GENERALIZING THE NOTION OF A PERIODIC SEQUENCE

by

R. M. MacGregor

Memorandum No. ERL-M443

15 May 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

GENERALIZING THE NOTION OF A PERIODIC SEQUENCE*

R.M. MacGregor

Computer Science Division
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

Abstract

The usual definition of a periodic sequence is made more general, e.g. the sequence 1,1,2,2,3,3,... will have a "generalized period" of 2. A simple programming, dubbed ORVA, has been devised to generate sequences of numbers, and will be used as a basis for defining the generalized periodicity of a sequence. It is shown that a sequence with period p has generalized period p, and that basic properties of periodic sequences carry over to the generalized case. It is concluded that this new definition is a reasonable extension of the traditional notion of periodicity.

*Research sponsored by National Science Foundation Grant GJ-35604X1.

Introduction

We are going to extend the traditional definition of a periodic sequence to include sequences which (intuitively) behave in periodic fashion, but are not periodic. For example, the sequences

1,0,2,0,3,0,...

and

1,1,4,2,9,4,16,8,...

will have generalized period 2, and

1,2,3,2,3,4,3,4,5,...

has generalized period 3. A sequence like

1,1,2,1,2,3,2,1,2,3,4,3,2,1,...

does not have a generalized period.

The ORVA Language

In order to make precise the notion of generalized periodicity we have devised a very simple programming language, called ORVA, which generates sequences of numbers. ORVA stands for ORdered VAriable -- each variable introduced into a program has a unique positive integer rank. This defines a strict total ordering of all variables, which will limit the way in which variable assignments are made. For most purposes the numerical rank of a variable is unimportant and only its order relative to other variables is considered.

The ORVA language has only three types of statements -- assignment, output, and an unconditional "goto". An assignment has the form

$$x_n := c_n x_n + c_{n-1} x_{n-1} + \cdots + c_1 x_1 + c_o$$

where the variables x_1 to x_{n-1} must have a lower rank than that of x_n, and c_1 to c_n are real numbers. The leading coefficient c_n must be non-negative. An output statement has the form "PRINT x_i", outputting only a single value per statement. The "goto" has the form "GO L" where L is a label attached to some preceding statement.

Unless otherwise specified, a subscript for a variable x_i denotes its rank relative to other variables x_j, but says nothing about its relation to a variable w_j, i.e., $i < j$ implies $\text{rank}(x_i) < \text{rank}(x_j)$ but possibly $\text{rank}(x_i) > \text{rank}(w_j)$ for some w_j.

Sample program:

$x_1 := 1$

$x_2 := 0$

L: $x_2 := x_2 + x_1$

$x_1 := x_1 + 2$

PRINT x_2

GO L

The resulting output is the sequence 1,4,9,16,25,... . Typically the first statements initialize variables and the rest of the code is an infinite loop.

Definition of Generalized Periodicity

<u>Definition</u>. Suppose an ORVA program generates some sequence S. A <u>reduced</u> ORVA program is one which generates S with the minimal number of "PRINT" statements in its loop.

Definition. The __generalized period__ (abbreviated g.p.) of a sequence that can be generated by an ORVA program is defined to be the number of "PRINT" statements in the loop of a reduced ORVA program for that sequence. We say that a sequence is __generally periodic__ if and only if it can be generated by an ORVA program.

The idea of generalized periodicity and this formulation of its definition is due to Manuel Blum.

We will now describe a construction which allows us to put ORVA programs into a form which is easy to work with.

Definition. The __normal form__ for a loop allows only one assignment to each variable, zero or one print statements for each variable, and places the assignment statements first, followed by the output statements, followed by a goto. The assignments are made to variables in order of rank, with the highest ranking variable being assigned first.

__Example.__

$$
\begin{array}{ll}
L: & x_n := c_{n,n}x_n + \cdots + c_{n,1}x_1 + c_{n,o} \\
 & x_{n-1} := c_{n-1,n}x_{n-1} + \cdots + c_{n-1,1}x_1 + c_{n-1,o} \\
 & \vdots \\
 & x_1 := c_{1,1}x_1 + c_o \\
 & \text{PRINT } x_{i_1} \\
 & \vdots \\
 & \text{PRINT } x_{i_m} \\
 & \text{GO } L
\end{array}
$$

We require $i_j = i_k$ iff $j = k$, and $m \leq n$.

Theorem 1. Every ORVA loop can be put into normal form, without altering the number of "PRINT" statements.

Proof. Given a loop of code, we will produce an equivalent loop in normal form which generates the same output for each cycle of the loop.

First we move all PRINT statements to the end of the loop and eliminate multiple printing of the same variable. For each statement "PRINT x_i" we introduce a previously unused variable w_j with $\mathrm{rank}(w_j) > \mathrm{rank}(x_i)$. Delete the statement "PRINT x_i" and insert the code "$w_j := x_i$; PRINT w_j". If "PRINT x_i" occurs in more than one place in the loop a different w_j must be used in each instance. Each "PRINT w_j" can be moved toward the end of the loop without altering the value of w_j. Hence, we can now move all PRINT statements to the end of the loop so that they form a block of output statements which immediately preceeds the "GO" statement. If we don't shuffle the original order of the PRINT statements then the result is a program which produces the same output as before.

We now have a loop consisting of a block of assignment statements, followed by a block of output statements, followed by a goto. We must show how to convert an arbitrary block of assignment statements into an equivalent one where the assignments are to variables of successively decreasing rank. The proof is by induction on n, the highest rank of any variable in the block.

Base. For $n = 1$ all assignments are to the same variable, so we have a block

$$x_1 := a_1x_1 + a_o$$
$$x_1 := b_1x_1 + b_o$$
$$\vdots$$
$$x_1 := m_1x_1 + m_o$$

We can combine the first two assignments into the single statement

$$x_1 := b_1(a_1x_1+a_o) + b_o \qquad (= b_1a_1x_1 + (b_1a_o+b_o))$$

so we can delete the first two statements and insert this one in their place. If we continue to combine the first two assignments of each new block we will eventually be left with one statement (e.g. $x_1 := c_1x_1 + c_o$) which is equivalent to the entire original block of statements. This statement is in the desired form.

Induction Step. By hypothesis assume that any block of statements all of whose variables have rank less than n can be converted into a block where assignments are to variables of successively decreasing rank, $n > 1$. Let $\text{rank}(x_i) = i$.

Consider the statements

(1) $x_i := a_ix_i + a_{i-1}x_{i-1} + \cdots + a_o$

(2) $x_n := b_nx_n + b_{n-1}x_{n-1} + \cdots + b_o$

and the statement

(3) $x_n := b_nx_n + \cdots + b_{i+1}x_{i+1} + b_i(a_ix_i+\cdots+a_o) + b_{i-1}x_{i-1} + \cdots + b_o$

$$(= b_nx_n + \cdots + b_{i+1}x_{i+1} + b_ia_ix_i + (b_ia_{i-1}+b_{i-1})x_{i-1} + \cdots$$
$$+ (b_ia_1+b_1)x_1 + (b_ia_o+b_o)) \quad .$$

If (1) and (2) occur successively in a block and $i = n$ then we can delete them and insert (3) in their place. If $i < n$ then we insert (3), followed by (1). Using this method, we find the last assignment in a block to x_n and move it upwards to the top of the block (by combining or switching with the immediately preceding assignment and making necessary changes in the scalars). When we reach the top the block will consist of an assignment to x_n, followed by assignments to variables of lower rank. By the induction hypothesis these can be converted into a block of decreasingly ranked assignments, so that the whole block has the required form. Q.E.D.

We will henceforth assume that all ORVA loops initially are in normal form.

Properties of Generally Periodic Sequences

We will now prove some theorems which describe the properties of sequences which are generated by ORVA programs.

Convention. We denote the value of a variable x after t iterations of the loop by $x(t)$. $x(0)$ is the value of x as the loop is about to be entered for the first time. Hence, each variable x in an ORVA loop is associated with a function $x(t)$ over the non-negative integers.

Theorem 2 (Monotonicity Theorem). For any variable x in an ORVA loop there is an integer t_o such that for all $t \geq t_o$, $x(t)$ is either constant or strictly monotonic. We call such a function ultimately monotonic (abbreviated u.m.).

Proof. The proof is an induction on n, the highest rank of any variable in an ORVA loop in normal form.

Base. $n = 1$. Let x have rank 1. Then the loop has one assignment: $x := ax+b \quad (a \geq 0)$.

Case 1. $a = 0$. Then $x(t) = b$ for all t, hence it is constant and therefore u.m.

Case 2. $a = 1$. An easy induction proves that $x(t) = bt+x(0)$, so $x(t)$ is constant if $b = 0$ and strictly monotonic otherwise.

Case 3. $a > 0$, $a \neq 1$.

Claim. $x(t) = a^t(x(0)+\frac{b}{a-1}) - \frac{b}{a-1}$.

Proof of Claim by induction on t:

If $t = 0$ then $a^0(x(0)+\frac{b}{a-1}) - \frac{b}{a-1} = x(0)$. ✓

Assuming that the claim holds for $x(t-1)$ we have

$$\begin{aligned}
x(t) &= ax(t-1) + b && \text{by definition} \\
&= a(a^{t-1}(x(0)+\tfrac{b}{a-1}) - \tfrac{b}{a-1}) + b && \text{by inductive hypothesis} \\
&= a^t(x(0)+\tfrac{b}{a-1}) + b - \tfrac{ab}{a-1} \\
&= a^t(x(0)+\tfrac{b}{a-1}) + \tfrac{((a-1)-a)b}{a-1} \\
&= a^t(x(0)+\tfrac{b}{a-1}) - \tfrac{b}{a-1} \quad ✓
\end{aligned}$$

so the claim is true. Since a function f of the form $f(t) = a^t k_1 + k_2$ is strictly monotonic when $a > 0$, $x(t)$ is u.m.

Induction Step. $n > 1$. Assume by hypothesis that all variables in a loop with rank $< n$ are u.m.

First we will need a

Lemma. Fix n. Assume that if a variable has rank < n then it is u.m. Let w have rank n and let $x_1, x_2, \ldots, x_{n-1}$ have ranks 1 through n-1 respectively (so they are u.m. by assumption). Define the function $w(t)$ by

$$w(t) = c_{n-1}x_{n-1}(t) + c_{n-2}x_{n-2}(t) + \cdots + c_1x_1(t)$$

where the c_i are arbitrary constants. Then we claim that $w(t)$ is u.m.

Proof of Lemma. We will produce a variable z with rank n-1 such that $w(t) = z(t)$ for all $t \geq 0$. Then since $z(t)$ is u.m. by assumption, $w(t)$ will also be u.m.

Consider a loop in normal form containing two variables x and y with $n > \text{rank}(y) > \text{rank}(x)$. Let their assignments in the loop be

$$y := ay + bx + f(x_1, \ldots, x_i)$$
$$x := cx + g(x_1, \ldots, x_j)$$

where we are using "f" and "g" as a shorthand to denote a sum of other lower ranking variables. Then we can write

$$y(t+1) = ay(t) + bx(t) + f(t)$$
$$x(t+1) = cx(t) + g(t) \quad .$$

We wish to show that $z(t) = py(t) + qx(t)$ is u.m. For any scalars p and q. Without loss of generality assume $z(0) = py(0) + qx(0)$. At the beginning of the loop containing x and y insert the assignment

(4) $$z := az + (pb+qc-qa)x + pf(x_1, \ldots, x_i) + qg(x_1, \ldots, x_j) \quad .$$

__Claim.__ $z(t) = py(t) + q(t)$

Proof by induction on t: For $t = 0$ the claim is true by assumption. Assume the claim holds for $z(t-1)$, so that

$$\begin{aligned} z(t) &= az(t-1) + (pb+qc-qa)x(t-1) + pf(t-1) + qg(t-1) \\ &= a(py(t-1)+qx(t-1)) + (pb+qc-qa)x(t-1) + pf(t-1) + qg(t-1) \\ &\quad \text{by induction hypothesis} \\ &= p(ay(t-1)+bx(t-1) + f(t-1)) + q(cx(t-1)+g(t-1)) \\ &= py(t) + qx(t) \end{aligned}$$

so the claim is true.

Now observe that z does not depend upon the variable y. Hence we can eliminate the variable y and its assignment statement, let $\mathrm{rank}(z)$ have the value $\mathrm{rank}(y)$, and leave (4) in the loop. Then since $\mathrm{rank}(z) < n$ the assumption tells us that $z(t)$ is u.m., and hence $py(t)+qx(t)$ is u.m.

Observe that using this method we can successively produce $c_1x_1+c_2x_2$, $(c_1x_1+c_2x_2)+c_3x_3$, $((c_1x_1+c_2x_2)+c_3x_3)+c_4x_4$, etc. with each sum being u.m. Therefore we can produce a variable z such that $z(t) = c_1x_1(t)+c_2x_2(t)+\cdots+c_{n-1}x_{n-1}(t)$ and $\mathrm{rank}(z) = n-1$. This proves the lemma.

Now we wish to show that if

$$w := a_n w + a_{n-1}x_{n-1} + \cdots + a_1 x_o + a_o$$

is the first statement in a loop in normal form and $\mathrm{rank}(w) = n$ then $w(t)$ is u.m.

Case 1. $a_n = 0$. Then the lemma can be applied to show $w(t)$ is u.m.

Case 2. $a_n > 0$. Let $f(t) = a_{n-1}x_{n-1}(t) + \cdots + a_1x_1(t) + a_o$. Because the variables x_1 to x_{n-1} have rank $< n$ the lemma again applies to tell us that $f(t)$ is u.m.

A. Suppose for all $t > t_o$ we have $f(t) \geq f(t-1)$. Fix some $t > t_o$. Then if $w(t) > w(t-1)$ we get $w(t+1) = a_n w(t) + f(t) > a_n w(t-1) + f(t)$ $\geq a_n w(t-1) + f(t-1) = w(t)$ hence

(i) $$w(t) > w(t-1) \Rightarrow w(t+1) > w(t) \quad .$$

Similarly $w(t) \geq w(t-1) \Rightarrow w(t+1) \geq w(t)$.

B. Suppose for all $t > t_o$ we have $f(t) \leq f(t-1)$. Fix some $t > t_o$. Then if $w(t) < w(t-1)$ we have $w(t+1) = a_n w(t) + f(t) < a_n w(t-1) + f(t)$ $\leq a_n w(t-1) + f(t-1) = w(t)$ giving

(ii) $$w(t) < w(t-1) \Rightarrow w(t+1) < w(t) \quad .$$

Similarly $w(t) \leq w(t-1) \Rightarrow w(t+1) \leq w(t)$.

Now it is easy to show that $w(t)$ is u.m. First determine if A or B holds for $f(t)$. Suppose A is true. Then after t_o steps we inspect $w(t)$. If $w(t+1) > w(t)$ for any $t > t_o$ we know that $w(t)$ is strictly monotone increasing, by (i). Otherwise if $w(t+1) \leq w(t)$ for all $t > t_o$ but, for some $t > t_o$, $w(t+1) \geq w(t)$, then $w(t)$ is constant from then on, again using (i). Lastly, it can happen that $w(t+1) < w(t)$ for all $t > t_o$, so that $w(t)$ is strictly monotone decreasing.

Similarly, if B holds then we use (ii) to get an equivalent result.

This proves the induction step. Q.E.D.

Corollary. If a sequence has generalized period one then it is ultimately monotonic.

The Monotonicity Theorem is a useful tool in proving properties of generally periodic sequences. It forms the basis for an easy proof of the next theorem.

Theorem 3. A sequence of period p has generalized period p.

Proof. Let $\sigma_o, \sigma_1, \ldots, \sigma_{p-1}, \sigma_p, \sigma_{p+1}, \ldots$ be a sequence with p as its smallest period $(\sigma_i = \sigma_{i+p})$. We can generate this sequence with an ORVA loop that uses p variables equal to constants $\sigma_o, \ldots, \sigma_{p-1}$, and has p PRINT statements.

Let us assume that an ORVA program exists which generates $\sigma_o, \sigma_1, \sigma_2, \ldots$ using $m < p$ PRINT statements in its loop. Assuming that the loop is in normal form, then there are m different variables $z_1, \ldots, z_m$ which are printed at the end of each cycle in the loop.

Consider any z_i. By the Monotonicity Theorem $z_i(t)$ is u.m. If $z_i(t)$ was ultimately strictly increasing or decreasing it would take on more than p different values, and hence could not be printing correct values for the sequence $\sigma_o, \sigma_1, \ldots$. We conclude that for all $i = 1, \ldots, m$, $z_i(t)$ is constant (ultimately). Hence the sequence $z_1(0), z_2(0), \ldots, z_m(0)$, $z_1(1), z_2(1), \ldots$ has period m because $z_i(t) = z_i(t+1)$ for all $t > t_o$. But $m < p$ and we assumed that we were generating a sequence of period p. Contradiction. Q.E.D.

It is clearly desirable that Theorem 3 be true, if our definition for a generalized period is to be a good one. We can now see why certain constraints were placed upon the ORVA language:

<u>Remark</u>. If we allow the leading coefficient of the general assignment statement to be negative then Theorems 2 and 3 don't hold. Example:

```
     x := 1
L:   x := -x+1
     PRINT x
     GO L
```

This program outputs the period two sequence 0,1,0,1,... with only one PRINT statement in its loop. x(t) is not u.m.

<u>Remark</u>. If we eliminate the rankings we again can find a counterexample to Theorems 2 and 3. Example:

```
     x := 1
     y := 2
L:   z := x
     x := y
     y := z
     PRINT z
     GO L
```

This program outputs 1,2,1,2,... with only one PRINT statement.

<u>Definition</u>. For any numerical sequence $S = \sigma_o,\sigma_1,\sigma_2,...$ the <u>sequence of differences</u> is $\Delta S = \sigma_1-\sigma_o,\sigma_2-\sigma_1,\sigma_3-\sigma_2,...$.

<u>Theorem 4</u>. If a sequence S has generalized period p then the corresponding sequence of differences ΔS has g.p. p. Conversely, if

a sequence T with g.p. p is considered a sequence of differences, then any sequence S such that $\Delta S = T$ must have g.p. p.

Proof. Suppose we are considering a sequence where the (reduced) ORVA loop prints successively the variables $z_1, z_2, \ldots, z_p$ in each iteration. Suppose z_1 is assigned in the loop by the statement

$$z_1 := cz_1 + c_n x_n + \cdots + c_o \quad \text{and} \quad \text{w.l.o.g.}$$

assume that $z_1, \ldots, z_p$ are initialized before entering the loop.

Case 1. $p = 1$. We construct an ORVA loop to generate successive differences as follows: Add the statement "$w := -z_1$" at the top of the loop, where w is a new variable. Insert "$w := w+z_1$" directly before "PRINT z_1", and change "PRINT z_1" to "PRINT w". The loop now outputs $w(t) = z_1(t) - z_1(t-1)$, as desired.

Case 2. $p > 1$. Create p new variables $w_1, \ldots, w_p$ and at the beginning of the loop add

$$\begin{aligned} w_1 &:= z_2 - z_1 \\ w_2 &:= z_3 - z_2 \\ &\vdots \\ w_{p-1} &:= z_p - z_{p-1} \\ w_p &:= cz_1 + c_n x_n + c_{n-1} x_{n-1} + \cdots + c_o - z_p \quad . \end{aligned}$$

Delete all PRINT statements and insert "PRINT w_1;...;PRINT w_p" at the end of the loop. This prints the sequence of differences.

Note that we have shown that $\text{g.p.}(\Delta S) \le \text{g.p.}(S)$.

Now assume that a sequence of differences ΔS is generated by a reduced ORVA loop printing the variables $w_1, \ldots, w_m$, and the original sequence S started with σ_o. Before the loop insert "$z_1 := \sigma_o$; PRINT z_1". Let $z_1,\ldots,z_m$ be new variables. Delete all "PRINT w_i" statements.

Case 1. $m = 1$. At the end of the loop insert "$z_1 := z_1+w_1$; PRINT z_1".

Case 2. $m > 1$. At the end of the loop add

$$
\begin{aligned}
&z_2 := z_1+w_1 \\
&\quad\vdots \\
&z_m := z_{m-1}+w_{m-1} \\
&z_1 := z_1+w_m+\cdots+w_1 \\
&\text{PRINT } z_2 \\
&\quad\vdots \\
&\text{PRINT } z_m \\
&\text{PRINT } z_1
\end{aligned}
$$

This new loop outputs the original sequence S. We see g.p.$(S) \le m =$ g.p.(ΔS). Hence g.p.$(S) =$ g.p.(ΔS). Q.E.D.

Definition. Let $S = \sigma_o,\sigma_1,\sigma_2,\ldots$. Then $-S = -\sigma_o,-\sigma_1,-\sigma_2,\ldots$. Let $T = \tau_o,\tau_1,\tau_2,\ldots$. Then $S+T = \sigma_o+\tau_o,\sigma_1+\tau_1,\sigma_2+\tau_2,\ldots$.

Theorem 5. g.p.$(S) =$ g.p.$(-S)$.

Proof. In the program generating S replace each statement "PRINT x_i" by "$w_i := -x_i$; PRINT w_i", where w_i is a new variable. Then g.p.$(-S) \le$ g.p.(S). But then g.p.$(S) =$ g.p.$(-(-S)) \le$ g.p.$(-S) \le$ g.p.(S).

Q.E.D.

Theorem 6. If g.p.$(S) =$ g.p.$(T) = p$ then g.p.$(S+T) \le p$.

Proof. Assume w.l.o.g. that the programs for S and T do not have any variables in common. Form a new loop consisting of all code from the loop for S and the loop for T. Assuming that variables $s_1,\ldots,s_p$ and $t_1,\ldots,t_p$ are printed, delete all PRINT statements and in place of each PRINT s_i and PRINT t_i we insert "w_i := s_i+t_i; PRINT w_i" where w_i is a new variable.

Similarly altering the initial code (before the loop) leads to a program for S+T with p print statements. Q.E.D.

Note that g.p.(S+(-S)) = 1 for all sequences S. We will show later that g.p.(S+T) divides p when g.p.(S) = g.p.(T) = p.

Corollary 6. If g.p.(S) = g.p.(T) = p then g.p.(S-T) $\leq$ p.

Theorem 7. If x_i is a variable in an ORVA program then there exist positive real constants $\lambda_1,\ldots,\lambda_m$ and polynomials $p_1(t),\ldots,p_m(t)$ such that $x_i(t) = p_1(t)\lambda_1^t + \cdots + p_m(t)\lambda_m^t$. The numbers $\lambda_1,\ldots,\lambda_m$ correspond to non-zero leading coefficients in the assignment statements.

Proof. Let $\vec{x}(t)$ denote the vector $\langle x_1(t),\ldots,x_n(t)\rangle$ where x_1 to x_n are variables in an ORVA loop in normal form (rank(x_i) < rank(x_{i+1})). Each iteration of the loop is equivalent to a matrix transformation $\vec{x}(t+1) = A\cdot\vec{x}(t)$, where A is an $n\times n$ lower triangular matrix. Induction will prove that $A^t\cdot\vec{x}(0) = \vec{x}(t)$.

The diagonal elements of A are also its eigenvalues, and since a diagonal element corresponds to the non-negative leading coefficient of an assignment, the eigenvalues are non-negative. There exist matrices E and T such that $A = T^{-1}ET$ and E is in Jordan-canonical form, with the eigenvalues of A being the diagonal elements of E. Then $\vec{x}(t) = T^{-1}E^tT\vec{x}(0)$.

Suppose E is a strictly diagonal matrix, say

$$E = \begin{bmatrix} \lambda_1 & & & \bigcirc \\ & \lambda_2 & & \\ & & \ddots & \\ \bigcirc & & & \lambda_n \end{bmatrix} \quad \text{and} \quad E^t = \begin{bmatrix} \lambda_1^t & & & \bigcirc \\ & \lambda_2^t & & \\ & & \ddots & \\ \bigcirc & & & \lambda_n^t \end{bmatrix}.$$

Then

$$x_i(t) = \underset{\substack{\uparrow \\ i^{\text{th}} \text{ place}}}{\langle 0,\ldots,0,1,0,\ldots,0\rangle} T^{-1}E^tT\vec{x}(0)$$

$$= \langle a_1,\ldots,a_n\rangle \cdot E^t \cdot \langle b_1,\ldots,b_n\rangle \quad \text{for some } a_j\text{'s and } b_j\text{'s}$$

$$= a_1b_1\lambda_1^t + \cdots + a_nb_n\lambda_n^t$$

which is in the proper form.

Next take the case that E is a Jordan block, say

$$E = J = \begin{bmatrix} \lambda & 1 & & & \bigcirc \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ \bigcirc & & & & \lambda \;\; 1 \\ & & & & \lambda \end{bmatrix}.$$

Using the binomial identity $\binom{t}{m} = \binom{t-1}{m-1} + \binom{t-1}{m}$ and interpreting $\binom{t}{m}$ as 0 if $m > t$ we can prove by induction that

$$J^t = \begin{bmatrix} \lambda^t & t\lambda^{t-1} & \binom{t}{2}\lambda^{t-2} & \binom{t}{3}\lambda^{t-3} & & \binom{t}{n}\lambda^{t-n} \\ & \lambda^t & t\lambda^{t-1} & \binom{t}{2}\lambda^{t-2} & & \binom{t}{n-1}\lambda^{t-(n-1)} \\ & & \ddots & & & \\ & \bigcirc & & \ddots & & \\ & & & & \ddots & \\ & & & & & \lambda^t \end{bmatrix}.$$

Then

$$x_i(t) = \langle a_1,\ldots,a_n\rangle \cdot J^t \cdot \langle b_1,\ldots,b_n\rangle$$

$$= \langle a_1,\ldots,a_n\rangle \cdot \langle \sum_{i=0}^{n-1} b_{i+1}\binom{t}{i}\lambda^{t-i},\ \sum_{i=0}^{n-2} b_{i+1}\binom{t}{i}\lambda^{t-i},\ldots,b_n\lambda^t\rangle$$

$$= p(t)\lambda^t \quad \text{for some polynomial } p(t)\ .$$

Finally, in the most general case where E contains several Jordan sub-blocks, a variable x_i is the sum of elements of the first two types, so $p_1(t)\lambda_1^t + \cdots + p_m(t)\lambda_m^t = x_i(t)$ for some p_j's and λ_j's. Q.E.D.

The outline for this proof comes from another presented by Pravin Varaiya.

<u>Theorem 8</u>. Given any function of t of the form $p_1(t)\lambda_1^t + \cdots + p_m(t)\lambda_m^t$ where $\lambda_i > 0$ for $i = 1,\ldots,m$, we can produce an ORVA program containing a variable w such that $w(t)$ is that function for $t \geq 0$.

<u>Proof</u>. We will prove the theorem for the case $w(t) = t^k\lambda^t$. The more general case follows easily. Proof by induction on k:

<u>Base</u>. $k = 0$. Let $w(0) = 1$ and place "$w := \lambda w$" in the loop. Then we have $w(t) = \lambda^t$.

<u>Induction Step</u>. $k > 0$. Assume we have variables $x_o,\ldots,x_{k-1}$ such that $x_i(t) = t^i\lambda^t$ for $i < k$. Let $w(0) = 0$. At the top of the loop containing the x_i add the statement $w := \lambda w + \binom{k}{1}\lambda x_{k-1} + \cdots + \binom{k}{k}\lambda x_o$.

<u>Claim</u>. $w(t) = t^k\lambda^t$.

Proof of Claim by induction on t: If $t = 0$ then $w(0) = 0 = 0^k\lambda^o$. ✓

Assume the claim true for the first t values of w.

$$\begin{aligned} w(t+1) &= \binom{k}{0}\lambda t^k\lambda^t + \binom{k}{1}\lambda t^{k-1}\lambda^t + \cdots + \binom{k}{k}\lambda t^o\lambda^t \qquad \text{by hypothesis} \\ &= \lambda^{t+1}\left[\binom{k}{0}t^k + \binom{k}{1}t^{k-1} + \cdots + \binom{k}{k}t^o\right] \\ &= \lambda^{t+1}(t+1)^k \ . \ \checkmark \end{aligned}$$

This proves the claim and hence the theorem. Q.E.D.

For the following definitions let $R = \rho_o, \rho_1, \rho_2, \ldots$ and $S = \sigma_o, \sigma_1, \sigma_2, \ldots$ be generally periodic sequences.

<u>Definition</u>. R and S are <u>equivalent</u> $(R \equiv S)$ if there exist constants i_o and j_o such that

$$\rho_{i_o+i} = \sigma_{j_o+i} \qquad \text{for all } i \geq 0 \ .$$

For example, suppose $R = 0,1,2,3,\ldots$ and $S = 2,3,4,\ldots$. Then $R \equiv S$.

<u>Definition</u>. R is <u>contained</u> in S $(R \subseteq S)$ if there exist constants i_o and j_o such that $\rho_{i_o} = \sigma_{j_o}$ and an increasing function f such that $f(0) = 0$ and $\rho_{i_o+i} = \sigma_{j_o+f(i)}$.

Example: If $R = 1,2,4,8,\ldots$ and $S = 1,2,3,\ldots$ then $R \subseteq S$.

<u>Definition</u>. If the function f above can be represented as $f(t) = mt$ where $m \geq 1$ then we say that R is an <u>m-section</u> of S. Alternatively, we can say that S is m-times as dense as R $(D(S/R) = m)$.

Example: Let $R = 1,4,16,64,\ldots$ and $S = 1,2,4,8,16,\ldots$. Then

R is a 2-section of S, $D(S/R) = 2$.

<u>Notation</u>. Let $(r(t))_{t=0}^{\infty}$ denote the sequence $r(0), r(1), r(2), \ldots$ generated by a variable r.

Suppose $R = r(t)_{t=0}^{\infty}$ and $S = (s(t))_{t=0}^{\infty}$. If R is an m-section of S then we have $r(i_o+t) = s(j_o+mt)$ for all $t \geq 0$. We note a corollary to Theorem 7.

<u>Corollary 7</u>. If $S = (s(t))_{t=0}^{\infty}$ has g.p. 1 then there exist constants $\lambda_1, \ldots, \lambda_n$ and polynomials $p_1(t), \ldots, p_n(t)$ such that $s(t) = p_1(t)\lambda_1^t + \cdots + p_n(t)\lambda_n^t$.

Using Theorem 8 we can derive another result.

<u>Corollary 8</u>. Let S have g.p. 1. Then given an integer $m \geq 1$, each m-section of S has g.p. 1.

<u>Proof</u>. Let $S = (s(t))_{t=0}^{\infty}$ and suppose $R = (r(t))_{t=0}^{\infty}$ is an m-section of S with $r(i_o+t) = s(j_o+mt)$ for all $t \geq 0$. Then for $t \geq i_o$ $r(t) = s(j_o + m(t-i_o))$. By Corollary 7 there exist $\lambda_1, \ldots, \lambda_n$ and $p_1(t), \ldots, p_n(t)$ such that

$$\begin{aligned} r(t) &= p_1(j_o+m(t-i_o))\lambda_1^{(j_o+m(t-i_o))} + \cdots + p_n(j_o+m(t-i_o))\lambda_n^{(j_o+m(t-i_o))} \\ &= p_1'(t)\lambda_1'^{t} + \cdots + p_n'(t)\lambda_n'^{t}, \end{aligned}$$

so R has g.p. 1 by Theorem 8. Q.E.D.

<u>Theorem 9</u>. Let R be a sequence with generalized period 1. For each integer $m \geq 1$ there exists a unique (up to equivalence) sequence S with g.p. 1 such that R is an m-section of S ($D(S/R) = m$).

Example. If a sequence $1,i_1,9,i_2,25,i_3,\ldots$ is known to have g.p. 1 then the sequence $i_1,i_2,i_3,\ldots$ must be equivalent to the sequence $4,16,36,\ldots$.

Proof. Let $R = (r(t))_{t=0}^{\infty}$. By Theorem 7 there exist constants $\lambda_1,\ldots,\lambda_n$ and polynomials $p_1(t),\ldots,p_n(t)$ such that $r(t) = p_1(t)\lambda_1^t + \cdots + p_n(t)\lambda_n^t$.

Existence: Fix $m \geq 1$. By Theorem 8 there exists a sequence $S = (p_1(\frac{t}{m})\lambda_1^{t/m} + \cdots + p_n(\frac{t}{m})\lambda_n^{t/m})_{t=0}^{\infty}$ with g.p. 1. Then $D(S/R) = m$.

Uniqueness: Suppose $S = (s(t))_{t=0}^{\infty}$ and $S' = (s'(t))_{t=0}^{\infty}$ both have g.p. 1 and $D(S/R) = D(S'/R) = m$. By definition there exist constants i_o, j_o, and j_o' such that $r(i_o+im) = s(j_o+im) = s'(j_o'+im)$ for all $i \geq 0$. Consider the sequence $S-S'$. By the corollary to Theorem 6 $S-S'$ has g.p. 1. Every m^{th} term of this sequence is zero. Since it is ultimately monotonic, it must be equivalent to the sequence $0,0,0,\ldots$. Hence $S \equiv S'$. Q.E.D.

Before we prove our final theorem we need a lemma.

Lemma. Let R, S, and W be sequences with g.p. 1, such that $W \subseteq R$ and $W \subseteq S$. Suppose W is a p-section of R and W is a q-section of S ($D(R/W) = p$ and $D(S/W) = q$). Then p divides q implies $R \subseteq S$.

Proof. Let $q = mp$ where m is a positive integer. By Theorem 7 W is equivalent to a sequence $(p_1(t)\lambda_1 + \cdots + p_n(t)\lambda_n^t)_{t=0}^{\infty}$. By Theorem 8 there exist sequences $R' = (p_1(\frac{t}{p})\lambda_1^{t/p} + \cdots + p_n(\frac{t}{p})\lambda_n^{t/p})_{t=0}^{\infty}$ and $S' = (p_1(\frac{t}{q})\lambda_1^{t/q} + \cdots + p_n(\frac{t}{q})\lambda_n^{t/q})_{t=0}^{\infty}$ each having g.p. 1. Since $S' = (p_1(\frac{t}{mp})\lambda_1^{t/mp} + \cdots + p_n(\frac{t}{mp})\lambda_n^{t/mp})_{t=0}^{\infty}$ we observe that R' is an m-section of S', hence $R' \subseteq S'$.

By construction $D(R'/W) = p = D(R/W)$ and $D(S'/W) = q = D(S/W)$ so by Theorem 9 we have $R \equiv R'$ and $S \equiv S'$. But then $R \subseteq S$. Q.E.D.

We may refer to a generalized period of a sequence to mean the number of print statements in the loop of a (possibly) unreduced ORVA program for that sequence. The next theorem shows that all generalized periods of a sequence must be integral multiples of the fundamental (or smallest) generalized period. This is in accord with the similar result for periodic sequences, and hence strengthens our belief that we have a good definition for the generalized period of a sequence.

Theorem 10. Let X have a g.p. of p and let Y have a g.p. of q. Let d be the greatest common divisor of p and q. If $X \equiv Y$ then there exists a sequence Z with a g.p. of d such that $Z \equiv X \equiv Y$.

Proof. We will assume that $X = Y$ and find an appropriate Z such that $Z = X = Y$. The result for equivalence follows.

Let $X = ((x_i(t))_{i=0}^{p-1})_{t=0}^{\infty}$ and $Y = ((y_i(t))_{i=0}^{q-1})_{t=0}^{\infty}$. Let X_i denote $(x_i(t))_{t=0}^{\infty}$ and $Y_j = (y_j(t))_{t=0}^{\infty}$, so that each X_i is a p-section of X, and Y_j is a q-section of Y. For convenience let $\mu(0), \mu(1), \mu(2), \ldots$ denote the sequence $x_o(0), x_1(0), \ldots, x_{p-1}(0), x_o(1), x_1(1), \ldots = X = Y$. Note that

$$x_i(t) = \mu(pt+i) \quad \text{for all } i \geq 0$$

and

$$y_i(t) = \mu(qt+i) \quad \text{for all } i \geq 0 \quad .$$

Next define a sequence $A_o = (a_o(t))_{t=0}^{\infty}$ with $a_o(t) = x_o(qt) = \mu(pqt) = y_o(pt)$ for all $t \geq 0$. Intuitively, we chose points where X_o and Y_o "intersect". We have $A_o \subseteq X_o$ with $D(X_o/A_o) = q$ and $A_o \subseteq Y_o$ with

$D(Y_o/A_o) = p$. Corollary 8 tells us that X_o and Y_o, and hence A_o, all have g.p. 1. We now use Theorem 9 to find the unique sequence with g.p. 1 $Z_o = (z_o(t))_{t=0}^{\infty}$ which is $\frac{pq}{d}$ times as dense as A_o, $D(Z_o/A_o) = \frac{pq}{d}$. Because p and q both divide $\frac{pq}{d}$ the preceding lemma tells us that $X_o \subseteq Z_o$ and $Y_o \subseteq Z_o$. We would like to show that $X_i \subseteq Z_o$ whenever d divides i, $i = 0,\ldots,p-1$.

Pick any such i, setting $i = cd$. There exist constants a' and b' such that $d = a'p+b'q$. Either $a' < 0 < b'$ or $b' < 0 < a'$. W.l.o.g. assume the former inequality holds, so we set $a = -ca'$, $b = cb'$ to get $ap+i = bq$ with a and b non-negative.

We now look at points where X_i and Y_o "intersect". Define $B_o = (b_o(t))_{t=0}^{\infty}$ with $b_o(t) = y_o(b+tp) = \mu(bq+tpq) = \mu(ap+i+tqp) = x_i(a+tq)$. Using the lemma freely we have

$$\begin{aligned}
D(Y_o/A_o) &= p \quad \text{and} \quad D(Z_o/A_o) = \tfrac{pq}{d} \quad \text{implies} \quad D(Z_o/Y_o) = \tfrac{q}{d}\\
D(Y_o/B_o) &= p \quad \text{and} \quad D(Z_o/Y_o) = \tfrac{q}{d} \quad \text{implies} \quad D(Z_o/B_o) = \tfrac{pq}{d}\\
D(X_i/B_o) &= q \quad \text{and} \quad D(Z_o/B_o) = \tfrac{pq}{d} \quad \text{implies} \quad D(Z_o/X_i) = \tfrac{p}{d}\,.
\end{aligned}$$

This requires that $X_i \subseteq Z_o$, as desired.

We now know that $X_i \subseteq Z_o$ for $i = 0,d,2d,\ldots,p-d$. For each such X_i, $D(Z_o/X_i) = \frac{p}{d}$, and there are $\frac{p}{d}$ different such X_i's, hence

$$\begin{aligned}
Z_o &= X_o(0),X_d(0),X_{2d}(0),\ldots,X_{p-d}(0),X_o(1),\ldots,\\
&= \mu(0),\mu(d),\mu(2d),\ldots\,.
\end{aligned}$$

Therefore the sequence $\mu(0),\mu(d),\mu(2d),\ldots$ has g.p. 1.

In a similar manner we can define a sequence A_1 with $a_1(t) = x_1(qt) = y_1(pt)$, construct Z_1 such that $D(Z_1/A_1) = \frac{pq}{d}$, and eventually conclude that the sequence $\mu(1),\mu(d+1),\mu(2d+1),\ldots$ has g.p. 1. Continuing in this fashion we will eventually have sequences $Z_o, Z_1, \ldots, Z_{d-1}$ each with g.p. 1 such that $Z = ((z_i)_{i=0}^{d})_{t=0}^{\infty} = X = Y$. Q.E.D.

Corollary 10.1. If a sequence has a generalized period of q and its fundamental period is p then p divides q.

Corollary 10.2. Let S and T be sequences with g.p.(S) = g.p.(T) = p. Then g.p.(S+T) divides p.

Proof. In Theorem 6 we produced a program for S+T which had p PRINT statements in its loop. By Theorem 10 g.p.(S+T) divides p. Q.E.D.

Conclusion

A significant number of properties of periodic sequences carry over to the generalized case. We feel that they constitute a good justification for choosing our particular definition of a generalized period. The ORVA language has proved to be a useful tool for dealing with generally periodic sequences.

There are sequences, such as the factorial sequence 1,2,6,24,..., which are not generally periodic but are not particularly complex. Hence, it would be nice to further extend the present definition to cover a broader class of sequences. However, a little research revealed that a "natural" attempt to extend the definition resulted in a huge increase in complexity, so that further generalization constitutes a non-trivial problem.

Acknowledgement

I have profitted greatly from discussions with Manuel Blum concerning the material and its presentation.