Reprinted from IEEE TRANSACTIONS
ON *COMPUTERS*
Volume C-20, Number 4, April, 1971
pp. 456–459

General Shift-Register Sequences of Arbitrary Cycle Length

ALVY RAY SMITH, III, MEMBER, IEEE

Abstract—An r-ary shift-register sequence is desired that has arbitrary cycle length $L \leq r^k$ for arbitrary r and k, where k is the number of stages (degree) of the shift register. The existence of such sequences is established for "almost all" cycle lengths L. Furthermore, existence of such sequences which are "zero free" for almost all cycle lengths L is proved.

Index Terms—Coding theory, maximum-length cycle, shift-register sequence, zero free.

Manuscript received January 12, 1970; revised October 29, 1970.
The author is with the Department of Electrical Engineering, New York University, Bronx, N. Y. 10453.

A coding problem arising in the theory of iterative arrays of finite-state machines (namely, given such an array, does there exist an equivalent array of binary machines [1]) was found to be closely related to the following problem stated in the terminology of shift-register theory: Do there exist "zero-free" shift-register sequences of arbitrary cycle lengths $L \leq r^k$ for arbitrary r and k, where k is the number of stages (degree) of a shift register and r is the number of states per stage? A *zero-free* sequence does not contain the length-k subsequence 0^k where the r states per stage are given, without loss of generality, the convenient names $0, 1, 2, \cdots, r-1$. The question has been answered affirmatively for the case $r=2$ by Golomb [2, p. 192] and Yoeli [3]. We now proceed to answer the generalized question affirmatively for almost all lengths L. The qualifier "almost all" will be made precise in the theorems below and will be conjectured, in fact, to be unnecessary.

Lemma 1: (Algorithm for a maximum-length shift-register sequence.) A maximum-length ($L=r^k$) r-ary shift-register sequence (SRS) of degree k is formed from the leftmost digit of each k-digit number in the list constructed in accordance with steps 1 and 2. The leftmost digits are taken in the order generated.

1) Begin a list of length-k strings (i.e., k-digit numbers in positional notation) with the string $0^{k-1}(r-1)$ as the "initial word."

2) Each succeeding number in the list will have as its first $k-1$ digits (its *prefix*) the last $k-1$ digits (the *suffix*) of the preceding number. The kth digit is chosen to be the largest integer ($\leq r-1$) such that the string so formed has not previously appeared in the list.

Example 1: For $r=k=3$, the algorithm of Lemma 1 generates the following ordered table of length-3 strings.

1	002	10	211	19	200
2	022	11	112	20	001
3	222	12	121	21	011
4	221	13	210	22	111
5	212	14	102	23	110
6	122	15	020	24	101
7	220	16	201	25	010
8	202	17	012	26	100
9	021	18	120	27	000

Notice that the string formed from the leftmost digits of this list is the maximum-length SRS 002221220211210201200 111010.

Proof: We review a proof adapted directly from [2, p.133] for the convenience of the reader. By definition of the algorithm, if a string of length k appears in the list, then it appears only once. Let $I_1 I_2 \cdots I_n = 0^{k-1}(r-1)$ be the initial word. Then we show by induction on m that $b_m \cdots b_1 I_1 \cdots I_{k-m}$ is in the list for all choices of $b_1, \cdots, b_m \in \{0, 1, \cdots, r-1\}$. For $m=k$, this says that all k-digit strings appear in the list. If the suffix $a_2 \cdots a_{k-1}$ of the last string in the list is 0^{k-1}, then clearly the leftmost digits form a cycle of length r^k. First we show that $a_2 \cdots a_{k-1}$ is indeed 0^{k-1}.

Since $a_2 \cdots a_{k-1}$ is the suffix of the last string, it must be the case that all r strings of the form $a_2 \cdots a_{k-1}x$ have already been listed. This would mean that $a_2 \cdots a_{k-1}$ occurs as a suffix $r+1$ times in the list, which is impossible. Hence one of the occurrences of the prefix $a_2 \cdots, a_{k-1}$ must not be the suffix of the preceding string in the list, which can only be the case in the initial word. Now we induct on m.

Assume $m=1$. By the argument just given, the last suffix is $I_1 \cdots I_{k-1}$ and it occurs as a suffix r times in the list. Since each occurrence is distinct, $b_1 I_1 \cdots I_{k-1}$ must appear for each possible choice of b_1.

Assume $1 < m \leq k$ and that $b_{m-1} \cdots b_1 I_1 \cdots I_{k-m+1}$ is in the list for every possible $b_{m-1} \cdots b_1$. If the rightmost j digits of $b_{m-1} \cdots b_1 I_1 \cdots I_{k-m}$ equal the leftmost j digits of the initial word for $j > k-m$ and some choice of $b_{m-1} \cdots b_1$, then by the inductive hypothesis $b_m \cdots b_1 I_1 \cdots I_{k-m}$ is in the list for each possible b_m. If $j \leq k-m$, then $b_{m-1} \cdots b_1 I_1 \cdots I_{k-m} \neq I_1 \cdots I_{k-1}$. That is, $b_{m-1} \cdots b_1 I_1 \cdots I_{k-m}$ is not the prefix of the initial word. Hence every time it appears as a prefix in the list, it must be the suffix of the preceding word. But it must appear all r possible times since, by the inductive hypothesis, it occurs followed by $I_{k-m+1}=0$. Q.E.D.

Consider a list λ such as that generated in the algorithm above. We shall call any list formed from successive elements of λ a *sublist* of λ. If the last element of a sublist of λ is also the last element of λ, then the sublist is a *terminal sublist* of λ.

Lemma 2: A number comprised of digits each of which are equal to or less than $r_0 - 1$ cannot precede $r_0 0^{k-1}$ in the list generated by the algorithm of Lemma 1 for $r > r_0$.

Proof: Let W_{k-h} be a length-$(k-h)$ r-ary word with $1 \leq h < k$. Then when $W_{k-h}0^h$ appears in the list of the Lemma 1 algorithm, the r^h possible numbers which end with W_{k-h} must already have occurred in the list. This is seen readily by induction on h; it is true by construction for $h=1$. Also by construction, the addition of $W_{k-(n+1)}0^{n+1}$ to the list implies the numbers $W_{k-(n+1)}0^n 1$, $W_{k-(n+1)}0^n 2, \cdots$, $W_{k-(n+1)}0^n(r-1)$ have been included in the list previously. But this implies in turn that the prefix $W_{k-(n+1)}0^n$ has occurred as a suffix all r possible times. By the inductive hypothesis applicable here, this means that all r^n possible words ending with each $r' W_{k-(n+1)}$, $0 \leq r' < r$, have occurred in the list, or the r^{n+1} possible words ending in $W_{k-(n+1)}$ have occurred.

In particular, $W_1 0^{k-1} = r_0 0^{k-1}$ must succeed all the r^{k-1} possible numbers ending in r_0. The construction process then allows no introduction of digits larger than $r_0 - 1$ to succeeding numbers in the list. Q.E.D.

Corollary 1: The list generated by the algorithm of Lemma 1 for $r=r_0$ and $k=k_0$ is a terminal sublist of the list generated by the same algorithm for any $r > r_0$ and $k=k_0$.

Proof: By Lemma 2, the number succeeding $r_0 0^{k-1}$ must be $0^{k-1}(r_0 - 1)$. But this is the initial word for the algorithm with $r=r_0$. Q.E.D.

Example 2: For $r=2$ and $k=3$, the algorithm of Lemma 1 generates the SRS 00111010 from the following ordered table.

1	001	5	101
2	011	6	010
3	111	7	100
4	110	8	000

Notice that this list is a terminal sublist of the list generated in Example 1.

Lemma 3: There exist r-ary shift-register sequences of degree k for all but n cycle lengths L, $1 \leq L \leq r^k$, where $n = 2^{k-1} - (k-1)$ if all the sequences are zero free, but $n = 2^{k-2} - (k-1)$ in the general case. In particular, there exist r-ary zero-free shift-register sequences for all cycle lengths L such that $1 \leq L \leq r^k - 2^k + 2k - 1$.

Proof: The proof proceeds by construction. For L such that $1 \leq L \leq k$, the SRS is simply $0^{L-1}1$. These are clearly zero-free sequences. For all larger $L \leq r^k$, consider the following procedure.

The first L digits of the length-r^k sequence generated by the algorithm of Lemma 1 form a length-L zero-free SRS if the Lth digit is not 0. We need only check the $k-1$ length-k numbers "around the ends" of the length-L number so formed, i.e., the numbers beginning with the $(L-k+2)$th, $(L-k+3)$th, $\cdots$, Lth digits, respectively. But by the construction process, all these numbers end in zeros and hence cannot have previously occurred in the list generated by Lemma 1.

If the Lth digit is a zero and the $(L-1)$th digit is not, then the first L digits form a length-L SRS which is not zero free.

If the Lth digit is a zero and so are the $(L-1)$th, $(L-2)$th, $\cdots$, $(L-i)$th digits, $1 \leq i \leq (k-2)$, then the first L digits do not form a SRS because the string 0^k occurs $i+1$ times. In this case the following procedure yields a length-L zero-free SRS for $k < L \leq r^k - 2^k + 2k - 1$. Append string $0^{k-1}1^j$ ($k-1$ zeros followed by j ones) to the left of the length-L number generated by Lemma 1 and delete from it the rightmost $(k-1)+j$ digits, $1 \leq j \leq k$. For at least one value of j, the Lth digit of the length-L number so formed must be other than 0 or 1. This is because, from Corollary 1, all length-k binary strings must be concentrated in the final 2^k digits of the length-r^k number generated by Lemma 1. It is also Corollary 1 which ensures no ambiguity in the addition of the binary string $0^{k-1}1^j$.

The number of cases not covered by the algorithms above are as follows. 1) In the zero-free case, the number n of lengths L not covered by the above is the number of zeros in the last $r^k - 2^k + 2k - 1$ digits of the maximum-length SRS, i.e., in the "binary section," guaranteed by Lemma 2, minus its leading $k-1$ zeros. Thus $n = 2^{k-1} - (k-1)$. 2) In the general case, n is as in 1 but reduced by the number n' of cases in which a 1 immediately precedes a 0 in the binary section. It is simple to see that n' is half the number of zeros, or $n' = 2^{k-2}$. Thus here $n = 2^{k-1} - 2^{k-2} - (k-1) = 2^{k-2} - (k-1)$. Q.E.D.

Corollary 2: There exist r-ary zero-free shift-register sequences of degree k for all cycle lengths L, $1 \leq L < r^k$, if $k \leq r+1$.

Proof: The corollary is true if $L \leq r^k - 2^k + 2k - 1$; so assume $L > r^k - 2^k + 2k - 1$. Mark off the left most L digits of the maximum-length SRS generated by the algorithm of Lemma 1, as in the proof of the theorem. The rightmost digit must fall in the binary section minus its leading $k-1$ zeros and k ones. If the digit is a 1, then the theorem gives the desired SRS; hence assume the digit is a 0. The nearest 1 on its right in the maximum-length SRS must be at most $k-2$ positions removed. That is, there exists immediately to the right of the length-L string already formed the string 0^i1, $0 \leq i \leq k-3$. Append this string and delete the rightmost digit of $i+1$ substrings of the form s^k. $1 \leq s \leq r-1$. Clearly these deletions do not alter the SRS property, and the string so formed is the desired length-L SRS. We have assumed $i+1$ distinct substrings s^k exist; hence $k-2 \leq r-1$. Q.E.D.

Notice that the technique employed in the proof of Corollary 2 can be used to reduce n in Lemma 3. The number n of lengths not covered now becomes, in the zero-free case, the number of zeros separated from the next 1 on the right by $r-2$ or more zeros. Thus n is the number of k-tuples ending with $0^j0^{r-1}1$, $1 \leq j \leq k-r$, less those beginning with all zeros (which correspond to the leading zeros of the binary section). This is given by

$$n = \sum_{i=0}^{k-r-1} 2^i - (k-r) + 1 = 2^{k-r} - (k-r)$$

where an extra one for the maximum-length sequence is included in the summation. For the general case, n is reduced, as in the proof of Lemma 3, by the number n' of places an excluded zero is preceded immediately by a 1. Thus n' is the number of k-tuples ending with 10^j0^{r-1} (plus one for the excluded subsequence 0^k) or

$$n' = \sum_{i=0}^{k-r-2} 2^i + 1 = 2^{k-r-1}.$$

Thus $n = 2^{k-r-1} - (k-r)$. This argument and the preceding results can be summarized in the following statement.

Theorem: There exist r-ary shift-register sequences of degree k for all cycle lengths L, $1 \leq L \leq r^k - 2^k + 2k - 1$. Furthermore, there exists such sequences for all but n lengths where $n = 2^{k-r} - (k-r)$ in the zero-free case and $n = 2^{k-r-1} - (k-r)$ in the general case. (Take $n = 0$ if the value of its expression is negative.)

Example 3: The algorithms of the theorem generate the following SRS table (except for $L = 25$ which is covered by Corollary 2) for $r = k = 3$.

$L=1$	1	$L=14$	00222122021121
2	01	15	001110022212202
3	001	16	0022212202112102
4	0022	17	00110022212202112
5	00222	18	002221220211210201
6	002221	19	0022212202112102012
7	0022212	20	00110022212202112102
8	00222122	21	001110022212202112102
9	001100222	22	0022212202112102012001
10	0022212202	23	00222122021121020120011
11	00222122021	24	002221220211210201200111
12	002221220211	25	0022212202112102012001101
13	0022212202112	26	00222122021121020120011101

All these numbers are zero free. Notice that case $L = 25$ is not covered by the theorem. A zero-free number has been obtained however by simply eliminating the string 111 from the list of strings generating the length-26 SRS.

Since the proofs above utilize only one of the many (see [2]) algorithms for generating maximum-length SRSs, and from experience with small r and small k (as in Example 3 above), the following is proposed.

Conjecture: There exist zero-free shift-register sequences of degree k for lengths L, $1 \leq L \leq r^k$, $k \geq 0$, for all r.

Notice that the number of lengths not covered by the algorithms above is a function of both k and r. Hence the smallest pair (r, k) not completely treated is (3, 5) for which there are possibly 243 sequences of distinct lengths. Sequences for all 243 lengths exist if there is no zero-free requirement. Sequences for all but two of the lengths exist in the zero-free case, and one of these is, of course, the maximum-length sequence with $L=243$. The other must be of length $L>220$.

Notice also that for $k \geq 5$, the binary section generated by the Lemma 1 algorithm always begins $0^{k-1}1^k01^{k-2}0^21^{k-3}0101^{k-3}0^3$. Hence for $k \geq 5$ the bound on L for which zero-free SRS sequences of the desired variety are guaranteed to exist can be improved to $L \leq r^k - 2^k + 5k - 3$. Thus, in the (3, 5) example above, the sequence not covered must be of length $L>233$.

References

[1] A. R. Smith, III, "Cellular automata theory," Digital Systems Laboratory, Stanford University, Stanford, Calif. Tech. Rept. 2, 1969.

[2] S. W. Golomb, *Shift-Register Sequences.* San Francisco: Holden-Day, 1967.

[3] M. Yoeli, "Binary ring sequences," *Amer. Math. Mon.*, vol. 69, Nov. 1962, pp. 852–855.