given by q^3t^3+qt+1.*

Proof. This is obtained by using the bounds of corollaries 3.1 and 4.1.

Corollary 5.3. *The set L recognized by a pushdown machine is regular if and only if the intersection of L with every regular set of the form $\{\alpha_1\alpha_2^{\,i}\alpha_3\alpha_4^{\,i}\alpha_5\}$ is regular.*

Proof. In the proof of Theorem 5, we found such a set when L was non-regular.

Corollary 5.4. *A reduced finite state machine which recognizes the same set as a pushdown machine cannot have more than qt^M states if $t > 1$ or qM states if $t = 1$.*

Proof. The number of states cannot be larger than the number of configurations with tape word of length less than or equal to M.

This last corollary implies that the order of magnitude of the number of states is $t^{q^{q^q}}$ as stated in the introductory paragraph. Because the suitable ℓ-invisible segments can in fact be obtained constructively, it is possible to construct this machine without enumeration, but this is of little comfort in view of the orders of magnitude involved. If this bound cannot be improved significantly, then it would appear profitable in some cases to maintain a pushdown design for a recognizer even if a finite state design is possible. We can now state the main result:

THEOREM 6. *It is recursively decidable whether or not the set recognized by a given (deterministic) pushdown machine is regular.*

Proof. Enumerate all the finite state machines which do not have more states than the bound given in Corollary 5.4 and test each of these to see if it is equivalent to the pushdown machine. If one of these machines is equivalent to the pushdown machine, then the set is regular and otherwise it is not. A proof that the equivalence of a finite state machine and a pushdown machine is solvable may be found in Ginsburg and Greibach (1966). This problem reduces to the better-known emptiness problem by constructing the pushdown machine which recognizes the proper difference of the two sets in question and testing the resulting set to see if it is empty.

RECEIVED: March 6, 1967; revised: August 14, 1967

REFERENCES

GINSBURG, S. AND SPANIER, E. H. (1964), Bounded algol-like languages. *Trans. Am. Math. Soc. Vol.* **113**, No. 2, 333–368.

GINSBURG, S. AND GREIBACH, S. (1966), Deterministic context free languages. *Inform. Control* **9**, 620–648.