SETS OF SET-EQUATIONS EQUIVALENT TO CONTEXT-FREE GRAMMARS AND THEIR SOLUTION IN SOME CASES

By: Marvin C. Paull

WORK DONE UNDER:

Contract No. AF 1863

Air Force Office of Scientific Research
Directorate of Mathematical and
Information Sciences
Arlington, Virginia 22209

June, 1971

DCS Technical Report #8

Department of Computer Science
Livingston College
Rutgers University

Sets of Set-Equations Equivalent to Context-Free Grammars and Their Solutions in Some Cases

Given a context free grammar* G, we can obtain a set of set equations, E, which is equivalent to G in the sense that the set of strings which can be derived from a variable X in G, is the same as the value of X in E which is required to satisfy the set of equations E.

For a given grammar G, the equivalent set of equations E is easily obtained. For the variable X in G let:

$$X \to \alpha_1$$
$$X \to \alpha_2$$
$$\vdots$$
$$X \to \alpha_n$$

be all rules with X on the left.

Then in E we will have the equation:

$$X = \alpha_1 \cup \alpha_2 \cup \ \cdots \ \alpha_n$$

In this way we obtain one equation in E for each variable in G. All and only those equations obtained as above are included in E.

E then is a set of equations in variables (same as those of G), constants (= to the terminal symbols of G) and operations indicated by concatenation, and by $\cup$. The following gives the interpretation of each of these in E.

(1) A variable is a variable whose value is a set of strings

(2) A constant, say x, is the set consisting of the single element x.

(3) Concatenation or juxtaposition, say XY, is interpreted as the set of all strings xy where $x \in X$, $y \in Y$.

(4) $\cup$ is interpreted as set union.

---

*Ref 3 contains the basic definitions and information about context-free grammars.

Because of the properties of these operations we can do certain manipulations on the equations of E without changing their solution. These include factoring;

ex. $XY \cup XZ = X(Y \cup Z)$,

and substituting; ex., if $X = \alpha$, $Y = \gamma X \delta$ then we can conclude that $Y = \gamma \alpha \delta$. These kinds of manipulations will be used in what follows.

Before considering the solution of sets of set-equations, we will establish the basic proposition that a set of set-equations E obtained as above from a context-free grammar G is in fact equivalent to that grammar in the sense described above.

Consider a context-free grammar, G, in which from each variable some terminal string is derivable, and in which for no variable X is there a derivation $X \Rightarrow X$, and also consider the set of set equations E obtained from G as above.

Theorem:
(1) If the terminal string, s, can be derived in G from variable X, then it can be proved in E to be a member of the set X; and (2) If a string s of length ≥ 1 can be proved to be in X of E, then it must be derivable from X in G.

Proof: First consider (1), suppose there is a derivation of a string s in G from X:

$$X \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \alpha_n = s$$

Then in E we can surely show that:

$X \supseteq \alpha_1$, $\alpha_1 \supseteq \alpha_2$, $\cdots$ $\alpha_{n-1} \supseteq \alpha_n$ where $\supseteq$ is set inclusion. We can prove this because if $X \Rightarrow \alpha_1$ in G then $X = \gamma_1 \cup \alpha_1 \cup \gamma_2$ in E, or $X \supseteq \alpha_1$. And if $\alpha_j = \alpha_{j_1} Z \alpha_{j_2} \Rightarrow \alpha_{j_1} \beta \alpha_{j_2} = \alpha_{j+1}$ in G, then $Z = \omega_1 \cup \beta \cup \omega_2$ in E, and therefore $Z \supseteq \beta$ in E, and therefore $\alpha_j = \alpha_{j_1} Z \alpha_{j_2} = \alpha_{j_1} \beta \alpha_{j_2} = \alpha_{j+1}$ in E.

Next consider (2). What we can show here is that if a string s cannot be both non-derivable in G, and at the same time proveable in X of E. From this (2) would follow. Hypothesis: Suppose that s is not derivable from X in G but is in X of E, and suppose s is the smallest such string. If $X = \alpha_1 \cup \alpha_2 \cup \ldots \cup \alpha_n$ then since $s \in X$; $s \in \alpha_s$. If α_s is a single variable, then consider the equation $\alpha_s = \beta_1 \cup \beta_2 \cup \ldots \cup \beta_q$. s must be a member of β_j for $1 \leq j \leq q$, say β_s. Again β_s may be a single variable, but by continuing this process long enough (for more steps than there are variables in G) we can eventually show that $s \in \delta_s$ where δ_s is (a) all terminals, or (b) contains at least two symbols, one of which is a terminal, or (c) contains at least two symbols all of which are variables and two of these variables do not derive the 0 length string, this must eventually happen because otherwise, using the fact that each variable derives some string, we could show that the grammar G we started with did have a derivation $X \overset{*}{\Rightarrow} X$.

Case (a) cannot occur because if the δ_s is a terminal string and = s, we could show how to derive s from X in G, contrary to the hypothesis.

Case (b) cannot occur because;

if δ_s contains terminals and variables, the terminals must match parts of s, and so it must be possible to prove that the remaining sub-parts of s are proveably in the sets which satisfy the variables in δ_s, but still not derivable from these variables in G. (If they were derivable in G then s would be deriveable in G contrary to hypotheses). But s was by hypothesis the smallest such terminal string. So again, the hypothesis is contradicted.

Finally Case (c) cannot occur because if δ_s consists solely of variables, two of which derive ε in G. Therefore, since the length of $s \geq 1$ s is not ε, it must be that we cannot prove in E that ε is a member

of either of these variables. Thus each of these sets must proveably contain a subpart of s, which is not deriveable from that variable in G, but since s was supposed to be the smallest such terminal string, we have a contradiction.

The theorem as it stands may not be quite satisfactory because of the 's of length ≥ 1.' The 'of length ≥ 1' could be removed if we insisted that G be such that every derivation in G eventually lead to a phrase containing at least one terminal symbol. So that for example we would disallow following grammar

$$\begin{aligned} & S \rightarrow a \\ G: \quad & S \rightarrow aS \\ & S \rightarrow SS \end{aligned}$$

which does not derive ε from S, but has a corresponding set of set-equations, namely:

$$S = a \cup aS \cup SS$$

in which s is satisfied by the set:

{a,aa,aaa,...} as well as by the set:

{ε,a,aa,aaa,....}

In the next section we shall discuss the solution of a set of set-equations in two cases. A solution is a description of the sets which satisfy the set equations. The first case is for regular or finite state grammars. This case has been previously considered-first in reference 1. The second case is for linear grammars.

Solutions of sets of set-equations

Finite State Grammar*

Definition: A CFG is finite state if all its rules are of the form:

<non-terminal symbol> → terminal symbol string <non-terminal symbol>

Ex.
$A \rightarrow eA$
$A \rightarrow aA$
$A \rightarrow cB$
$B \rightarrow bB$
$B \rightarrow b$
$B \rightarrow dA$

The associated set of set equations is of the form:

$$A_1 = t_{11}A_1 \cup t_{12}A_2 \cup t_{13}A_3 \cdots t_{1n}A_n \cup t_1$$

$$A_2 = t_{21}A_1 \cup t_{22}A_2 \cup t_{23}A_3 \cdots t_{2n}A_n \cup t_2$$

$$\vdots$$

$$A_n = t_{n1}A_1 \cup t_{n2}A_2 \cup t_{n3}A_3 \cdots t_{nn}A_n \cup t_n$$

where any number of the terms $t_i A_j$, or t_j may be absent and where A_j is a set variable, t_{ij} and t_j are in general expressions involving $\cup$, concatenation, * and terminal constants (initially only a union of terminals).

Ex. (1) $A = (a \cup e)A \cup cB$

$B = dA \cup bB \cup b$

Such a finite state set of equations can always be solved in a manner analogous to that used for linear equations.

First note that if X, Z and A are any sets:

S1 $A = X^*Z$ is a solution for:

$A = XA \cup Z$

because by substitution:

$X^*Z = X(X^*Z) \cup Z$

or

$Z \cup X^1Z \cup X^2Z \cup X^3Z \cdots = X(Z \cup X^1Z \cup X^2Z \cup \cdots) \cup Z$

*This material is in references 1 and 2

Next note that we can put the first (or similarly any) of the equations of our finite state set in the following form as a result of using S1.

$$\textcircled{1}\quad A_1 = t_{11}^*(t_{12}A_2 \cup \ldots\ t_{1n}A_n \cup t_1)$$

$$= t_{11}^*t_{12}A_2 \cup \ldots\ t_{11}^*t_nA_n \cup t_{11}^*t_1$$

This equation has to have A_1 appear on the right. Now if this equation is substituted for each appearance of A_1 in the succeeding equations, we will have in addition to (1) a set of equations with one fewer set variable (A_1 has been eliminated) which is still in the form of the original set.

To see this, note that the second equation will appear as follows after the substitution and rearrangement:

$$A_2 = (t_{22} \cup t_{21}t_{11}^*t_{12})A_2 \cup (t_{23} \cup t_{21}t_{11}^*t_{13})A_3 \cup \ldots$$

$$\cup\ (t_{2n} \cup t_{21}t_{11}^*t_{1n})A_n \cup (t_{21}t_{11}^*x_1 \cup t_2)$$

Now the same process may be repeated to reduce the number of equations. Finally, we will obtain a solution for A_n in terms of *, $\cup$, concatenation and terminal constants. By substitutions in previously obtained solutions, one can obtain the solution for any or all the set variables. If A_n was the distinguished symbol, we might, however, be satisfied with its solution.

Ex.

(2) $A = (a \cup e)^*cB$ by use of S1 and by substitution on (1)

$B = d(a \cup e)^*cB \cup bB \cup b$

$B = (d(a \cup e)^*c \cup b)B \cup b$

thus

(3) $B = (d(a \cup e)^*c \cup b)^*b$ by use of S1, (3) is a solution for B

$A = (a \cup e)^*c(d(a \cup e)^*c \cup b)^*b$ by substitution in (2).

Simple Balanced Grammars

We will now extend this approach to a somewhat more inclusive set of CFG's called simple balanced grammars. Here rules may be of the form allowed for a finite state grammar and in addition, we allow rules of the form:

<non-terminal symbol> → terminal string$_1$ <non-terminal symbol> terminal string$_2$ in which terminal string$_1$ or terminal string$_2$ or both are not o-length.

In general, a term having the associated set of set equations is of the form:

$$A_1 = U_1^{k_{11}} t_{11}A_1s_{11} \cup U_1^{k_{12}} t_{12}^j A_2 s_{12}^j \cup \ldots U_1^{k_{1z}} t_{1n}^j A_n s_{1n}^j \cup t_1$$

$$A_2 = U_1^{k_{21}} t_{12}^f A_1 s_{12} \cup \ldots t_2$$

$$\vdots$$

$$A_n = U_1^{k_{n1}} t_{1n}^f A_{\cdot} t_{1n}^r \cup \ldots U_1^{k_{nz}} t_{nn}^f A_n t_{nn}^r \cup t_n$$

Again, any term on the right may be absent. This set of equations too can always be solved, but to do this we need a new special solution.

If X,A,B and Z are any sets, then:

S2 $A = X^nZY^n$ is a solution of $A = X\ A\ Y \cup Z$, in which $A^nZB^n = \{Z, AZB, AAZBB, \ldots\}$. S2 is in fact a generalization of S1 and may be verified as S1 was. Applying this result to equations of the form that appears above, with n A terms of the form t_jAs_j on the right; t_j and s_j being any terminal expressions, and β being a union of terms, we have in general:

$$A = t_1As_1 \cup t_2As_2 \cup \ldots t_nAs_n \cup \beta$$

We get, applying S2:

$$t_1^{n_1}(t_2As_2 \cup t_3As_3 \cup \ldots t_nAs_n \cup \beta)s_1^{n_1}$$

and "concatenating out" we get:

$$A = t_1^{n_1} t_2 A s_2 s_1^{n_1} \cup t^{n_1} t_3 A s_3 s^{n_1} \cup \ldots t_1^{n_1} t_n A s_n s_1^{n_1} \cup t_1^{n_1} \beta s_1^{n_1}$$

Now if in addition, we concatenated $t_1^{n_1} \beta s_1^{n_1}$ out, we would still have all our terms in balanced form and would have one less term containing an A. We apply an analogous step again to obtain:

$$A = (t_1^{n_1} t_2)^{n_2} (t_1^{n_1} t_3 A s_3 s_1^{n_1} \cup \ldots t_1^{n_1} t_n A s_n s_1^{n_1} \cup t_1^{n_1} \beta s_1^{n_1})(s_2 s_1^{n_1})^{n_2}$$

Again, we can see that after multiplying out, we will reduce the number of terms having A in them by one, and still have all the terms in the balanced form.

Continuing in this way, we could after n steps remove all occurences of A on the right.

This procedure may be applied to one of the equations, say the first, of our set of equations to remove all occurences of A_1 on the right. Then we may substitute this right side for all occurences of A_1 in the remaining equations. After each substitution, one may concatenate out to again obtain a set of equations whose right sides are each unions of balanced terms. But now we have a set of equations with one less set variable (A_1 has been eliminated).

This entire process can be repeated as previously donc in the finite state case until we have a solution for A_n, and can by further substitution obtain a solution for all A_j, $j = 1$ to n.

Ex. (1) $A = aAb \cup cAd \cup e$

$A = a^{n_1}(cAd \cup e)b^{n_1}$ by applying S2 to (1)

$A = a^{n_1} cAdb^{n_1} \cup a^{n_1} eb^{n_1}$

$A = (a^{n_1} c)^{n_2} a^{n_1} eb^{n_1} (db^{n_1})^{n_2}$ by applying S2

Note that if a superscript ^{n} appears on only one term it may be replaced with a *.

Concluding Remarks:

Although the press of time requires us to terminate this report now, the following considerations indicate that further results are available in this line of inquiry.

In the finite state case any language or set of strings which can be described by a regular expression, i.e. finite set of variables X,Y,Z..., the concatenation, *, and ∪ operation; has a corresponding finite state grammar defining that set.

The analogous property does not hold in the balanced grammar case. For example $a^{n}b^{n}c^{n_1}d^{n_1}$ is a language which is defined by the grammar.

$S \rightarrow X\ Y$

$X \rightarrow aXb$

$X \rightarrow ab$

$Y \rightarrow cYd$

$Y \rightarrow cd$

but there is no balanced grammar which defines this language, because any derivation tree in a balanced grammar has only 1 path on which variables appear, and the language $a^{n}b^{n}c^{n_1}d^{n_1}$ requires at least two such paths as realized with a CFG. This can be shown by application of the 'uvwxy' theorem (reference 3 Chapt. 4.)

Because the balanced grammar, unlike the finite state, is not complete in the sense that, for example, the concatenation of languages defined by balanced grammars cannot in general be defined by a balanced grammar, we can define a more general, but not so neatly defined, grammar for which solutions can be obtained.

A 'closed balanced grammar' then might be defined as:

1. A balanced grammar is a closed balanced grammar
2. If B_j is the distinguished symbol of a closed balanced grammar

then any grammar which contains one or more rules of the form $X \rightarrow \tau_1 B_1 B_2 \cdots B_n \ \tau_1$ and τ_2 are terminal strings is a closed balanced grammar.

A solution would exist for such a grammar. Another direction in which further study is warranted is to establish the connection of this approach with Dyck Languages. (see reference 3 pp. 67).

Finally we note that both of the two 'solutions' discussed here hinge on a basic solution to a set equation i.e., S1 and S2 . There are other such basic solutions available as for example the equation:

$$A = X A^k \cup Z$$

has the solution:

$$A = X^{x_1} Z^{z_1} X^{x_2} Z^{z_2} \cdots X^{x_n} Z^{z_n}$$

where x_i and z_i are integers such that:

for $j > 1$

$$\sum_{i=j}^{n} [(k-1)x_i - z_i] \geq 0$$

and for $j=1$

$$\sum_{i=1}^{n} [(k-1)x_i - z_i] = 1$$

The existence of other basic solutions imply that solutions can be found for still wider classes of languages.

References:

(1) Arden D. N.; "Delayed Large and Finite State Machines, Theory of Computing Machine Design University of Michigan Press, Ann Arbor, Michigan. pp. 1-35, 1960.

(2) Hennie, F. C. "Finite State Models for Logic Machines", John Wiley & Sons, 1968. Chapt. 5.

(3) Hopcroft, J. E., Ullman, J. D., "Formal Languages and Their Relation to Automata". Addison Wesley, 1969 Chapt's. 2-5.