Roots of Star Events

J. A. BRZOZOWSKI

University of Ottawa, Ottawa, Canada*

ABSTRACT. A regular event W is a star event if there exists another event V such that $W = V^*$. In that case, V is called a root of W. It is shown that every star event has a unique minimum root, which is contained in every other root. An algorithm for finding the minimum root of a regular event is presented, and the root is shown to be regular. The results have applications to languages, codes, canonical forms for regular expressions, simplification of expressions, decomposition of sequential machines, and semigroup theory.

1. *Algebraic Background*

We begin by reviewing some properties of semigroups [1, 2], and by proving some results about general semigroups to be used later in more specific applications.

A *semigroup* is a nonempty set S together with a binary associative operation called multiplication and denoted by juxtaposition. Thus $z = xy$ denotes the product of x and y. We are particularly interested in certain subsets of a semigroup [1]. Let G and H be subsets of semigroup S; then so are $G \cup H$, $G \cap H$, and $G - H$ (union, intersection, and relative complement, respectively). Furthermore, for G, $H \subseteq S$, define the product

$$GH = \{k \mid k = gh, g \in G, h \in H\}. \tag{1}$$

It is clear that $(GH) \subseteq S$, and that the product of subsets of a semigroup is associative. Also the following distributive laws hold: For $G, H, K \subseteq S$,

$$G(H \cup K) = GH \cup GK, \tag{2}$$

$$(G \cup H)K = GK \cup HK.$$

If $G \subseteq S$, then the set of all products of elements of G (products consisting of single terms included) is a subsemigroup of S. This set of products is defined by:

$$G^+ = G \cup G^2 \cup G^3 \cup \cdots = \bigcup_{n=1}^{\infty} G^n. \tag{3}$$

In case $S = G^+$, S is said to be *generated* by G, and G is a *generating* set of S [1]. Every semigroup has at least one generating set (S generates itself). Note that the following properties hold:

$$(G^+)^+ = G^+, \tag{4}$$

$$\text{if} \quad G \supseteq H, \quad \text{then} \quad G^+ \supseteq H^+, \tag{5}$$

* Department of Electrical Engineering

The majority of this work was done while the author was on a visiting appointment in the Department of Electrical Engineering, University of California, Berkeley, California. Part of this work was supported by the National Research Council of Canada, Grant No. A-1617. This paper was presented at the Seventh Annual Symposium on Switching and Automata Theory, Berkeley, California, 1966.

for all subsets G and H of semigroup S. A generating set G of semigroup S is *irreducible* iff no proper subset of G generates S. In general, infinite semigroups may have no irreducible generating sets [1]. An element s of semigroup S is *indecomposable* iff it cannot be represented in the form $s = xy$ with $x, y \in S$. It follows that if s is indecomposable in S, then s must be a member of every generating set of S. If the set of all indecomposable elements generates S, then that set is the unique irreducible generating set of S [1]. The set S_I of all indecomposable elements of S is [1]:

$$S_I = S - S^2. \tag{6}$$

Definition 1. *A semigroup with length* is a semigroup S such that to every $s \subset S$ is assigned a unique positive integer $l(s)$, called the length of s, which obeys

$$l(xy) = l(x) + l(y), \tag{7}$$

for all $x, y \in S$.

It is clear that every free semigroup generated by a finite alphabet of letters is a semigroup with length, where the length $l(x)$ of a word x is the number of letters in x. However, a semigroup need not be free in order to have length. For example, consider the free semigroup S generated by $\{0, 1\}$, $0 \neq 1$. Then define $x = y$ iff $l(x) = l(y) \geq 2$, for $x, y \in S$. This defines a new semigroup S' which is not free but has length.

THEOREM 1. *Every semigroup S with length has a unique irreducible generating set S_I, contained in every other generating set and consisting of all the indecomposable elements of S, i.e.,*

$$S = (S - S^2)^+ = S_I{}^+. \tag{8}$$

PROOF. Clearly $S \supseteq S_I{}^+$. Let s be a shortest element of S; then s is indecomposable and $s \in S_I$. This establishes a basis for a proof by induction on $l(s)$. Suppose for all $s \in S$, $l(s) \leq n$ implies $s \in S_I{}^+$. Choose any $t \in S$ with $l(t) = n + 1$. If none exists, the argument holds vacuously. Otherwise, if t is indecomposable, then $t \in S_I$. If t is decomposable then there exist $x, y \in S$ such that $t = xy$ and $1 \leq l(x), l(y) \leq n$. By the induction hypothesis $x \in S_I{}^+$ and $y \in S_I{}^+$. But $S_I{}^+$ is closed under multiplication; hence $xy = t \in S_I{}^+$. Thus the induction step holds and $S_I{}^+ \supseteq S$. Now $S = S_I{}^+$, and the theorem follows.

COROLLARY. *Let S be a semigroup with length and let $G \subseteq S$, $H \subseteq S$. If $S = G^+ = H^+$, then $S = (G \cap H)^+$.*

PROOF. This is true because every generating set of S must contain the set S_I of all the indecomposable elements of S and S is generated by S_I.

Since $S = S^+$, S itself generates S and S is the largest generating set of S. Thus the generating sets of S form a lattice under inclusion, with S as the largest element and S_I as the least element.

A *monoid* M is a semigroup possessing a unit element e, satisfying $em = me = m$ for all $m \in M$. A semigroup can have at most one unit [1]. Given a subset G of M, we define the monoid generated by G to be

$$G^* = \{e\} \cup G \cup G^2 \cup \cdots = \bigcup_{n=0}^{\infty} G^n, \tag{9}$$

where G^0 is defined to be $\{e\}$. Note that

$$(G^*)^* = G^*, \tag{10}$$

and if $G \supseteq H$ then

$$G^* \supseteq H^*. \tag{11}$$

Of course if $e \in G$, then $G^* = G^+$, but in general $G^* = G^+ \cup \{e\}$. We are interested in monoids generated by subsets without e and so the union in (9) begins with $n = 0$. If $M = G^*$ we say that G *generates* M. A generating set G of monoid M is *irreducible* iff no proper subset of G generates M. It follows that an irreducible generating set does not contain e. An element m of a monoid M has no *proper decomposition* iff it cannot be represented in the form $m = xy$, with $x, y \subset M$ and $x, y \neq e$. Of course every $m \in M$ has trivial decompositions $m = em = me$; these are not proper.

Definition 2. *A monoid with length* is a monoid M, in which to every $m \subset M$, $m \neq e$ is assigned a positive integral length $l(m)$, $l(e) = 0$, and $l(xy) = l(x) + l(y)$ for all $x, y \in M$. (It is clear that $l(e)$ must be zero if l is to have the length property and e the property of a unit.)

THEOREM 2. *Every monoid M with length has a unique irreducible generating set M_I contained in every other generating set and consisting of all the elements of M that have no proper decomposition. Also*

$$M_I = (M - \{e\}) - (M - \{e\})^2. \tag{12}$$

PROOF. This follows directly from Theorem 1.

COROLLARY. *If M is a monoid with length and $M = G^* = H^*$, then $M = (G \cap H)^*$.*

As before, the generating sets of M form a lattice under inclusion, with M as the largest element and M_I as the smallest.

Let $A = \{a_1, a_2, \cdots, a_k\}$ be a finite nonempty alphabet and let A^* be the free monoid generated by A under concatenation, and with λ as the empty word. The *length* of a word w is the number of symbols in w, and $l(\lambda) = 0$. Any subset of A^* is called an *event*.

Definition 3. An event W is a *star event* iff there exists an event V such that $W = V^*$. In that case V is called a *root* of W.

THEOREM 3. *If W is a star event there exists a unique root W_I of W contained in every other root of W and given by $W_I = (W - \Lambda) - (W - \Lambda)^2$, where $\Lambda = \{\lambda\}$. W_I will be called the* minimum *root of W.*

PROOF. Every star event is a monoid with length. Hence Theorem 3 is a special case of Theorem 2. As before, we have if $W = U^* = V^*$, then $W = (U \cap V)^*$, and the roots of W form a lattice.

2. *Regular Events*

A brief review of terminology and notation is given below. The reader is referred to the author's paper on derivatives for further details [3].

Definition 4. Let $A = \{a_1, a_2, \cdots, a_k\}$ be a finite nonempty alphabet and let λ and ϕ be two distinct symbols not in A. Let $\cup$, $\cap$ and $\cdot$ be binary operators and let * and $-$ be unary operators.

Regular expression is defined inductively:

(1) $a_1, a_2, \cdots, a_k, \lambda$, and ϕ are regular expressions.

(2) If P and Q are regular expressions then so are $(P \cup Q)$, $(P \cap Q)$, (PQ) (the dot is omitted), P^*, and $\bar{P}$.

(3) Nothing else is a regular expression unless its being so follows from a finite number of applications of Rules 1 and 2.

Let A^* be the free monoid generated by A under concatenation $(\cdot)$, with λ, the empty word, as identity. Regular expressions denote events according to the mapping $|\ |$:

Regular expressions $\rightarrow$ events,

$|a_i| = \{a_i\}$, $|\lambda| = \{\lambda\}$, $|\phi| = \phi$, the empty event,

$|P \cup Q| = |P| \cup |Q|$ (union), $|P \cap Q| = |P| \cap |Q|$ (intersection),

$|PQ| = |P||Q|$ (concatenation), $|P^*| = |P|^*$ (closure), and

$|\bar{P}| = \overline{|P|}$ (complement with respect to A^*).

The notation $P = Q$, $P \supseteq Q$, $w \in P$, for P and Q regular expressions is to be interpreted $|P| = |Q|$, $|P| \supseteq |Q|$, $w \in |P|$, respectively. An event W is *regular* iff there exists a regular expression R such that $|R| = W$.

Definition 5. Let $u \in A^*$, $W \subseteq A^*$. The *left* quotient of W by u is denoted by $u \backslash W$ and defined by

$$u\backslash W = \{v \mid uw \in W\}.$$

Definition 6. Let P and Q be regular expressions, and let $\delta(P) = \lambda$ if $\lambda \in P$ and $\delta(P) = \phi$ if $\lambda \notin P$. The *derivative* of a regular expression with respect to a word $u \in A^*$ is a regular expression found recursively as follows.

(1) For $a \in A$, $D_a a = \lambda$, $D_a b = \phi$, for $b = \phi$ or $b = \lambda$ or $b \in A$ and $b \neq a$.

(2) $D_a(P \cup Q) = D_aP \cup D_aQ$,
$D_a(P \cap Q) = D_aP \cap D_aQ$,
$D_a(PQ) = (D_aP)Q \cup \delta(P)D_aQ$,
$D_a(P^*) = (D_aP)P^*$,
$D_a(\bar{P}) = \overline{D_aP}$.

(3) $D_\lambda P = P$.

(4) $D_{ua}P = D_a(D_uP)$.

It can be shown that derivatives denote quotients, i.e., if $W = |R|$, then $u\backslash W = |D_uR|$. Every regular expression has a finite number of derivatives [3].

Definition 7. *A (deterministic) finite automaton* S over alphabet A is a quadruple $S = \langle Q, M, q_1, F\rangle$, where $Q = \{q_1, q_2, \cdots, q_n\}$ is a finite, nonempty set of *internal states* of S, M is the *transition function* $M: Q \times A \rightarrow Q$, $q_1 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final* or *accepting* states.

If finite automaton S accepts the event denoted by a regular expression R, then the states of S correspond to derivatives of R. State q_i is in F iff the corresponding derivative contains λ [3].

We need the following properties, whose proofs are found in [3] or are easily supplied:

(a) $w \in R$ iff $\lambda \in D_wR$.

(b) $R = A^*$ iff $\lambda \in D_w R$ for all $w \in A^*$.
(c) $R = \phi$ iff $\lambda \notin D_w R$, for all $w \in A^*$.
(d) $P \supseteq Q$ iff $P \cup \bar{Q} = A^*$.
(e) $P = Q$ iff $P \oplus Q = \phi$, where $\oplus$ is the exclusive OR operation.
(f) Every regular expression can be expanded in terms of its derivatives:

$$R = \bigcup_{i=1}^{k} a_i D_{a_i} R \cup \delta(R).$$

(g) For R as in (f)

$$\bar{R} = \bigcup_{i=1}^{k} a_i D_{a_i} \bar{R} \cup \delta(\bar{R})$$

$$= \bigcup_{i=1}^{k} a_i \overline{D_{a_i} R} \cup (\lambda \cap \overline{\delta(R)}).$$

(h) If R is given as in (f) and if

$$P = \bigcup_{i=1}^{k} a_i D_{a_i} P \cup \delta(P),$$

then

$$R \cap P = \bigcup_{i=1}^{k} a_i (D_{a_i} R \cap D_{a_i} P) \cup (\delta(R) \cap \delta(P))$$

$$= \bigcup_{i=1}^{k} a_i D_{a_i} (R \cap P) \cup (\delta(R) \cap \delta(P)).$$

(i) The relative complement $R - P = R \cap \bar{P}$ can be found by combining (g) and (h).

3. *Regular Star Events*

THEOREM 4. *Let W be a star event; W is regular iff W_I is regular.* (See Theorem 3.)

PROOF. Suppose W_I is regular and is denoted by expression R_I; then $R = R_I^*$ is regular. Conversely, if R is regular then so is $(R - \Lambda) - (R - \Lambda)^2 = R_I$, because only regular operations on regular sets are used.

It should be pointed out that a regular event may have an irregular root. For example,

$$\{0, 1\}^* = (\{0, 1\} \cup \{w \mid w = 0^n 1^n, n \geq 0\})^*,$$

where the root on the right is irregular.

First we consider the problem of determining whether a given regular expression denotes a star event, in which case we call it a *star expression*. The following theorem of Paz and Peleg [4] is adapted to our notation.

THEOREM 5. *Let R be a regular expression. Then each of the following conditions is necessary and sufficient for R to be a star expression:*

(a) $R = R^*$.
(b) $R = R^2$.
(c) $\lambda \in R$, *and for each* $w \in A^*$, $\lambda \in D_w R$ *iff* $D_w R \supseteq R$.

	0	1	
q_1	q_2	q_1	1
q_2	q_3	q_7	0
q_3	q_2	q_4	1
q_4	q_5	q_1	1
q_5	q_6	q_1	1
q_6	q_6	q_4	1
q_7	q_7	q_7	0

FIG. 1. Finite automaton for Example 1

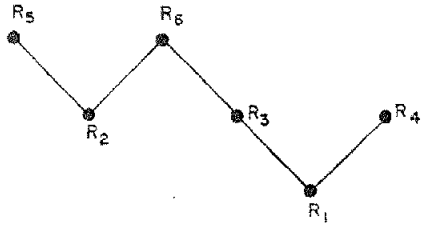

FIG. 2

PROOF. Part (a) follows because $(P^*)^* = P^*$. For part (b), clearly $R = R^2$ is necessary for $P^*P^* = P^*$. To prove it is sufficient, let the length of a shortest word of R be d. Then the length of a shortest word of R^2 is $2d$. But $R = R^2$, so $d = 2d = 0$. Thus $R = R^2$ implies $\lambda \in R$. Also note that $R = R^2$ implies $R = R^n$ for all $n \geq 1$. Thus $R = \lambda \cup R \cup R^2 \cup R^3 \cup \cdots = R^*$.

For part (c), assume that R is a star. Then $\lambda \in R$ and $D_wR = D_wR^2 = (D_wR)R \cup D_wR$, from which it is clear that $\lambda \in D_wR$ implies $D_wR \supseteq R$. Also if $D_wR \supseteq R$, then $\lambda \in D_wR$, since $\lambda \in R$. Conversely suppose part (c) is true for R. If $w \notin R$, then $\lambda \notin D_wR$ and $D_wR \not\supseteq R$. For any $w \in R$, $\lambda \in D_wR$, so $D_wR \supseteq R$. Hence $wR \subseteq R$. But this is true for all $w \in R$, so $R^2 \subseteq R$. Also since $\lambda \in R$, $R^2 \supseteq R$. Therefore $R = R^2$ and by (b) R is a star.

In order to test whether a given regular expression R is a star one could construct state diagrams for R and R^* and test for isomorphism. Alternately construct $R \oplus R^*$ and test whether $R \oplus R^* = \phi$, by constructing the derivatives of $R \oplus R^*$ and using property (e) of Section 2. Similarly (b) could be used. However, these methods require the construction of two state diagrams or deal with two expressions. This is not necessary, as is shown below.

Example 1. Consider the state table shown in Figure 1. It defines the finite automaton $S = \langle Q, M, q_1, F \rangle$ over input alphabet $A = \{0, 1\}$, where $Q = \{q_1, \cdots, q_7\}$, q_1 is the initial state, the M function is given by the table, and final states are distinguished by a 1 in the rightmost column (thus $F = \{q_1, q_3, q_4, q_5, q_6\}$).

Let R_1 be the regular expression accepted by S, i.e., corresponding to the initial state q_1. Let R_i be the derivative of R_1 corresponding to q_i. Then the derivative equations for R_1 are obtainable directly from Figure 1, and are as follows:

$$
\begin{aligned}
R_1 &= 0R_2 \cup 1R_1 \cup \lambda, \\
R_2 &= 0R_3 \cup 1R_7, \\
R_3 &= 0R_2 \cup 1R_4 \cup \lambda, \\
R_4 &= 0R_5 \cup 1R_1 \cup \lambda, \\
R_5 &= 0R_6 \cup 1R_1 \cup \lambda, \\
R_6 &= 0R_6 \cup 1R_4 \cup \lambda, \\
R_7 &= 0R_7 \cup 1R_7.
\end{aligned}
$$

(By noting that the last equation is independent of the others and has the solution $R_7 = \phi$, we can simplify the above equations by removing R_7 and rewriting $R_2 = 0R_3$.)

If R_1 were specified by a regular expression rather than by a state diagram, we could construct the derivatives of R_1, and then the derivative equations. At any rate, assume that the derivative equations are available and we are to test whether R_1 is a star expression by using part (c) of Theorem 5. Since q_1 is the initial state we must test whether all expressions corresponding to output states contain R_1. Obviously $R_1 \supseteq R_1$, so we next test R_3, directly from the derivative equations, by comparing it to R_1 term by term, and so on:

$$(R_3 \supseteq R_1) \rightarrow (R_4 \supseteq R_1) \rightarrow (R_5 \supseteq R_2) \rightarrow (R_6 \supseteq R_3)$$

and $(R_1 \supseteq R_7)$. The last condition is trivially true for $R_7 = \phi$. Continuing with the chain,

$$(R_6 \supseteq R_3) \rightarrow (R_6 \supseteq R_2) \rightarrow (R_6 \supseteq R_3) \quad \text{and} \quad (R_4 \supseteq R_7).$$

The last of these is trivially true, and in the chain there are no contradictions. Hence all of the above containments are true, giving the partial ordering from this test as shown in Figure 2. In any case R_3, R_4, R_6 all contain R_1. Also one verifies that $R_5 \supseteq R_1$. Thus all derivatives containing λ do in fact contain R_1. Hence R_1 is a star expression and it can be shown that

$$R_1 = (1 \cup 00 \cup 0010)^*.$$

Notice that the above process of testing for inclusion is very similar to the reduction of state tables, except that we are testing here only for inclusion relations among the derivatives, rather than for equivalence of derivatives.

4. *Simple Stars*

We now consider the problem of finding the minimum star roots of regular star events.

Definition 8. A regular star expression R is *simple* iff $\lambda \in D_w R$ implies $D_w R = R$.

This implies that the reduced Moore state diagram for R has only one output state, namely the initial state. This special case will be treated first because of its simplicity.

THEOREM 6. *If R is a simple regular star, then the minimum root of R is the regular set V of all words w of length > 0, such that w takes the state diagram of R from the initial state back to the initial state without going* through *the initial state.*

PROOF. If $w \in R$, $l(w) > 0$, then w leads to an output state. Since R is simple, the only output state is q_1, the initial state. Thus all $w \in R$ take the state diagram from q_1 back to q_1. Also w must be of the form $w = v_1 v_2 \cdots v_n$, $v_i \in V$, for $i = 1, 2, \cdots, n$. Thus $w \in V^n$ and $V^* \supseteq R$ follows. Conversely $w \in V^*$ implies $w \in V^n$ for some n, and w leaves the state diagram in q_1, so $w \in R$. Thus $R \supseteq V^*$ and therefore V is a root of R. Suppose V is not the minimum root. Let $U \subseteq V$ be a smaller root and let $w \in V$ but $w \notin U$. Then $w = u_1 u_2 \cdots u_n$, $u_j \in U$ for $j = 1, 2, \cdots, n$. Since $U \subseteq V$, we have $u_j \in V$ for $j = 1, 2, \cdots, n$. Thus V contains a word w that takes the state diagram of R from q_1 to q_1 and goes through q_1 at least once, which is a contradiction. Thus V is the minimum root.

COROLLARY 1. *V has the prefix and suffix properties.* (An event V has the *prefix* (*suffix*) property [5] when for any $u, v \in V$, $u \neq v$, u is not a prefix (suffix) of v, nor is v a prefix (suffix) of u.)

COROLLARY 2. *A regular expression for V is obtained from the derivative equations of R as follows*

(*a*) *All appearances of R on the right side of the equations are replaced by* λ.

(*b*) *On the left side of the equation for R, replace R by P. On the right side of the same equation remove* λ.

(*c*) *Solve the resulting equations for P.*

(*d*) $P = V$.

The proof of this construction is easily supplied. Of course the minimum root can also be obtained from the state diagram for R by other methods, e.g., signal flow graph techniques [6].

Example 2. $R = \lambda \cup 1(0 \cup 11)^*1$. The equations for R are:

$$R = 1R_1 \cup \lambda, \qquad R_1 = 0R_1 \cup 1R.$$

Modifying the equations as in Corollary 2, we obtain

$$P = 1R_1, \qquad R_1 = 0R_1 \cup 1\lambda = 0R_1 \cup 1.$$

Solving, we have $R_1 = 0^*1$, $P = 10^*1$, and $R = (10^*1)^*$.

Remark. This example points out a rather disappointing feature of the root. From $R = (10^*1)^*$, it is seen that the minimum root is infinite and one may erroneously conclude that R is of star height 2[7]. In fact R is of star height 1, for $R = \lambda \cup 1(0 \cup 11)^* 1$. Thus any canonical form based on minimum star roots will not necessarily display the star height of the expression. The proper star height will be displayed if R is finite and $R \neq \lambda$. The latter condition must be added, for λ is a star expression, $\lambda = \phi^*$.

5. *Finding the Minimum Root—General Case*

In general, if R is a star expression we can only conclude that $\lambda \in D_w R$ implies $D_w R \supseteq R$, by Theorem 5. Thus the state diagram contains several output states, and the root need not have the prefix and suffix properties. The situation is therefore considerably more complicated. In a sense, the problem of finding the minimum root is solved by Theorem 3, for given any star expression R, we have for its root the expression $R_I = (R - \lambda) - (R - \lambda)^2$. However, this expression is considerably more complicated than R itself and uses the relative complement operator which may be undesirable. For instance, in Example 2 we have $R = \lambda \cup 1(0 \cup 11)^*1$, and so

$$R_I = (1(0 \cup 11)^*1) - (1(0 \cup 11)^*1)^2.$$

This is certainly not very illuminating, and the form $R_I = 10^*1$ is a much simpler expression. Thus, although a closed form for the minimum root does exist, it has some serious disadvantages. For this reason we wish to obtain the minimum root by other methods.

LEMMA 1. *Let R be a regular expression and let $w \in A^*$, $l(w) > 0$. Then there exists a unique expansion of w with respect to R, $w = u_1 u_2 \cdots u_n v$, where $l(u_i) > 0$, for $i = 1, 2, \cdots, n$ and*

(1) $u_1 u_2 \cdots u_i \in R$ *for* $1 \leq i \leq n$

(2) *no proper prefix of w, other than the prefixes in* (1), *is a member of R;*

(3) $w \in R$ *iff* $v = \lambda$.

PROOF. Simply choose u_1, u_1u_2, $\cdots$, $u_1 \cdots u_n$, to be all the prefixes of w which lead to output states. If there are no such prefixes, then $w = v$.

LEMMA 2. *Let R be a regular star expression and let R_I be the set of all words that have no proper decomposition in R. Let $w \in R$, $l(w) > 0$ and let the expansion of w with respect to R be $w = u_1u_2 \cdots u_n$. Then $w \in R_I$ iff $\lambda \in \epsilon_w R$, where*

$$\epsilon_w R = D_{u_1 \cdots u_n}R - D_{u_2 \cdots u_n}R - \cdots - D_{u_n}R. \tag{13}$$

PROOF. If $\lambda \in \epsilon_w R$, then $\lambda \in D_{u_1 \cdots u_n}R = D_wR$ and $\lambda \notin D_{u_i \cdots u_n}R$, for $2 \leq i \leq n$. Thus $w \in R$ and no suffix $u_i \cdots u_n$ of w is in R, if the corresponding prefix $u_1 \cdots u_{i-1}$ is in R. Thus w has no proper decomposition. Conversely, assume $w \in R_I$, but $\lambda \notin \epsilon_w R$. Then either $w \notin R$ or $\lambda \in D_{u_i \cdots u_n}R$. In the latter case w has a proper decomposition $w = (u_1 \cdots u_{i-1})(u_i \cdots u_n)$. In any case we have a contradiction, so $w \in R_I$ implies $\lambda \in \epsilon_w R$.

We now show that the minimum root of a regular star expression R can be found from the derivative equation for R after some simple modifications are introduced. The minimum root is also displayed with the aid of a nondeterministic finite automaton.

Definition 9. *A nondeterministic finite automaton* S over alphabet A is a quadruple $S = \langle P, N, p_1, G \rangle$ where $P = \{p_1, p_2, \cdots, p_n\}$ is a finite, nonempty set of *internal states* of S; $N: P \times A \to$ subsets of P, is the *transition function*; $p_1 \in P$ is the *initial state*; and $G \subseteq P$ is the set of *final states*.

Nondeterministic state diagrams have been used by Even [8] for similar problems. We show that they can be used naturally to find minimum roots of arbitrary regular events.

Definition 10. Given a regular star expression R, let $S(R)$ be the nondeterministic finite automaton related to R and constructed as follows. Let

$$S(R) = \langle P, N, p_1, G \rangle.$$

(1) The states of $S(R)$ correspond to differences of derivatives of R, where a difference of derivatives is any expression of the form $\gamma_j R = D_{w_1}R - D_{w_2}R - \cdots - D_{w_k}R$.

(2) The initial state of $S(R)$ corresponds to R, i.e., $p_1 = R$.

(3) The only final state of $S(R)$ is $p_1 = R$, i.e., $G = \{R\}$.

(4) For any $a \in A$, $N(R, a) = \{D_aR\}$ if $\lambda \notin D_aR$, and $N(R, a) = \{R, D_aR - R\}$ if $\lambda \in D_aR$.

(5) Let $\gamma_j R = D_{w_1}R - D_{w_2}R - \cdots - D_{w_k}R$, and let $\gamma_{ja}R = D_{v_1}R - D_{v_2}R - \cdots - D_{v_k}R$, where $D_{v_j}R = D_{w_j a}R$, for $j = 1, 2, \cdots, k$. Then $N(\gamma_j R, a) = \gamma_{ja}R$, if $\lambda \notin \gamma_{ja}R$, and $N(\gamma_j R, a) = \{(\gamma_{ja}R) - R, R\}$, if $\lambda \in \gamma_{ja}R$.

Clearly the state diagram so defined has a finite number of states, since R has a finite number of differences of derivatives.

Example 3. Let $R = (0 \cup 01 \cup 011 \cup 110^*1)^*$. One verifies that R satisfies the reduced derivative equations:

$$\begin{aligned} R &= 0R_0 \cup 1R_1 \cup \lambda, \\ R_0 &= 0R_0 \cup 1R_0 \cup \lambda, \\ R_1 &= 1R_{11}, \\ R_{11} &= 0R_{11} \cup 1R. \end{aligned}$$

	0	1	
(R)	$(R_0 - R, R)$	(R_1)	λ
$(R_0 - R)$	ϕ	$(R_0 - R_1 - R, R)$	ϕ
(R_1)	ϕ	(R_{11})	ϕ
(R_{11})	(R_{11})	(R)	ϕ
$(R_0 - R_1 - R)$	ϕ	$(R_0 - R_{11} - R_1 - R, R)$	ϕ
$(R_0 - R_{11} - R_1 - R)$	ϕ	$(R_0 - R - R_{11} - R_1)$	ϕ

$N(p_i, p_j)$

FIG. 3

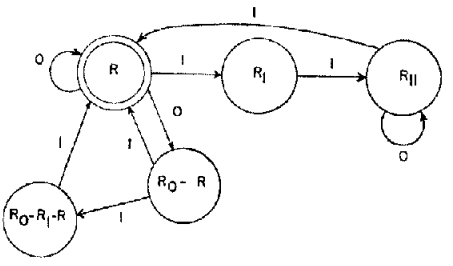

FIG. 4. $S(R)$ for Example 3

We construct $S(R)$ as follows. The initial state is R. Since $\lambda \in R_0$ and $\lambda \notin R_1$, this defines $N(R, 0) = \{R_0 - R, R\}$, $N(R, 1) = \{R_1\}$, and R is identified as the output state by the presence of λ. Next, since $R_0 - R = 0(R_0 - R_0) \cup 1(R_0 - R_1) \cup (\lambda - \lambda) = 0\phi \cup 1(R_0 - R_1)$, we have $N(R_0 - R, 0) = \phi$. Notice also that $R_0 - R_1 = 0R_0 \cup 1(R_0 - R_{11}) \cup \lambda$ contains λ. Hence $N(R_0 - R, 1) = \{R_0 - R_1 - R, R\}$. Also $N(R_1, 0) = \phi$, $N(R_1, 1) = \{R_{11}\}$, and so on. The table for $S(R)$ is shown in Figure 3. Notice that $R_0 - R_{11} - R_1 - R = R_0 - R - R_{11} - R_1$ corresponds to the empty state ϕ. The state diagram of $S(R)$ is shown in Figure 4, where the empty state ϕ is not shown.

LEMMA 3. *Let $R = R_I^*$ be regular and let $w \in A^*$ be expanded with respect to R as in Lemma 1, $w = u_1 u_2 \cdots u_n v$. Then there exists a path in $S(R)$ which corresponds to w and takes the initial state to a state corresponding to*

$$\Delta_w R = D_{u_1 \cdots u_n v} R - D_{u_2 \cdots v} R - \cdots - D_{u_n v} R - D_v R. \tag{14}$$

Furthermore, the path does not go through the initial state.

PROOF. Let p be a prefix of w and let v_i be a proper prefix of u_i for $i = 1, 2, \cdots, n$. Then the following sequence of differences as derivatives is found:

$$\begin{aligned} p &= v_1 ; & \Delta_p R &= D_{v_1} R. \\ p &= u_1 ; & \Delta_p R &= D_{u_1} R - R. \\ p &= u_1 v_2 ; & \Delta_p R &= D_{u_1 v_2} R - D_{v_2} R. \\ p &= u_1 u_2 ; \end{aligned}$$

according to the rules, first find $\gamma_j R = D_{v_1 u_2} R - D_{u_2} R$. If $\lambda \in \gamma_j R$, use $\Delta_p R = D_{u_1 u_2} R - D_{u_2} R - R$. If $\lambda \notin \gamma_j R$, this means $\lambda \in D_{u_2} R$ or $D_{u_2} R \supseteq R$ by Theorem 5. Thus $\gamma_j R = D_{u_1 u_2} R - (D_{u_2} R \cup R) = D_{u_1 u_2} R - D_{u_2} R - R$ as before. Hence $\Delta_w R$ has the proper form (14). Also, no $\Delta_p R$ above contains λ, so $R \neq \Delta_p R$. Thus each state in the above sequence is different from the initial state (R).

LEMMA 4. *Let Q be the set of all words of length > 0 that take $S(R)$ from the initial state back to the initial state without going through the initial state. Then $w \in Q$ iff $\lambda \in \epsilon_w R$.* (See Lemma 2.)

PROOF. Let $w = u_1 \cdots u_n v a = xa$ where $va = u_{n+1}$ and $l(a) = 1$. By Lemma 3, in $S(R)$ there is a state $\Delta_x R = D_x R - \cdots - D_{u_n v} R - D_v R$. Compute $D_a(\Delta_x R) = \epsilon_w R$. By construction if $\lambda \in \epsilon_w R$ then a transition from $\Delta_x R$ to R under a is introduced and so $w \in Q$. Otherwise $w \notin Q$. Thus $w \in Q$ iff $\lambda \in \epsilon_w R$.

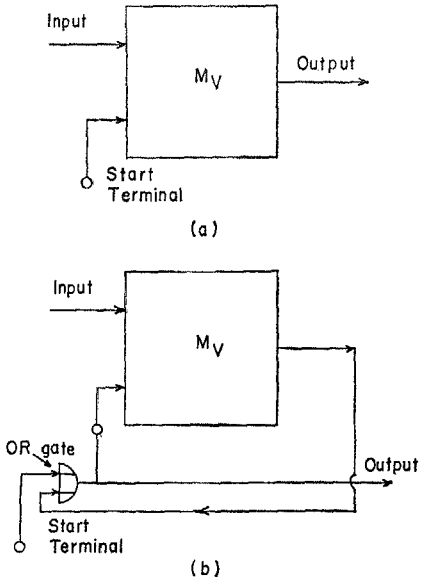

FIG. 5. (a) Machine M_V ; (b) realization of M_V*

THEOREM 7. $Q = R_I$.

PROOF. We show that $w \in R_I$ iff $\lambda \in \epsilon_w(R)$ in Lemma 2. Combining this with $w \in Q$ iff $\lambda \in \epsilon_w(R)$ of Lemma 4, we obtain $w \in R_I$ iff $w \in Q$.

The theorem now shows that the minimum root of any regular star expression R is conveniently exhibited by the state diagram of $S(R)$.

6. *Applications*

A. *Semigroups.* The study of roots of star events has lead us to examine generating sets of abstract semigroups. We show that every semigroup with length is generated by the set of its indecomposable elements. This is not necessarily true for semigroups without length.

B. *Canonical Forms of Regular Expressions.* Satisfactory canonical forms have not been found for arbitrary regular expressions. Since regular expressions can be formed using only the $\cup$, $\cdot$, and * operators, and the star operator is most powerful of these in generating words, we feel that by studying star events, we are considering one of the key problems. If $R = P^*$, it is natural to assume that finding the minimum root of R will lead to a useful canonical form, provided that concatenation and union can be handled. At any rate it is shown that a unique regular *set* can be found for the star root, though not necessarily a unique regular expression. In case the root is finite, one gets of course a unique expression.

C. *Simplification of Regular Expressions.* In many cases the finding of minimum star roots results in considerable simplification of the star expression. We leave it to the reader to find the regular expression accepted by the automaton of Example 1, in any way, and to compare it to the form $(1 \cup 00 \cup 0010)^*$, obtained by using the minimum root.

D. *Formal languages.* If W represents any formal language (not necessarily regular) it is of interest to find the smallest set of words generating the language.

E. *Decomposition of Sequential Machines.* If $W = V^*$ is regular, the machine for W is constructed from the machine for V as shown in Figure 5 [9]. Note that the machine M_V is not of standard design, but is a special machine with a starting terminal [9].

If V is minimum, one would expect the structure of M_V to be simple and inexpensive.

F. *Regular Codes.* Any event C can be considered as a *code* consisting of (code) words. Products of code words form *messages*. The set of all messages is clearly C^*. We can define a code C to be *effectively regular* iff C^* is regular. This is done because a code may be irregular and yet its message set may be regular. In that case the original code can be replaced by a regular code generating the same message set. A message set C^* is *uniquely decipherable* iff whenever $x_1x_2 \cdots x_m = y_1y_2 \cdots y_n$, with $x_1, x_2, \cdots, x_m, y_1, y_2, \cdots, y_m \in C$, then $n = m$ and $x_i = y_i$ for $1 \leq i \leq n$. It follows that C^* is uniquely decipherable iff it is isomorphic to the free monoid generated by C.

The results of this paper make it easier to treat infinite codes. For example, one can verify the following result:

THEOREM 8. *A regular message set C^* is uniquely decipherable iff $S(C^*)$, the nondeterministic automaton related to C^*, is information lossless* [8].

One also easily verifies that all message sets which are simple stars (Definition 8) are uniquely decipherable.

ACKNOWLEDGMENTS. The author wishes to thank Professors M. A. Harrison and L. H. Haines of University of California, Berkeley, for useful comments on this work.

REFERENCES

1. LJAPIN, E. S. Semigroups. Translations of Math. Monographs, No. 3, Amer. Math. Soc., Providence, R. I., 1963.
2. CLIFFORD, A. H., AND PRESTON, G. B. *The Algebraic Theory of Semigroups. Math. Surveys series, Vol. 1.* Amer. Math. Soc., Providence, R. I., 1961.
3. BRZOZOWSKI, J. A. Derivatives of regular expressions. *J. ACM 11*, 4 (Oct. 1964), 481–494.
4. PAZ, A., AND PELEG, B. Concatenative decompositions of regular events. Tech. Rep. No. 20, The Hebrew University, Jerusalem, Israel, 1965.
5. BRZOZOWSKI, J. A. Regular expressions for linear sequential circuits. *IEEE Trans. EC-14*, 2 (April 1965), 148–156.
6. —— AND MCCLUSKEY, E. J., JR. Signal flow graph techniques for sequential circuit state diagrams. *IEEE Trans. EC-12*, 2 (April 1963), 67–76.
7. EGGAN, L. C. Transition graphs and star height of regular events. *Mich. Math. J. 10* (1963), 385–397.
8. EVEN, S. On information lossless automata. Ph.D. dissertation, Harvard U., Cambridge, Mass., 1963.
9. BRZOZOWSKI, J. A. A survey of regular expressions and their applications. *IRE Trans. EC-11*, 3 (June 1962), 324–335.

RECEIVED SEPTEMBER, 1966; REVISED FEBRUARY, 1967