Theoretical Computer Science 14 (1981) 63–77
North-Holland Publishing Company

ON GENERALIZED LANGUAGE EQUATIONS*

Ernst LEISS
Department of Computer Science, University of Houston, Houston, TX 77004, U.S.A.

Communicated by M. Nivat
Received February 1979
Revised January 1980

Abstract. A system of generalized language equations over an alphabet A is a set of n equations in n variables: $X_i = G_i(X_1, \ldots, X_n)$, $i = 1, \ldots, n$, where the G_i are functions from $[P(A^*)]^n$ into $P(A^*)$, $i = 1, \ldots, n$, $P(A^*)$ denoting the set of all languages over A. Furthermore the G_i are expressible in terms of set-operations, concatenations, and stars which involve the variables X_i as well as certain fixed languages. In this note we investigate existence and uniqueness of solutions of a certain subclass of generalized language equations. Furthermore we show that a solution is regular if all fixed languages are regular.

1. Notation

If A is an alphabet, then A^* denotes the set of words over A. The length $|w|$ of a word w is the number of elements of A in w; note that the length of the empty word λ is 0. A language (over A) is any subset of A^*. If L_1 and L_2 are languages, so are

$$L_1 \cup L_2, \quad L_1 \cap L_2, \quad L_1 L_2, \quad \bar{L}_1, \quad L_1^*,$$

denoting the union, intersection, concatenation of L_1 and L_2, the complement of L_1 with respect to A^*, and the star of L_1, respectively.

A finite automaton $\boldsymbol{A}$ is a quintuple $\boldsymbol{A} = (A, Q, M, q_0, F)$, where A is the input alphabet, Q the finite nonempty set of states, $q_0 \in Q$ the initial state, $F \subseteq Q$ the set of final states, and M the transition function, $M : Q \times A \to P(Q)$, where $P(Q)$ denotes the powerset of Q. If $M(q, a)$ consists of one element for all $q \in Q$, $a \in A$ the automaton $\boldsymbol{A}$ is called deterministic. The transition function M is extended to $P(Q) \times A^*$ as usual. A word $w \in A^*$ is accepted by $\boldsymbol{A}$ iff $M(q_0, w) \cap F \neq \emptyset$. The language $L(\boldsymbol{A})$ of words accepted by the automaton $\boldsymbol{A}$ is regular. A finite automaton $\boldsymbol{A}$ is called nonreturning iff $M(q_0, x) \cap \{q_0\} = \emptyset$ for all $x \in AA^*$. Clearly, to each automaton $\boldsymbol{A}$ there exists a nonreturning automaton $\boldsymbol{A}'$ such that $L(\boldsymbol{A}) = L(\boldsymbol{A}')$.

* This research was done while the author was at the University of Waterloo; it was supported by the National Research Council of Canada under grant No. A-1617.

A boolean automaton $\boldsymbol{B}$ is a generalization of a finite automaton which will be needed in the last section [2]: $\boldsymbol{B}=(A, Q, M, q_0, F)$, where A, Q, q_0, and F have the same meaning as for finite automata, and the transition function M is defined as follows: $M: Q\times A\to \mathcal{B}_Q$, where $\mathcal{B}_Q$ (or $\mathcal{B}$ if Q is understood) is the set of all boolean functions in the variables $\{q_0,\ldots,q_n\}=Q$. Thus Q is considered as the set of states of $\boldsymbol{B}$ on the one hand and on the other hand as the set of variables of $\mathcal{B}_Q$. We assume that the variables take the values 0 and 1. Operations are denoted by the corresponding set-operations, i.e. $\cup$ for (boolean) union, $\cap$ for intersection, etc. The transition function M is extended to $\mathcal{B}_Q\times A^*$ as follows: $M(f,\lambda)=f$ for all $f\in\mathcal{B}$; if $f=f(q_0,\ldots,q_n)\in\mathcal{B}$, then $M(f,a)=f(M(q_0,a),\ldots,M(q_n,a))$ for all $a\in A$; $M(f,wa)=M(M(f,w),a)$ for all $f\in\mathcal{B}$, $w\in A^*$, $a\in A$. Furthermore we define a relation $=_F$ on $\mathcal{B}_Q$, called evaluation under F: If $f=f(F;Q-F)$, then $f=_F\alpha$ iff $f(1,\ldots,1;\ 0,\ldots,0)=\alpha$, $\alpha\in\{0,1\}$. A word $w\in A^*$ is accepted by $\boldsymbol{B}$ iff $M(q_0,w)=_F 1$. The set of words accepted by $\boldsymbol{B}$ is denoted by $L(\boldsymbol{B})$. Given a boolean automaton $\boldsymbol{B}=(A,Q,M,q_0,F)$ we construct the derived deterministic automaton $\boldsymbol{A_B}$, $\boldsymbol{A_B}=(A,P,N,q_0,G)$, where $P=\{f\in\mathcal{B}\mid f=M(q_0,x)$ for some $x\in A^*\}$, $G=\{f\in P\mid f=_F 1\}$, and $N(f,a)=M(f,a)$ for all $f\in P$, $a\in A$. Since $\mathcal{B}_Q$ is finite so is P, thus $\boldsymbol{A_B}$ is a finite automaton. Furthermore one shows $L(\boldsymbol{B})=L(\boldsymbol{A_B})$ [2]. Interpreting set theoretical union as boolean addition we can consider finite automata as a special case of boolean automata: a subset P of the set Q of states corresponds to the boolean union of all elements $p\in P$ and the empty state $\emptyset$ corresponds to the constant function $0\in\mathcal{B}$.

Example 1: An example of a boolean automaton is $B=(\{a,b\},\{q_0,q_1\},M,q_0\cap\bar{q}_1,\emptyset)$, where M is given by the following table:

	a	b
q_0	$\bar{q}_1$	$\bar{q}_0\cup q_1$
q_1	$q_0\cup q_1$	$q_0\cap\bar{q}_1$

The corresponding derived deterministic automaton $\boldsymbol{A_B}$ is given in the following diagram with $q_0\cap\bar{q}_1$ being the initial state and $\{\bar{q}_0\cap\bar{q}_1, \bar{q}_0\cup q_1, 1\}$ the set of final states of $\boldsymbol{A_B}$:

	a	b	$=_F$
$q_0\cap\bar{q}_1$	$\bar{q}_0\cap\bar{q}_1$	$\bar{q}_0\cup q_1$	0
$\bar{q}_0\cap\bar{q}_1$	0	0	1
$\bar{q}_0\cup q_1$	$q_0\cup q_1$	$q_0\cap\bar{q}_1$	1
0	0	0	0
$q_0\cup q_1$	1	1	0
1	1	1	1

A system of generalized language equations is a set of n equations in n variables

$$X_i=G_i(X_1,\ldots,X_n),\quad i=1,\ldots,n$$

together with an initialization X_0. The G_i are functions from n-tuples of languages to languages, i.e. $G_i : [P(A^*)]^n \to P(A^*)$, while X_0 is a regular expression over the alphabet $X_1, \ldots, X_n$. Furthermore it is assumed that the G_i can be expressed in terms of set-operations, concatenations, and stars, involving as operands the X_i and certain fixed languages which can be considered as constants but may be quite arbitrary. For example consider the following system of generalized language equations over the alphabet $A = \{a, b\}$:

$$X_1 = (U_1X_1 \cap V_1\bar{X}_1)^*X_2 \cup Y_1,$$

$$X_2 = \overline{U_2X_1} \cup \overline{V_2\bar{X}_2} \cdot \overline{X_1 - Y_1},$$

$$X_0 = X_1X_2 \cap X_2X_1,$$

where $U_1 = \{a\}$, $U_2 = ba^*$, $V_1 = \{a^nb^n \mid n \geq 1\}$, $V_2 = \{b\}$, $Y_1 = \{\lambda\}$.

2. Introduction

Certain classes of equations have been studied quite extensively in the theory of formal languages. For instance it is known [1] that derivative equations uniquely determine a regular language; these are equations of the form $X_i = \bigcup_{a \in A} aX_{i,a} \cup \delta_i$, $i = 1, \ldots, n$, to be solved for X_m, where $X_m, X_{i,a} \in \{X_1, \ldots, X_n\}$, $\delta_i \in \{\{\lambda\}, \emptyset\}$. These equations immediately yield a deterministic finite automaton $\boldsymbol{A} = (A, \{X_1, \ldots, X_n\}, M, X_m, \{X_i | \delta_i = \{\lambda\}\})$, where $M(X_i, a) = X_{i,a}$ for all $i \in \{1, \ldots, n\}, a \in A$. Then $L(\boldsymbol{A}) = X_m$. If we consider these derivative equations as a special case of generalized language equations, we observe that the fixed languages are $\emptyset$, $\{\lambda\}$, and $\{a\}$ for $a \in A$. It is quite natural to allow more complicated fixed languages $U_{i,j}$, Y_i thus arriving at equations of the form $X_i = \bigcup_{j=1}^n U_{i,j}X_j \cup Y_i$. However, this is somewhat problematic since now a solution of such a system of equations need not be unique any longer. For instance consider the equation $X = (a \cup \lambda)X \cup \lambda$ over the alphabet $A = \{a, b\}$. Clearly $\{a\}^*$ is a solution, but $\{a, b\}^*$ is also one. However, if we assume $\lambda \notin U_{i,j}$ for all i, j we regain uniqueness of solutions; in the case of equations in one variable $X = UX \cup Y$ we have the unique solution $X = U^*Y$. Similar results are known for systems of the form $X_i = \bigcup_{j=1}^n X_jU_{i,j} \cup Y_i$. These results can be found in [5]; [3] and [4] also deal with such equations but their approaches are slightly different.

Recalling the definition of generalized language equations we so far dealt with functions G_i which could be expressed as unions of terms of the form $U_{i,j}X_j$ (or $X_jU_{i,j}$) as 'variables' and the fixed languages Y_j; furthermore the $U_{i,j}$ and Y_j are assumed to be regular languages. For the following we will remove some of these restrictions, in particular we will allow arbitrary set-functions (intersections, complements, etc.) rather than only unions, and at least in the beginning the $U_{i,j}$ and Y_j can be arbitrary languages. Furthermore we also allow terms of the form $U_{i,j}\bar{X}_j$ as 'variables'. For the greater part of this note we will consider systems of one equation in one variable only, however it is quite easy to obtain analogous results for arbitrary systems. This straightforward generalization is usually left to the reader.

3. Uniqueness

If we have equations of the form $X_i = \bigcup_{j=1}^{n} U_{i,j}X_j \cup Y_i$, we can ensure uniqueness of a solution by assuming $\lambda \notin U_{i,j}$ for all i, j. Such a condition will also turn out to be sufficient for the uniqueness of a solution of the more general systems considered below.

We will deal with one equation in one variable

$$X = G(X), \tag{3.1}$$

where G is a function from languages to languages expressible as follows:

$$G(X) = g(U_1X, \ldots, U_IX; V_1\bar{X}, \ldots, V_J\bar{X}; Y_1, \ldots, Y_K), \tag{3.2}$$

g being an $(I+J+K)$-ary set-function, and U_i, V_j, Y_k are arbitrary languages such that $\lambda \notin U_i$, $\lambda \notin V_j$, $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$, $k \in \{1, \ldots, K\}$.

For example, let $U_1 = aa^*$, $U_2 = \{a^{2^i} \mid i \geq 0\}$, $V_1 = \{a^ib^{i-1} \mid i \geq 1\}$, $Y_1 = \emptyset$ and let $g(z_1, \ldots, z_4) = (\bar{z}_1 \cap z_3) \cup (z_1 \cap \bar{z}_2) \cup z_4$, then the corresponding generalized language equation over $A = \{a, b\}$ is

$$X = (\overline{aa^*X} \cap \{a^ib^{i-1} | i \geq 1\}\bar{X}) \cup (aa^*X \cap \overline{\{a^{2^i} | i \geq 0\}X}).$$

Since we are dealing with systems of one equation we will tacitly assume that it is to be solved for the only variable X.

Theorem 1. *Let U_i, V_j, Y_k be arbitrary languages, $\lambda \notin U_i$, $\lambda \notin V_j$, and let g be an $(I+J+K)$-ary set-function. Then there exists at most one solution of the equation*

$$X = g(U_1X, \ldots, U_IX; V_1\bar{X}, \ldots, V_J\bar{X}; Y_1, \ldots, Y_K). \tag{3.3}$$

Proof. Since g is a set-function in the $I+J+K$ variables U_iX, $V_j\bar{X}$, Y_k, it can be represented as follows:

$$g = g_1 \cup \cdots \cup g_L \cup g_{L+1} \cup \cdots \cup g_M,$$

where each g_m for $m = 1, \ldots, M$ is an intersection of the variables or their complements. Furthermore we assume that g_l for $l = 1, \ldots, L$ involves at least one of the variables U_iX, $V_j\bar{X}$ while $g_{L+1}, \ldots, g_M$ are intersections of the Y_k only, $M \geq L$. Let G be g considered as a function of X only; G has a corresponding representation

$$G = G_1 \cup \cdots \cup G_L \cup G_{L+1} \cup \cdots \cup G_M$$

and $G_{L+1} \cup \cdots \cup G_M$ is a constant therefore contained in any possible solution of $X = G(X)$.

We will assume that there exist two different solutions, i.e.

$$A = G(A), \quad B = G(B), \quad \text{and} \quad A \neq B.$$

Let w be a shortest word in $A \Delta B$ $(= (A \cap \bar{B}) \cup (B \cap \bar{A}))$; without loss of generality $w \in A - B (= A \cap \bar{B})$. Clearly $w \in A - B$ implies

$$w \in G_1(A) \cup \cdots \cup G_L(A).$$

Observe that for all $l = 1, \dots, L$, G_l is an intersection of terms of the following four forms (in addition to Y_k's):

$$\text{(a) } UX, \qquad \text{(b) } V\bar{X}, \qquad \text{(c) } \overline{UX}, \qquad \text{(d) } \overline{V\bar{X}}$$

In all four cases we will show that w is in a term with B substituted for X if it is in some term with A substituted for X, e.g. $w \in \overline{V\bar{A}}$ implies $w \in \overline{V\bar{B}}$. Clearly, this will yield a contradiction to the assumption $A \,\Delta\, B \neq \emptyset$ and therefore will prove uniqueness of a solution if one exists.

(a) Let $w \in UA$, i.e. $w = uw'$ for $u \in U$, $w' \in A$, and $|w'| < |w|$ since $\lambda \notin U$. Therefore $w' \in B$ (recall that w is a shortest word in $A \,\Delta\, B$) and $w = uw' \in UB$.

(b) Let $w \in V\bar{A}$, i.e. $w = vw'$ for $v \in V$, $w' \in \bar{A}$, $|w'| < |w|$. By the same argument as in (a) $w' \in \bar{B}$ (otherwise $w' \in A \,\Delta\, B$) therefore $w = vw' \in V\bar{B}$.

(c) Let $w \in \overline{UA}$. This is equivalent to $w \neq uw'$ for all $u \in U$, $w' \in A$, $|w'| < |w|$. Now by our assumption that w is a shortest word in $A \,\Delta\, B$ this condition is equivalent to $w \neq uw'$ for all $u \in U$, $w' \in B$ such that $|w'| < |w|$, therefore $w \in \overline{UB}$.

(d) Let $w \in \overline{V\bar{A}}$. This is equivalent to $w \neq vw'$ for all $v \in V$, $w' \in \bar{A}$, $|w'| < |w|$. Again this holds iff $w \neq vw'$ for all $v \in V$, $w' \in \bar{B}$ such that $|w'| < |w|$ and therefore $w \in \overline{V\bar{B}}$.

Now we know that $w \in G_l(A)$ iff $w \in G_l(B)$ since this holds for all the terms of the intersection hence we have

$$w \in G(A) \quad \text{iff} \quad w \in G(B)$$

contradicting the assumption $A \neq B$. Thus $A = B$ and any solution of $X = G(X)$ is unique.

The reader should note that no properties of the U_i, V_j, Y_k were used in the proof other than $\lambda \notin U_i$ and $\lambda \notin V_j$ for all i, j. That the condition $\lambda \notin U_i$ is necessary is obvious from the example in the introduction. That the condition $\lambda \notin V_j$ cannot be dropped follows from the following example. Consider $X = \overline{V\bar{X}}$, where $\lambda \in V$, say $V = V' \cup \{\lambda\}$, $\lambda \notin V'$. Then we have $X = \overline{V'\bar{X} \cup \bar{X}} = \overline{V'\bar{X}} \cap X$ and clearly $X = A^*$ and $X = \emptyset$ are two different solutions.

4. Existence and a test for membership

In the last section we considered the equation $X = G(X)$ when the function G was expressible as $g(U_1X, \dots, U_IX; V_1\bar{X}, \dots, V_J\bar{X}; Y_1, \dots, Y_K)$, the U_i, V_j, Y_k being arbitrary languages subject to the assumptions $\lambda \notin U_i$, $\lambda \notin V_j$. In this section we will show that under the same assumptions there always exists a solution in X of the equation $X = G(X)$. Furthermore we will show that this solution is recursive if the fixed languages U_i, V_j, Y_k are recursive, i.e. we can test whether a given word $w \in A^*$ is in the solution or not if we have a test for membership for all fixed languages.

We already noted that in the case of equations of the type $X = UX \cup Y$ allowing $\lambda \in U$ does not affect the existence, only the uniqueness of solutions. While this statement holds true for our more general equations as far as uniqueness is concerned for existence it changes quite unexpectedly in that there might not exist any solution if we allow $\lambda \in V_j$, as illustrated by the following example: Consider the equation $X = UX \cup V\bar{X}$, where $\lambda \notin U$, $\lambda \in V$. This implies $X = UX \cup (V-\lambda)\bar{X} \cup \bar{X}$, or $X \supseteq \bar{X}$. This is possible only if $\bar{X} = \emptyset$, i.e. $X = A^*$. However $A^* \neq UA^*$ since $\lambda \notin U$. Therefore there does not exist a language satisfying this equation. However if we assume $\lambda \notin U_i$, $\lambda \notin V_j$ for all i, j there always exists a solution as the following theorem shows. Clearly, this solution is unique by Theorem 1.

Theorem 2. *Let U_i, V_j, Y_k be arbitrary languages, $\lambda \notin U_i$, $\lambda \notin V_j$, and let g be an $(I+J+K)$-ary set-function. Then there always exists a solution of the equation in X*

$$X = g(U_1X, \ldots, U_IX;\ V_1\bar{X}, \ldots, V_J\bar{X};\ Y_1, \ldots, Y_K). \tag{4.1}$$

Proof. The function $g(U_iX;\ V_j\bar{X};\ Y_k)$ can be considered to be represented as a certain regular expression with boolean operators and concatenation operating on the U_i, V_j, Y_k, and X. Let

$$\gamma(U_1, \ldots, U_I;\ V_1, \ldots, V_J;\ Y_1, \ldots, Y_K; X) \tag{4.2}$$

(or γ for short) be the regular expression in these $I+J+K+1$ letters representing g. (If there are more such expressions fix one for the remainder of this paper.) Now define S_r for all $r \geqslant 0$ as follows:

$$S_r = [\gamma(U_i;\ V_j;\ Y_k; X)]^r, \tag{4.3}$$

where $[\alpha]^r$ for any regular expression α of this kind[1] is determined as follows:

$$\begin{array}{ll} \text{(a)} & [\alpha \cup \beta]^r = [\alpha]^r \cup [\beta]^r, \\ \text{(b)} & [\bar{\alpha}]^r = A^r - [\alpha]^r, \\ \text{(c)} & [\alpha \cdot \beta]^r = \bigcup_{s+t=r} [\alpha]^s \cdot [\beta]^t, \\ \text{(d)} & [\alpha]^r = \alpha \cap A^r \quad \text{for} \quad \alpha \in \{U_i, V_j, Y_k \mid i, j, k\}, \\ \text{(e)} & [X]^r = S_r. \end{array} \tag{4.4}$$

Since boolean operations can always be expressed in terms of unions and complements, rules (a) and (b) cover all boolean operations. Rule (c) covers concatenation, and rules (d) and (e) describe what happens with the constants. This definition appears to be circular since in the definiton of S_r the term S_r (by rule (e)) may be used. Note, however, that we assume $\lambda \notin U_i$, $\lambda \notin V_j$ thus whenever the term S_r occurs in the computation of S_r, it is concatenated to the empty language $\emptyset$. Alternatively, one could redefine rule (c) in (3.4) to read

$$\text{(c}'\text{)} \quad [\alpha \cdot \beta]^r = \bigcup_{\substack{s+t=r \\ s>0}} [\alpha]^s \cdot [\beta]^t.$$

[1] In particular, the star operator is not permitted in these expressions.

The reader can verify that S_r contains only words of length r.

Now let

$$S = \bigcup_{r \geq 0} S_r.$$

We claim that S is a solution of equation (4.1), i.e.

$$S = G(S),$$

where G is g as function of X only. Let us assume the contrary, $S \neq G(S)$, and let w be a shortest word in $S \,\Delta\, G(S)$; $|w| = l$. Clearly, $w \in G(S)$ implies $w \in G(S) \cap A^l$. It follows from (4.4) that steps (a) through (e) precisely reflect intersecting $G(S)$ with A^l; for one easily verifies

$$(L_1 \cup L_2) \cap A^l = (L_1 \cap A^l) \cup (L_2 \cap A^l),$$

$$\bar{L} \cap A^l = A^l - (L \cap A^l),$$

$$(L_1 \cdot L_2) \cap A^l = \bigcup_{s+t=l} (L_1 \cap A^s) \cdot (L_2 \cap A^t).$$

Since $|w|$ is minimal it follows that no such w can exist, $S \,\Delta\, G(S)$ is empty, and hence

$$S = G(S).$$

While for arbitrary languages L it is not possible to determine the intersections $L \cap A^r$, for recursive languages this can be done as there are only finitely many words of length r in A^*. Therefore we can state

Theorem 3. *Let U_i, V_j, Y_k be recursive languages, $\lambda \notin U_i$, $\lambda \notin V_j$, and let g be an $(I+J+K)$-ary set-function. Then the unique solution in X of the equation*

$$X = g(U_1X, \ldots, U_IX;\ V_1\bar{X}, \ldots, V_J\bar{X};\ Y_1, \ldots, Y_k)$$

is recursive.

As mentioned in the introduction all the results given for one equation in one variable can easily be generalized to n equations in n variables.

Example 2. Consider the following equation over the alphabet $A = \{a, b\}$:

$$X = UX \cap \overline{V\bar{X} \cap Y}, \tag{4.5}$$

where $U = \{a^p \mid p \text{ prime}\}$, $V = \{a^n b^{n-1} \mid n \geq 1\}$, and $Y = (AA)^*$.

We first rewrite (4.5)

$$X = \overline{\overline{UX} \cup \overline{\overline{V\bar{X}} \cup \bar{Y}}}$$

and then compute S_r for $r \geqslant 0$:

$$S_0 = \lambda - ([\overline{UX}]^0 \cup [\overline{\overline{V\bar{X}} \cup \bar{Y}}]^0)$$

$$= \lambda - ((\lambda - [UX]^0) \cup (\lambda - ((\lambda - [V\bar{X}]^0) \cup (\lambda - [Y]^0)))) = \emptyset$$

since $[UX]^0 = \emptyset$ and $[V\bar{X}]^0 = [V]^0 \cdot [\bar{X}]^0 = \emptyset$,

$$S_1 = A - ((A - [UX]^1) \cup (A - ((A - [VX]^1) \cup (A - [Y]^1)))) = \emptyset$$

since

$$[UX]^1 = [U]^0 \cdot [X]^1 \cup [U]^1 \cdot [X]^0 = \emptyset \cdot [X]^1 \cup \emptyset \cdot S_0 = \emptyset$$

while

$$[V\bar{X}]^1 = [V]^0 \cdot [\bar{X}]^1 \cup [V]^1 \cdot [\bar{X}]^0$$

$$= \emptyset \cup a \cdot (\lambda - [X]^0) = a \cdot (\lambda - S_0) = a \cdot (\lambda - \emptyset) = a.$$

One can in fact show that $S_i = \emptyset$ for all $i \geqslant 0$, thus $S = \emptyset$. That the empty set is indeed a solution is easily verified by substitution, $U \cdot \emptyset \cap \overline{VA^* \cap Y} = \emptyset$. By Theorem 1, this solution is the only one.

5. Construction of finite automata when the fixed languages are regular

In the preceding two sections we showed existence and uniqueness of a solution of the equation

$$X = g(U_iX;\ V_j\bar{X};\ Y_k) \tag{5.1}$$

under the assumption that $\lambda \notin U_i$, $\lambda \notin V_j$ for all i, j. In this section we will further assume that all the fixed languages U_i, V_j, Y_k are regular. Under these assumptions we show that the unique solution of (5.1) is regular by effectively constructing a deterministic finite automaton accepting it.

We start with the construction. Let

$$\boldsymbol{A}_{U_i} = (A, Q_{U_i}, M_{U_i}, q_0, F_{U_i}), \qquad \boldsymbol{A}_{V_j} = (A, Q_{V_j}, M_{V_j}, q_0, F_{V_j}) \quad \text{and}$$

$$\boldsymbol{A}_{Y_k} = (A, Q_{Y_k}, M_{Y_k}, q_0, F_{Y_k})$$

be nonreturning deterministic finite automata such that $U_i = L(\boldsymbol{A}_{U_i})$, $V_j = L(\boldsymbol{A}_{V_j})$, $Y_k = L(\boldsymbol{A}_{Y_k})$ for all $i = 1, \ldots, I$, $j = 1, \ldots, J$, $k = 1, \ldots, K$. Furthermore assume that $Q_{U_i} - \{q_0\}, \ldots, Q_{Y_K} - \{q_0\}$ are all pairwise disjoint, i.e. only the initial state q_0 is shared. Then we define a boolean automaton $\boldsymbol{B} = (A, Q, M, q_0, F)$ as follows:

$$Q = \bigcup_{i=1}^{I} Q_{U_i} \cup \bigcup_{j=1}^{J} Q_{V_j} \cup \bigcup_{k=1}^{K} Q_{Y_k},$$

$$F = \begin{cases} \bigcup_{k=1}^{K} F_{Y_k} - \{q_0\}, & \text{if } \lambda \notin G(X), \\ \bigcup_{k=1}^{K} F_{Y_k} \cup \{q_0\}, & \text{if } \lambda \in G(X), \end{cases}$$

where $G(X)$ is $g(U_iX; V_j\bar{X}; Y_k)$ as a function of X only. As already indicated we can always test whether $G(X)$ contains λ or not without knowing the solution as $\lambda \in G(X)$ iff $[\gamma]^0 = \{\lambda\}$. Finally

$$M(q, a) = \begin{cases} g(N(M_{U_i}(q_0, a)); N'(M_{V_j}(q_0, a)); M_{Y_k}(q_0, a)), & \text{if } q = q_0, \\ N(M_{U_i}(q, a)), & \text{if } q \in Q_{U_i} - \{q_0\} \text{ for some } i, \\ N'(M_{V_j}(q, a)), & \text{if } q \in Q_{V_j} - \{q_0\} \text{ for some } j, \\ M_{Y_k}(q, a), & \text{if } q \in Q_{Y_k} - \{q_0\} \text{ for some } k \end{cases}$$

and N, N' are as follows:

$$N(q) = \begin{cases} q \cup q_0, & \text{if } q \in F_{U_i} \text{ for some } i, \\ q, & \text{otherwise,} \end{cases}$$

$$N'(q) = \begin{cases} q \cup \bar{q}_0 & \text{if } q \in F_{V_j} \text{ for some } j, \\ q & \text{otherwise.} \end{cases}$$

Then $L(\boldsymbol{B})$ is the unique solution of (5.1). Since the language accepted by a boolean automaton is always regular, this solution is regular, too. Furthermore $\boldsymbol{A_B}$, the derived deterministic automaton to $\boldsymbol{B}$, is effectively constructible and also accepts the solution.

Before we prove that $L(\boldsymbol{B})$ actually is a solution for X of (4.1) we give two examples.

Example 3. Consider the equation $X = UX \cup V\bar{X}$, where $U = a^*b((a \cup b)a^*b)^*$, $V = (a \cup b)\,(b^*a(a \cup b))^*b^*$. The automata $\boldsymbol{A}_U$ and $\boldsymbol{A}_V$ are given below.

$$\boldsymbol{A}_U = (\{a, b\}, \{X, B, C\}, M_U, X, \{C\})$$

$$\boldsymbol{A}_V = (\{a, b\}, \{X, D, E\}, M_V, X, \{D\})$$

M_U:

	a	b
X	B	C
B	B	C
C	B	B

M_V:

	a	b
X	D	D
D	E	D
E	D	D

Thus $\boldsymbol{B}$ is as follows:

$$\boldsymbol{B} = (\{a, b\}, \{X, B, C, D, E\}, M, X, \emptyset),$$

M given by

	a	b
X	$B \cup D \cup \bar{X}$	$C \cup X \cup D \cup \bar{X}$
B	B	$C \cup X$
C	B	B
D	E	$D \cup \bar{X}$
E	$D \cup \bar{X}$	$D \cup \bar{X}$

Therefore the derived deterministic automaton A_B is constructed as follows:

	a	b	$=_\emptyset$
X	$B \cup D \cup \bar{X}$	$C \cup D \cup X \cup \bar{X} \equiv 1$	0
$B \cup D \cup \bar{X}$	$B \cup E \cup \overline{B \cup D \cup \bar{X}}$	1	1
1	1	1	1
$B \cup E \cup \overline{B \cup D \cup \bar{X}}$	$B \cup D \cup \bar{X}$	1	0

Therefore $X = (aa)^*(a \cup (a \cup \lambda)bA^*)$ is a solution. This can also be verified by substituting this language into the given equation. By Theorem 1 this solution is the only one.

Example 4. Consider the equation

$$X = [(\overline{V\bar{X}} - \bar{Y}_1) \cap \overline{UX}] \cup Y_2,$$

where

$$U = bb^*,\ V = a,\ Y_1 = bb^*,\ Y_2 = \lambda.$$

The corresponding boolean automaton is as follows:

$$\boldsymbol{B} = (\{a, b\}, \{X, B, C, D\}, M, \{X, D\}),$$

M being given by

	a	b
X	0	$\bar{B} \cap D \cap \bar{X}$
B	0	$B \cup X$
C	0	0
D	0	D

Hence A_B is as follows:

	a	b	$=_F$
X	0	$\bar{B} \cap D \cap \bar{X}$	1
0	0	0	0
$\bar{B} \cap D \cap \bar{X}$	0	$(\overline{B \cup X}) \cap D \cap (B \cup X) \equiv 0$	0

Thus $L(\boldsymbol{B}) = \{\lambda\}$. One can directly verify that this is indeed a solution:

$$[(\overline{a\bar{\lambda}} - \overline{bb^*}) \cap \overline{bb^*}] \cup \lambda = \overline{a\bar{\lambda}} \cap bb^* \cap \overline{bb^*} \cup \lambda = \lambda.$$

We now come to the proof that $L(\boldsymbol{B})$, or L for short, as given in the general constructions is a solution of (5.1). As in the proof of Theorem 1 we represent G as union of intersections of terms of the four types, U_iX, $\overline{U_iX}$, $V_j\bar{X}$, and $\overline{V_j\bar{X}}$, i.e.

$$G(L) = G_1(L) \cup \cdots \cup G_m(L) \cup \cdots \cup G_n(L),$$

where $G_{m+1}(X) \cup \cdots \cup G_n(X)$ does not depend on X. We claim

$$G(L) = L. \tag{5.2}$$

(a) We first prove $L \supseteq G(L)$. It is not difficult to see that by construction of $\boldsymbol{B}$, L contains $G_{m+1}(L) \cup \cdots \cup G_n(L)$. Hence let us assume $w \in G_l(L)$, i.e. there exist sets $I_1, I_2 \subseteq \{1, \ldots, I\}$, $J_1, J_2 \subseteq \{1, \ldots, J\}$, and $K_1, K_2 \subseteq \{1, \ldots, K\}$ such that

$$w \in U_iX \quad \text{for all } i \in I_1 \quad \text{and} \quad w \in \overline{U_iX} \quad \text{for all } i \in I_2$$

and

$$w \in V_j\bar{X} \quad \text{for all } j \in J_1 \quad \text{and} \quad w \in \overline{V_j\bar{X}} \quad \text{for all } j \in J_2$$

and

$$w \in Y_k \quad \text{for all } k \in K_1 \quad \text{and} \quad w \in \bar{Y}_k \quad \text{for all } k \in K_2.$$

Without loss of generality we may assume that g in the definition of the transition function M of $\boldsymbol{B}$ is represented such that it exactly mirrors the representation of G. By the definition of M it is easy to see that whenever $u \in U_i$ a transition to X is provided and whenever $v \in V_j$ a transition to $\bar{X}$ is provided. Consequently if w is as described above then due to the conditions $\lambda \notin U_i$, $\lambda \notin V_j$, the problem whether w is in L can be reduced to the same problem for a shorter word. Hence, by induction (the basis being L contains $G_{m+1}(L) \cup \cdots \cup G_n(L)$), the claim follows.

(b) We now prove $G(L) \supseteq L$. Suppose $w \in L$. If $w = \lambda$, then by definition of $\boldsymbol{B}$, $\lambda \in G_{m+1}(L) \cup \cdots \cup G_n(L)$. Therefore assume $|w| > 0$, i.e. $w = aw'$ with $a \in A$ and consider $M(q_0, w)$. $M(q_0, a)$ can again be represented as a union of intersections such that the representation of $M(q_0, a)$ and G are identical except that G has U_iX, $V_j\bar{X}$, or Y_k whenever $M(q_0, a)$ has $N(M_{U_i}(q_0, a))$, $N'(M_{V_j}(q_0, a))$, or $M_{Y_k}(q_0, a)$. Now the argument is analogous to that in (a). Thus $w \in G(L)$.

This concludes the proof of the claim (5.2). Therefore we can summarize.

Theorem 4. *Let U_i, V_j, Y_k be regular languages, $\lambda \notin U_i$, $\lambda \notin V_j$, and let g be an $(I+J+K)$-ary set-function. There is an effective construction of a deterministic automaton $\boldsymbol{A}$ accepting precisely the unique solution of the equation in X*

$$X = g(U_1X, \ldots, U_IX;\ V_1\bar{X}, \ldots, V_J\bar{X};\ Y_1, \ldots Y_K).$$

The construction gives rise to a corollary.

Corollary. *If V is prefix free, i.e. $x \in V$ and $xy \in V$ imply $y = \lambda$, then the solution of*

$$X = V\bar{X}$$

is given by $X = (VV)^* V(\mathrm{PPF}(V) \cup \mathrm{PD}(V))$, *where*

$$\mathrm{PPF}(V) = \{x \in A^* \mid xy \in V \text{ for some } y \in A^* \text{ and } x \notin V\}$$

and

$$\mathrm{PD}(V) = \{x \in A^* \mid xy \notin V \text{ for all } y \in A^* \text{ and for all } x' \in \mathrm{PPF}(x),\ x' \notin V\}.$$

$\mathrm{PPF}(V)$ is the set of all prefixes of some $v \in V$ which are not in V, and $\mathrm{PD}(V)$ can be visualized as follows: Let $\boldsymbol{A} = (A, Q, M, q_0, F)$ be the reduced automaton such that $L(\boldsymbol{A}) = V$; $\mathrm{PD}(V)$ is the set of all words x which lead to the 'dead' state of $\boldsymbol{A}$ (the state $q \notin F$ with $M(q, a) = q$ for all $a \in A$) such that no prefix of x is in V. Note that V prefix free implies that there is always a dead state in $\boldsymbol{A}$ and furthermore there is precisely one final state from which all transitions lead into the dead state. Therefore in the corresponding boolean automaton for the solution of the equation, $N'(q)$ can be written either as q or as $\overline{q_0}$ depending on whether q is a rejecting or the unique final state of $\boldsymbol{A}$. This yields a special form for $\boldsymbol{A}_B$ which gives rise to the above representation.

Similar but increasingly more complex expressions can be obtained for more complicated equations.

We conclude with some remarks on generalizations. While it was not mentioned in the construction one can easily modify it in order to obtain solutions for equations involving concatenation from the left such as $X = V_1\overline{V_2\bar{X}}$. An example is given below (Example 5). Note that within the framework of the construction concatenation from the right is out of the question since the result need not be regular any more. Consider for instance $X = 0X1 \cup \lambda$. It is well known that $\{0^n 1^n \mid n \geq 0\}$ is a solution, and it is not difficult to see that it is the only one. Thus this equation has no regular solution.

In the preceding sections we only dealt with concatenation from the left, e.g. UX, $V\bar{X}$, generalizing derivative equations. We can, however, handle concatenation from the right, too, provided no terms involving concatenation from the left appear in the equation. This is done by first reversing the equations, thus obtaining equations as considered here, determining their solution, and reversing this solution to obtain a solution of the original equations (see Example 6). Theorems 1 and 2 also hold for this modification (in the proofs we have to replace UX, $V\bar{X}$ by XU, $\bar{X}V$).

Example 5. We determine the solution of the equation $X = V_1\overline{V_2\bar{X}}$, where $V_1 = (a \cup b^*)a$, $V_2 = aa^*$. First we determine the nonreturning automata $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ such that $L(\boldsymbol{A}_i) = V_i$, $i = 1, 2$.

$\boldsymbol{A}_1 = (\{a, b\}, \{X, B, C, D\}, M_1, X, \{B, D\})$, $\boldsymbol{A}_2 = (\{a, b\}, \{E, G\}, M_2, E, \{G\})$ where M_1 and M_2 are as follows:

M_1:

	a	b
X	B	C
B	D	$\emptyset$
C	D	C
D	$\emptyset$	$\emptyset$

M_2:

	a	b
E	G	$\emptyset$
G	G	$\emptyset$

Therefore $B = (\{a, b\}, \{X, B, C, D, E, G\}, M, X, \{B, D\})$, M being given by

	a	b
X	B	C
B	$D \cup \overline{G \cup \bar{X}}$	$0 \cup \bar{0}$
C	D	C
D	$0 \cup \overline{G \cup \bar{X}}$	$0 \cup \bar{0}$
G	$G \cup \bar{X}$	0

Note that B and D remain accepting since $\lambda \in \overline{aa^* \bar{X}}$ while G becomes rejecting according to the general construction. Furthermore E can be dropped since it is not used in the final result. Thus the derived deterministic automaton A_B is as follows:

	a	b	$=_F$
X	B	C	0
B	$D \cup \overline{G \cup \bar{X}}$	1	1
C	D	C	0
$D \cup \overline{G \cup \bar{X}}$	$\overline{G \cup \bar{X}}$	1	1
1	1	1	1
D	$\overline{G \cup \bar{X}}$	1	1
$\overline{G \cup \bar{X}}$	$\overline{G \cup \bar{X} \cup \bar{B}}$	C	0
$\overline{G \cup \bar{X} \cup \bar{B}}$	$\overline{G \cup \bar{X} \cup \bar{B}}$	C	0

By substitution one can also directly verify that $L(A_B)$ is, in fact, a solution of the given equation.

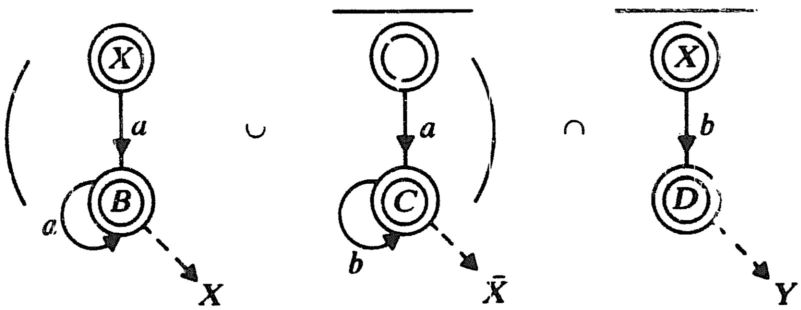
Example 6. Consider the equation $X = \overline{\bar{X}bb}$ over $A = \{a, b\}$. In order to determine its solution in X we reverse it, obtaining $Y = \overline{bb\bar{Y}}$, and solve it in the usual way. We find as solution $Y = (bb)^*b(\lambda \cup aA^*)$. Now we reverse this solution, which yields $(A^*a \cup \lambda)b(bb)^*$, and one easily verifies that this is a solution for the given equation.

Example 7. We determine the solution in X for the following set of equations:

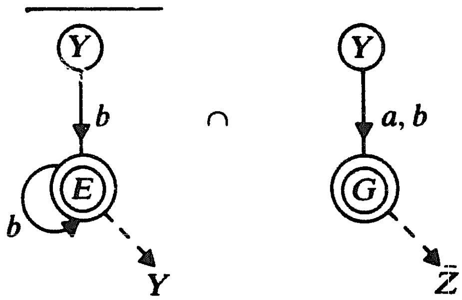
$$X = (aa^*X \cup \overline{ab^*\bar{X}}) \cap \overline{bY}, \qquad Y = \overline{bb^*\bar{Y}} \cap (a \cup b)\bar{Z}, \qquad Z = a^*\bar{Z} \cup \overline{b\bar{X}}$$

Clearly $\lambda \in X$, $\lambda \notin Y$, $\lambda \in Z$ by inspection. For simplicity we use a graphical representation; empty states are not shown.

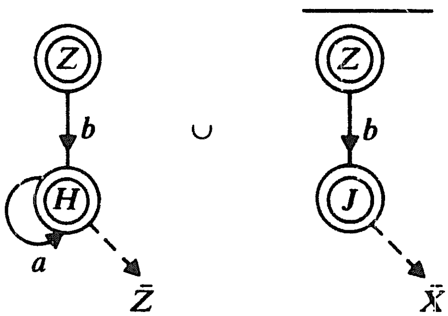
X:



Y:



Z:



Combining all these transitions yields the following boolean automaton $B' = (\{a, b\}, \{X, B, C, D, Y, E, G, Z, H, J\}, M', X, \{X, Z\})$, M' given by

	a	b
X	$(B \cup X \cup \overline{C \cup \bar{X}}) \cap \bar{0}$	$(0 \cup \bar{0}) \cap \overline{D \cup Y}$
B	$B \cup X$	0
C	0	$C \cup \bar{X}$
D	0	0
Y	$\bar{0} \cap (G \cup \bar{Z})$	$\overline{E \cup Y} \cap (G \cup \bar{Z})$
E	0	$E \cup Y$
G	0	0
Z	$0 \cup \bar{0}$	$H \cup \bar{Z} \cup \overline{J \cup \bar{X}}$
H	$H \cup \bar{Z}$	0
J	0	0

which can be simplified to $\boldsymbol{B} = (\{a, b\}, \{X, B, Y, E, Z, H\}, M, X, \{X, Z\})$, M being

	a	b
X	$B\cup X$	$\bar{Y}$
B	$B\cup X$	0
Y	$\bar{Z}$	$\overline{E\cup Y}\cap\bar{Z}$
E	0	$E\cup Y$
Z	1	$H\cup X\cup\bar{Z}$
H	$H\cup\bar{Z}$	0

Therefore $\boldsymbol{A_B}$ is as follows:

	a	b	$=_F$
X	$B\cup X$	$\bar{Y}$	1
$B\cup X$	$B\cup X$	$\bar{Y}$	1
$\bar{Y}$	Z	$E\cup Y\cup Z$	1
Z	1	$H\cup X\cup\bar{Z}$	1
$E\cup Y\cup Z$	1	$E\cup H\cup X\cup Y\cup\bar{Z}$	1
1	1	1	1
$H\cup X\cup\bar{Z}$	$B\cup H\cup X\cup\bar{Z}$	$\bar{Y}\cup\overline{H\cup X\cup\bar{Z}}$	1
$E\cup H\cup X\cup Y\cup\bar{Z}$	$B\cup H\cup X\cup\bar{Z}$	1	1
$B\cup H\cup X\cup\bar{Z}$	$B\cup H\cup X\cup\bar{Z}$	$\bar{Y}\cup\overline{H\cup X\cup\bar{Z}}$	1
$\bar{Y}\cup\overline{H\cup X\cup\bar{Z}}$	Z	$E\cup Y\cup Z$	1

Thus $X = A^*$, furthermore $Z = A^*$ since it occurs as state in $\boldsymbol{A_B}$, and finally $Y = \emptyset$ since $\bar{Y}$ occurs as state in $\boldsymbol{A_B}$. Since the solution is very simple one can easily verify the result by substitution:

$$X\colon\ (aa^*A^*\cup\overline{ab^*\overline{A^*}})\cap\overline{b\emptyset} = A^*, \qquad Y\colon\ \overline{bb^*\emptyset}\cap(a\cup b)\overline{A^*} = \emptyset,$$

$$Z\colon\ ba^*\emptyset\cup\overline{b\bar{A}^*} = A^*.$$

References

[1] J.A. Brzozowski, Derivations of regular expressions, *JACM* **11** (1964) 481–494.

[2] J.A. Brzozowski and E. Leiss, On equations for regular languages, finite automata, and sequential networks, *Theoret. Comput. Sci.* **10** (1980) 19–35.

[3] J.H. Conway, *Regular Algebra and Finite Machines* (Chapman & Hall, London, 1971).

[4] S. Eilenberg, *Automata, Languages, and Machines, Vol. A* (Academic Press, New York, 1974).

[5] A. Salomaa, *Theory of Automata* (Pergamon, Oxford, 1969).