RAIRO
INFORMATIQUE THÉORIQUE

CELIA WRATHALL

Characterizations of the Dyck sets

RAIRO – Informatique théorique, tome 11, n° 1 (1977), p. 53-62.

<http://www.numdam.org/item?id=ITA_1977__11_1_53_0>

R.A.I.R.O. Informatique théorique/Theoretical Computer Science
(vol. 11, n° 1, 1977, p. 53 à 62)

CHARACTERIZATIONS OF THE DYCK SETS (*) (1)

by Celia WRATHALL (2)

Communicated by M. NIVAT

Abstract. – *Two representations of the Dyck sets, using language-theoretic operations, are presented.*

Two representations of Dyck sets in terms of simpler languages are given here. First, the Dyck set on k letters ($k \geqq 2$) is shown to be definable from the Dyck set on one letter by use of language-theoretic operations. Second, membership in the Dyck set on one letter is expressed in terms of the language $\{ 0^n 1^n : n \geqq 0 \}$. In both cases complementation is used, along with some of the AFL operations; use of complementation is necessary.

In Section 1, some basic definitions from formal language theory are reviewed, including the definition of the Dyck sets. In Section 2, the Dyck set on two letters is shown to be equal to a certain language that is defined from the Dyck set on one letter by application of the operations of inverse homomorphism, product, union and complementation (Theorem 2.3). The method of definition generalizes easily to the Dyck set on k letters for any $k \geqq 2$. Deterministic automata that accept the Dyck sets have been described elsewhere [9, 13]. The representation of the Dyck set on two letters that will be given here can be used to simplify the mode of operation of such automata. One result of these constructions is a representation of the context-free languages using two-way counter machines and length-preserving homomorphism. In Section 3 the Dyck set on one letter is considered. By making use of a known characterization of that set, it is shown that the Dyck set on one letter can be defined from $\{ 0^n 1^n : n \geqq 0 \}$ by application of the Boolean and AFL operations. The results are summarized in Theorem 3.2.

(*) Received : December 1975. Revised : June 1976.
(1) This research was supported in part by the National Science Foundation under Grants DCR-75-15945, and MCS-76-05744.

(2) Department of System Science, University of California, Los Angeles, California, USA and Computer Systems Laboratory, University of California, Santa Barbara, California, USA.

1. PRELIMINARIES

This section establishes the notation and terminology to be used in this paper.

If Σ is a finite set of symbols (an alphabet) then Σ^* denotes the free monoid over Σ, that is, the set of all strings of symbols in Σ. The identity of Σ^* is the empty string, e. The length of a string $x \in \Sigma^*$ is denoted $|x|$; $|e| = 0$. If $x \in \Sigma^*$ and $a \in \Sigma$ then $\# a(w)$ is the number of occurrences of the symbol a in w.

"Homomorphisms" are monoid homomorphisms $h : \Sigma^* \to \Gamma^*$, with Σ and Γ alphabets. If $h : \Sigma^* \to \Gamma^*$ is a homomorphism, the "inverse homomorphism" h^{-1} takes strings in Γ^* to subsets of Σ^* : for $x \in \Sigma^*$, $y \in \Gamma^*$, $x \in h^{-1}(y)$ if and only if $h(x) = y$. A homomorphism $h : \Sigma^* \to \Gamma^*$ is length-preserving if for all $a \in \Sigma$, $|h(a)| = 1$, and it is nonerasing if for all $a \in \Sigma$, $|h(a)| \geqq 1$. The homomorphism h performs k-limited erasing on $L \subseteq \Sigma^*$ if for all $x \in L$, $x = uyv$ with $h(y) = e$ implies that $|y| \leqq k$. The homomorphism h performs linear erasing on L if there is a constant m such that for all $x \in L$,

$$|x| \leqq m \,.\, \max \{ |h(x)|, 1 \}.$$

The product of two languages L_1 and L_2 is

$$L_1 L_2 = \{ x_1 x_2 : x_1 \in L_1 \text{ and } x_2 \in L_2 \}.$$

The Kleene * of L is $L^* = \bigcup_k L^k$ where $L^0 = \{ e \}$ and $L^{j+1} = L^j L$. The class of regular sets is the smallest class of languages containing the finite languages and closed under the operations of union, product and Kleene * An AFL [6] ("abstract family of languages") is a class of languages containing at least one nonempty language and closed under the AFL operations: nonerasing homomorphism, inverse homomorphism, union, product, Kleene * and intersection with regular sets. A full AFL is an AFL that is also closed under arbitrary homomorphism.

The Dyck sets on one and two letters are defined as follows. Let Σ_2 be the alphabet $\{ a_1, a_2, \bar{a}_1, \bar{a}_2 \}$ and let Σ_1 be the alphabet $\{ a_1, \bar{a}_1 \}$. Let $\sim$ be the binary relation on Σ_2^* defined by $x \sim y$ if and only if $x = ua_i\bar{a}_i v$ for some $u, v \in \Sigma_2^*$, $i = 1$ or 2, and $y = uv$. That is, $x \sim y$ if y results when some pair $a_1\bar{a}_1$ or $a_2\bar{a}_2$ is cancelled from x. Let $\overset{*}{\sim}$ denote the reflexive and transitive closure of $\sim$, so that $x \overset{*}{\sim} y$ if and only if for some $n \geqq 1$ and $z_1, \ldots, z_n \in \Sigma_2^*$, $x = z_1 \sim \ldots \sim z_n = y$. Then the Dyck set on two letters is $D_2 = \{ x \in \Sigma_2^* : x \overset{*}{\sim} e \}$ and the Dyck set on one letter is

$$D_1 = \{ x \in \Sigma_1^* : x \overset{*}{\sim} e \}.$$

Two properties of the Dyck sets are apparent:

(1) no string in D_1 begins with $\bar{a}_1$ or ends with a_1 ; and

(2) for $i = 1, 2$, for any x and y, if $x \sim y$ then $x \in D_i$ if and only if $y \in D_i$.

2. THE DYCK SET ON TWO LETTERS

In this section we show that membership in D_2 can be expressed in terms of membership in D_1.

DEFINITIONS: Let $h: \Sigma_2^* \to \Sigma_1^*$ be the homomorphism determined by defining

$$h(a_1) = h(a_2) = a_1 \qquad \text{and} \qquad h(\bar{a}_1) = h(\bar{a}_2) = \bar{a}_1.$$

Let

$$A = \Sigma_2^* a_1 (\Sigma_2^* - h^{-1}(D_1)\bar{a}_1 \Sigma_2^*) \cup \Sigma_2^* a_2 (\Sigma_2^* - h^{-1}(D_1)\bar{a}_2 \Sigma_2^*).$$

Note that for $x \in \Sigma_2^*$, $x \notin A$ if and only if whenever $x = ua_iv$ for $i = 1$ or 2 then $v \in h^{-1}(D_1)\bar{a}_i\Sigma_2^*$; that is, $x \notin A$ if and only if for every occurrence of a_i in x there is a "matching" occurrence of $\bar{a}_i$.

It is easy to see that $h(D_2) = D_1$, so that $D_2 \subseteq h^{-1}(D_1)$; moreover, the language A contains the strings that are in $h^{-1}(D_1)$ but not in D_2. To prove this, we first establish two properties of the language $h^{-1}(D_1) - A$.

LEMMA 2.1: *For any $x, y \in \Sigma_2^*$, if $x \sim y$, then $x \in h^{-1}(D_1) - A$ if and only if $y \in h^{-1}(D_1) - A$.*

Proof: Suppose $x \sim y$. Then by definition, $x = ua_j\bar{a}_jv$ for some u and v, and $y = uv$. Then

$$h(x) = h(u)a_1\bar{a}_1h(v) \sim h(u)h(v) = h(y),$$

so $h(x) \in D_1$ if and only if $h(y) \in D_1$. It remains to show that $x \in A$ if and only if $y \in A$.

If $x \in A$ then for some $a \in \{a_1, a_2\}$ and some u_1, v_1, $x = u_1av_1$ and $v_1 \notin h^{-1}(D_1)\bar{a}\Sigma_2^*$. Now if $u_1 = u$ then $a = a_j$ and $v_1 = \bar{a}_jv \in h^{-1}(D_1)\bar{a}_j\Sigma_2^*$; therefore $u_1 \neq u$. Two cases remain:

(i) $|u_1| < |u|$: Then $u = u_1au_2$ for some u_2 and $v_1 = u_2a_j\bar{a}_jv$. Now $h(u_2a_j) = h(u_2)a_1 \notin D_1$, so it is not hard to see that if $u_2v \in h^{-1}(D_1)\bar{a}\Sigma_2^*$ then also $v_1 \in h^{-1}(D_1)\bar{a}\Sigma_2^*$, a contradiction. Since $y = u_1a(u_2v)$, $y \in A$.

(ii) $|u_1| > |u|$: Then $u_1 = ua_j\bar{a}_ju_2$ for some u_2 so $y = uu_2av_1$. Since

$$v_1 \notin h^{-1}(D_1)\bar{a}\Sigma_2^*, \quad y \in A.$$

If $y \in A$ then $y = u_1av_1$ and $v_1 \notin h^{-1}(D_1)\bar{a}\Sigma_2^*$. Again there are two cases, $|u_1| < |u|$ and $|u_1| \geqq |u|$; using arguments similar to those above, it can be seen that $x \in A$. □

LEMMA 2.2: *For any $x \neq e$ in Σ_2^*, if $x \in h^{-1}(D_1) - A$ then $x = ua_i\bar{a}_iv$ for some $u, v \in \Sigma_2^*$ and $i = 1$ or 2.*

Proof: For $y \in \Sigma_2^*$, define

$$d(y) = \# a_1(y) + \# a_2(y) - \# \bar{a}_1(y) - \# \bar{a}_2(y)$$
$$= \# a_1(h(y)) - \# \bar{a}_1(h(y)).$$

Suppose $x = \sigma_1 \ldots \sigma_n$ is in $h^{-1}(D_1) - A$ with $n \geqq 1$, $\sigma_i \in \Sigma_2$ for $1 \leqq i \leqq n$. Let

$$m = \max \{ d(\sigma_1 \ldots \sigma_i) : 1 \leqq i \leqq n \}$$

and

$$k = \min \{ j : 1 \leqq j \leqq n, d(\sigma_1 \ldots \sigma_j) = m \};$$

that is, k is the leftmost position in x at which the maximum depth m is achieved. Since $h(x) \in D_1$ and every nonempty string in D_1 begins with a_1, $m > 0$. Let $u = \sigma_1 \ldots \sigma_{k-1}$ (that is, if $k = 1$, then $u = e$). By the choice of k, $d(\sigma_1 \ldots \sigma_k) \geqq d(u)$, so σ_k is either a_1 or a_2, say a_1. Since every nonempty string in D_1 ends with $\bar{a}_1$, $k < n$, so let $v = \sigma_{k+2} \ldots \sigma_n$. Then $x = u a_1 \sigma_{k+1} v$ and since $x \notin A$, $\sigma_{k+1} v \in h^{-1}(D_1)\bar{a}_1 \Sigma_2^*$. Since

$$d(\sigma_1 \ldots \sigma_k) = m \geqq d(\sigma_1 \ldots \sigma_{k+1}),$$

σ_{k+1} is a "barred" symbol, either $\bar{a}_1$ or $\bar{a}_2$. But no string in $h^{-1}(D_1)$ can begin with a "barred" symbol, so in fact $\sigma_{k+1} = \bar{a}_1$ and $x = u a_1 \bar{a}_1 v$. □

The set A was defined from $h^{-1}(D_1)$ by use of Boolean operations and product with regular sets. Thus the following theorem gives a definition of D_2 from D_1.

THEOREM 2.3: $D_2 = h^{-1}(D_1) - A$, *where*

$$A = \bigcup_{i=1,2} \Sigma_2^* a_i (\Sigma_2^* - h^{-1}(D_1)\bar{a}_i \Sigma_2^*).$$

Proof: The proof is by induction on $|x|$ for $x \in \Sigma_2^*$. For the basis step, $|x| = 0$, note that $e \in D_2$, $e \in h^{-1}(D_1)$ but $e \notin A$. Suppose for some $k > 0$, for all $y \in \Sigma_2^*$, $|y| < k$ implies $y \in D_2$ if and only if $y \in h^{-1}(D_1) - A$, and consider any string x with $|x| = k$. If $x \in D_2$ then since $|x| > 0$, there is a string y with $|y| = |x| - 2$ such that $x \sim y \overset{*}{\sim} e$. Since then $y \in D_2$ and $|y| < k$, $y \in h^{-1}(D_1) - A$. From Lemma 2.1, also $x \in h^{-1}(D_1) - A$. Conversely, if $x \in h^{-1}(D_1) - A$ then since $|x| > 0$, from Lemma 2.2 there is a string y such that $x \sim y$ and $|y| = |x| - 2$. From Lemma 2.1, $y \in h^{-1}(D_1) - A$ so $y \in D_2$ and therefore $x \in D_2$. □

By making the appropriate extensions in the definitions of the homomorphism h and the language A, one obtains a representation in the form of Theorem 2.3 for the Dyck set on k letters, $k \geqq 2$. The operations used above to define D_2 from D_1 were inverse homomorphism, union, product with regular sets and complementation. If complementation were removed, then the remaining operations, even with the addition of the other AFL operations and arbitrary homomorphism, would not be sufficient to yield D_2

from D_1. This follows from the fact that the full AFL generated by D_1 (i. e., the one-counter languages) is properly contained in the context-free languages [8]. From the Chomsky-Schützenberger Theorem, any (full) AFL containing D_2 must contain all the context-free languages, so the full AFL generated by D_1 cannot contain D_2. The addition of intersection to the AFL operations is also not sufficient to yield D_2 from D_1, since (using [5]) the intersection-closed AFL generated by D_1 does not contain the context-free language $\{ wcw^r : w \in \{ a, b \}^* \}$. On the other hand, the intersection-closed *full* AFL generated by D_1 does contain D_2, since it is the class of all recursively enumerable sets [12].

Theorem 2.3 can be rephrased as follows: a string $x \in \Sigma_2^*$ is in D_2 if and only if

(1) $h(x) \in D_1$; and

(2) whenever $x = ua_iv$ $(i = 1, 2)$, there is a string $w \in h^{-1}(D_1)$ such that $w\bar{a}_i$ is a prefix (i. e., initial substring) of v.

Descriptions of D_2 similar to this one have been the basis for constructing automata that accept the Dyck sets.

COROLLARY 2.4:

(i) [13] *D_2 can be accepted by a deterministic Turing machine that operates in* $\log(n)$ *space.*

(ii) [9] *D_2 can be accepted by a deterministic two-way one-counter automaton.* □

The automaton described in [13] is a device with two-way (read-only) input and has for storage counters which are bounded by the length of the input. so there is a $\log(n)$ tape-bounded Turing machine that accepts the same set. The counter of the device given in [9] is also bounded by the length of the input. The automaton in [9] essentially operates by checking condition (2) above for each symbol a_i $(i = 1,2)$ in the input $x = ua_iv$, using a counter to determine which initial substrings of v are elements of $h^{-1}(D_1)$. It also checks another condition, that every symbol $\bar{a}_i$ in x has a "matching" symbol a_i to its left; that is,

(3) whenever $x = u\bar{a}_iv$ $(i = 1,2)$ there exists $w \in h^{-1}(D_1)$ such that a_iw is a suffix of u.

It is not hard to see, however, that if conditions (1) and (2) are satisfied by x then x also satisfies (3).

Since the class of languages DSPACE $(\log(n))$ accepted by deterministic Turing machines in $\log(n)$ space is closed under inverse homomorphism and intersection (with regular sets), from Corollary 2.4 (i) and a result in [2], we see that any context-free language L can be represented as $L = h_L(L')$ where $L' \in$ DSPACE $(\log(n))$ and h_L is a length-preserving homomorphism.

Further, using [3], any language accepted by a nondeterministic multitape Turing machine in linear time can be so represented, as the image under a length-preserving homomorphism of a language in DSPACE ($\log(n)$). It is not known whether DSPACE ($\log(n)$) is closed under application of length-preserving homomorphisms. From Corollary 2.4 (ii), statements similar to these hold for the class of languages accepted by deterministic automata with two-way input and one counter.

The space bound in Corollary 2.4 is also a lower bound for recognition of the Dyck sets: from [1], any nondeterminisitic off-line Turing machine that accepts D_2 must use at least $\log(n)$ space on an infinite set of input strings.

Other definitions of D_2 from simpler languages have been given in order to demonstrate that predicates of a certain simple form exist for membership in D_2. The characterization of the Dyck sets suggested in [15] can be restated using language-theoretic operations. Let $f_1 : \Sigma_2^* \to \Sigma_1^*$ and $f_2 : \Sigma_2^* \to \Sigma_1^*$ be the homomorphisms (similar to the homomorphism h of Theorem 2.3) determined by defining for $i = 1,2$,

$$f_i(a_i) = a_1 \quad , \quad f_i(\overline{a}_i) = \overline{a}_1 \qquad \text{and} \qquad f_i(a_j) = f_i(\overline{a}_j) = e$$

for $j \neq i$ ($j = 1,2$). Let $J = f_1^{-1}(D_1) \cap f_2^{-1}(D_1)$, so that J is a "shuffle" of two copies of D_1. Define a language K by $K = \Sigma_2^* a_1 J \overline{a}_2 \Sigma_2^* \cup \Sigma_2^* a_2 J \overline{a}_1 \Sigma_2^*$. Then $D_2 = J - K$. The language J is properly contained in $h^{-1}(D_1)$, and K is properly contained in A. From this representation, a deterministic two-way automaton can be easily constructed to recognize D_2 using two (linear-bounded) counters.

3. THE DYCK SET ON ONE LETTER

In order to define D_1 from $L_0 = \{ 0^n 1^n : n \geqq 0 \}$, we first express D_1 in terms of $L_1 = \{ w \in \{ 0, 1 \}^* : \# 0(w) = \# 1(w) \}$ and then express L_1 in terms of L_0. The following theorem restates a well-known characterization of the Dyck set on one letter.

THEOREM 3.1 : *There exist homomorphisms h_1, h_2, h_3, with h_1 and h_2 length-preserving, and a regular set R such that*

$$D_1 = h_1(L_1 - h_2(R \cap h_3^{-1}(L_1)))$$

where $L_1 = \{ w \in \{ 0, 1 \}^* : \# 0(w) = \# 1(w) \}$.

Proof : Let $h_1 : \{ 0, 1 \}^* \to \Sigma_1^*$ be the homomorphism determined by defining $h_1(0) = a_1$, $h_1(1) = \overline{a}_1$. Let $h_2 : \{ 0, 1, c, d \}^* \to \{ 0, 1 \}^*$ and $h_3 : \{ 0, 1, c, d \}^* \to \{ 0, 1 \}^*$ be the homomorphisms determined by defining

$$h_2(0) = h_2(c) = h_3(0) = 0 \quad , \quad h_2(1) = h_2(d) = h_3(1) = 1$$

and $h_3(c) = h_3(d) = e$.

That is, h_2 changes c to 0 and d to 1 and h_3 erases occurrences of c and d; both h_2 and h_3 leave 0 and 1 fixed. Let R be the regular set denoted by the expression $(0 + 1)^* d(0 + 1 + d)^* (c + d)^*$. Then for $x \in \{ 0, 1 \}^*$,

$$x \in h_2(R \cap h_3^{-1}(L_1))$$

if and only if for some prefix y of x, $\# 0(y) < \# 1(y)$. Hence

$$x \in h_1(L_1 - h_2(R \cap h_3^{-1}(L_1)))$$

if and only if (i) $\# a_1(x) = \# \bar{a}_1(x)$ and (ii) for every prefix y of x. $\# a_1(y) \geq \# \bar{a}_1(y)$. These two conditions are known to characterize the strings in D_1, and the theorem follows. □

It will be shown that L_1 can be defined from L_0 and regular sets by use of inverse homomorphism, length-preserving homomorphism, union and intersection. The operation of complementation need not be used, because L_1 can be accepted in linear time by a deterministic automaton with one-way input and two counters, that operates in such a way that each of the counters makes only one turn during any computation. By "splitting" the computations of this automaton as in [4], two languages L_1' and L_1'' can be found such that L_1 is the image under a linear-erasing homomorphism of $L_1' \cap L_1''$ and L_1' and L_1'' are one-turn one-counter languages (that is, are accepted by automata with one counter that make only one turn during any computation). Since the AFL generated by L_0 contains the one-turn one-counter languages [8], L_1 can therefore be defined from L_0 using the AFL operations together with intersection and application of a linear-erasing homomorphism. The algebraic definition of L_1 from L_0 (which reduces the linear-erasing homomorphism to a length-preserving homomorphism) is based on these ideas.

Let

$$C_1 = \{ e \} \cup \{ ucvcw : u, v, w \in \{ 0, 1 \}^*, \# 0(u) = \# 0(vw), \# 1(uv) = \# 1(w) \quad \text{and} \quad \# 0(u) = \# 1(w) \geq 1 \}.$$

Note that if $x = ucvcw$ is in C_1 then $\# 0(x) = \# 1(x)$, $\# 0(x)$ is even, and the two occurrences of c in x mark the positions in x where half the 0's and half the 1's in x have occurred. Similarly, let

$$C_2 = \{ ucvcw : u, v, w \in \{ 0, 1 \}^*, \# 0(u) + 1 = \# 0(vw), \# 1(uv) + 1 = \# 1(w) \quad \text{and} \quad \# 0(u) + 1 = \# 1(w) \geq 1 \}.$$

Let $g_1 : \{ 0, 1, c \}^* \to \{ 0, 1, c \}^*$ be the homomorphism that interchanges 0's 1's : $g_1(0) = 1$, $g_1(1) = 0$ and $g_1(c) = c$. Let $C_3 = C_1 \cup g_1(C_1) \cup C_2 \cup g_1(C_2)$. Let $g_2 : \{ 0, 1, c \}^* \to \{ 0, 1 \}^*$ be the homomorphism that erases c: $g_2(0) = 0$, $g_2(1) = 1$ and $g_2(c) = e$. Then $L_1 = g_2(C_3)$. Since any word in C_3 has at most two occurrences of the symbol c, g_2 is 2-limited on C_3. The effect of a k-limited homomorphism can be achieved by use of length-

preserving homomorphism, inverse homomorphism and intersection with a regular set [7, p. 44], so it suffices to show that C_1 and C_2 can be formed from L_0.

C_1 is the intersection of three one-turn one-counter languages, each of which checks one of the conditions on the number of symbols in a word. Let

$$C_4 = \{ ucvcw : \#0(u) = \#0(vw) \geqq 1 \},$$
$$C_5 = \{ ucvcw : \#1(uv) = \#1(w) \geqq 1 \}$$

and

$$C_6 = \{ ucvcw : \#0(u) = \#1(w) \geqq 1 \};$$

then $C_1 = \{ e \} \cup (C_4 \cap C_5 \cap C_6)$. It is not hard to see that C_4, C_5 and C_6 are inverse a-transducer mappings of L_0; hence they can be defined from L_0 by use of length-preserving homomorphism, inverse homomorphism and intersection with regular sets [6]. We show how C_4 can be defined from L_0 and regular sets; the definition for C_5 is essentially the same, and that for C_6 is simpler. Define four homomorphisms as follows:

$r_1 : \{ 0, 1, \mathfrak{S} \}^* \to \{ 0, 1 \}^*$, $\quad r_1(0) = 0,\ r_1(1) = 1,\ r_1(\mathfrak{S}) = e,$

$r_2 : \{ 0, 1, \mathfrak{S} \}^* \to \{ 0, \mathfrak{S} \}^*$, $\quad r_2(0) = r_2(1) = 0,\ r_2(\mathfrak{S}) = \mathfrak{S},$

$r_3 : \{ 0, 1, \mathfrak{c}, \mathfrak{S} \}^* \to \{ 0, 1, \mathfrak{S} \}^*$, $\quad r_3(0) = 0,\ r_3(1) = r_3(\mathfrak{c}) = e,\ r_3(\mathfrak{S}) = \mathfrak{S},$

$r_4 : \{ 0, 1, \mathfrak{c}, \mathfrak{S} \}^* \to \{ 0, 1, \mathfrak{c} \}^*$, $\quad r_4(0) = 0,\ r_4(1) = 1,\ r_4(\mathfrak{c}) = r_4(\mathfrak{S}) = \mathfrak{c}.$

Let R_1 and R_2 be the regular sets:

$$R_1 = \{ 0^m \mathfrak{S} 1^n : m, n \geqq 1 \},$$
$$R_2 = \{ 0, 1 \}^* \{ \mathfrak{S} \} \{ 0, 1 \}^* \{ \mathfrak{c} \} \{ 0, 1 \}^*.$$

Then

$$r_1^{-1}(L_0) \cap R_1 = \{ 0^n \mathfrak{S} 1^n : n \geqq 1 \};$$
$$r_2(r_1^{-1}(L_0) \cap R_1) = \{ 0^n \mathfrak{S} 0^n : n \geqq 1 \};$$
$$r_3^{-1}(r_2(r_1^{-1}(L_0) \cap R_1)) \cap R_2 = \{ u \mathfrak{S} v \mathfrak{c} w : u, v, w \in \{ 0, 1 \}^*, \#0(u) = \#0(vw) \geqq 1 \};$$

and

$$r_4(r_3^{-1}(r_2(r_1^{-1}(L_0) \cap R_1)) \cap R_2) = C_4.$$

The demonstration that C_2 can be formed from L_0 is similar.

The preceding discussion is combined with Theorems 2.3 and 3.1 in the following result.

THEOREM 3.2: *If $\mathscr{C}$ is a class of languages containing $\{ 0^n 1^n : n \geqq 0 \}$ and closed under union, difference with regular sets, inverse homomorphism and length-preserving homomorphism, then every Dyck set is in $\mathscr{C}$ and hence the family of context-free languages is properly contained in $\mathscr{C}$.*

Proof: "Difference with regular sets" is the operation taking a language L to R-L where R is any regular set. If $\mathscr{C}$ satisfies the conditions in the statement

then it is easily seen that $\mathscr{C}$ is also closed under intersection and difference and that $\mathscr{C}$ contains every regular set. $\mathscr{C}$ can also be shown to be closed under product, by a straightforward construction using length-preserving homomorphism, inverse homomorphism and intersection.

Now since L_0 and every regular set are in $\mathscr{C}$ and $\mathscr{C}$ is closed under length-preserving homomorphism, inverse homomorphism, union and intersection, $L_1 \in \mathscr{C}$. From Theorem 3.1, then, since $\mathscr{C}$ is also closed under difference, $D_1 \in \mathscr{C}$, and from Theorem 2.3, $D_2 \in \mathscr{C}$. To see the second part of the statement, suppose L is a context-free language. Then there exist homomorphisms h_1, h_2 and a regular set R such that $L = h_1(R \cap h_2^{-1}(D_2))$ and h_1 is length-preserving. This stronger form of the Chomsky-Schützenberger Theorem can be shown by considering a Greibach Normal Form grammar for L, as in [2]. Since $D_2 \in \mathscr{C}$ and $\mathscr{C}$ is closed under the operations used to form L from D_2, $L \in \mathscr{C}$. Proper containment follows since the class of context-free languages is not closed under intersection. □

As in Theorems 2.3 and 3.1, the operation of difference or complementation cannot be deleted in the statement above, since the closure of L_0 under the AFL operations and intersection does not contain the Dyck sets.

For a language L, let $\mathscr{C}(L)$ denote the smallest class of languages containing L and closed under the operations in Theorem 3.2: union, difference with regular sets, inverse homomorphism and length-preserving homomorphism. The question then arises: what context-free languages L have the property that all context-free languages are in $\mathscr{C}(L)$ (or, equivalently, that $L_0 \in \mathscr{C}(L)$)? It is not hard to show that $\mathscr{C}(L_0)$ is closed under all Boolean and AFL operations, as well as under application of linear-erasing homomorphisms. The regular sets also have these closure properties. Is there a Boolean-closed AFL lying strictly between the regular sets and $\mathscr{C}(L_0)$? The class $\mathscr{C}(L_0)$ is of independent interest, in that it is the class of languages associated with the rudimentary attributes [14].

The referee has pointed out that the representation of the Dyck sets in [10] is false; a counterexample is the string $a_1\bar{a}_1a_2a_1\bar{a}_2\bar{a}_1a_2\bar{a}_2$, which satisfies the predicate in [10] but is not in D_2.

The referee has also noted that a simpler definition of L_1 from L_0 is possible [11]: L_1 is the shuffle of L_0 with the language $L_0^R = \{ 1^n0^n : n \geqq 0 \}$.

REFERENCES

1. H. ALT and K. MEHLHORN, *Lower Bounds for the Space Complexity of Context-Free Recognition*, unpublished manuscript.
2. R. BOOK, *On the Chomsky-Schützenberger Theorem*, Technical Report, Department of Computer Science, Yale University, New Haven, Conn., 1975.

3. R. BOOK and S. GREIBACH, *Quasirealtime Languages*, Math. Systems Theory, Vol. 4, 1970, pp. 97-111.
4. R. BOOK, M. NIVAT and M. PATERSON, *Reversal-Bounded Acceptors and Intersections of Linear Languages*, SIAM J. Comput., Vol. 3, 1974, pp. 283-295.
5. P. FISCHER, A. MEYER and A. ROSENBERG, *Counter Machines and Counter Languages*, Math. Systems Theory, Vol. 2, 1968, pp. 265-283.
6. S. GINSBURG and S. GREIBACH, *Abstract Families of Languages*, Memoir 87, American Mathematical Society, Providence, R.I., 1969, pp. 1-32.
7. S. GINSBURG, S. GREIBACH and J. HOPCROFT, *Pre-AFL*, Memoir 87, American Mathematical Society, Providence, R.I., 1969, pp. 41-51.
8. S. GREIBACH, *An Infinite Hierarchy of Context-Free Languages*, J. Assoc. Computing Machinery, Vol. 16, 1969, pp. 91-106.
9. G. HOTZ and J. MESSERSCHMIDT, *Dyck-Sprachen sind in Bandkomplexität log n analysierbar*, Technical Report, Universität des Saarlandes, 1975.
10. N. JONES, *Context-Free Languages and Rudimentary Attributes*, Math. Systems Theory, Vol. 3, 1969, pp. 102-109.
11. M. LATTEUX, *Cônes rationnels commutativement clos*, This Journal.
12. M. MINSKY, *Recursive Unsolvability of Post's Problem of Tag and other Topics in the Theory of Turing Machines*, Annals of Math., Vol. 74, 1961, pp. 437-455.
13. R. RITCHIE and F. SPRINGSTEEL, *Language Recognition by Marking Automata*, Inf. and Control, Vol. 20, 1972, pp. 313-330.
14. C. WRATHALL, *Subrecursive Predicates and Automata*, Research Report 56, Department of Computer Science, Yale University, New Haven, Conn., 1975.
15. Y. YU, *Rudimentary Relations and Formal Languages*, Ph. D. dissertation, University of California, Berkeley, Calif., 1970.