Yale University Department of Music

Cohn Functions
Author(s): David Lewin
Source: *Journal of Music Theory*, Vol. 40, No. 2 (Autumn, 1996), pp. 181-216
Published by: Duke University Press on behalf of the Yale University Department of Music
Stable URL: http://www.jstor.org/stable/843888
Accessed: 16/12/2009 10:01

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at http://www.jstor.org/page/info/about/policies/terms.jsp. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/action/showPublisher?publisherCode=duke.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Duke University Press and *Yale University Department of Music* are collaborating with JSTOR to digitize, preserve and extend access to *Journal of Music Theory.*

http://www.jstor.org

COHN FUNCTIONS

David Lewin

Our point of departure is work done by Richard Cohn in 1993. A symposium on Cohn's ideas was held at SUNY Buffalo on July 28–29, 1993.[1] This has led to generalizing Cohn's work in a number of directions. The present essay is one such generalization.

Part 1: Cohn sets

This section will briefly summarize Cohn's initial observations. Let us consider the pcset {B,C,D}. If we cast away the pitch class C, and substitute for it the adjacent (NB) pitch class C♯, we obtain the pcset {B,C♯,D}, which is an inverted form of {B,C,D}. I shall say that we have performed a *Cohn flip* on {B,C,D}.

In similar fashion the pcset {C,E,G} can be Cohn-flipped into its inverted form {C,E♭,G}, casting out E and substituting the adjacent (NB) pitch class E♭.

{C,E,G} can also be Cohn-flipped into its inverted form {B,E,G}, casting out C and substituting the adjacent (NB) pitch class B. Thus {C,E,G} *can be Cohn-flipped in two different ways.* A pcset with that property will be called a *Cohn set.*

Cohn listed all possible Cohn sets in the mod 12 pc universe. They are

(a) the singletons: e.g. {C} Cohn-flips into either {B} or {C♯}.
(b) the harmonic triads: e.g. {C,E,G} Cohn-flips as described above.
(c) the anhemitonic pentatonic scales: e.g. {C,D,E,G,A} Cohn-flips into either {C,D,F,G,A} or {B,D,E,G,A}.
(d) the complements of (a)(b)(c) above—thus including diatonic scale-sets.
(e) and no other pcsets.

This list is suggestive. Cohn developed certain findings and raised certain questions about Cohn sets, generalized to universes mod N, where N need not be 12. For instance, if we represent the mod 7 universe by a C major scale, and interpret "adjacent" in a diatonic context, then the Cohn sets are

(a) the singletons: e.g. {C} Cohn-flips into either {B} or (the diatonically adjacent) {D}.
(b) the fifth-spanning dyads: e.g. {C,G} Cohn-flips into either {C,F} or {D,G}.
(c) the triads: e.g. {C,E,G} Cohn-flips into either {B,E,G} or {C,E,A}.
(d) the complements of (a)(b)(c) above—thus including seventh chords.
(e) and no other pcsets.

This list seems equally suggestive.

Cohn sets characteristically give rise to *Cohn cycles* of pcsets. These are formed by enchaining Cohn-flipped sets in the obvious way. Thus, in the mod 12 world, the C-major-scale set gives rise to the familiar cycle of scales through the circle of fifths, i.e.... {C,D,E,F,G,A,B}, {C,D,E,F,G,A,B♭}, {C,D,E♭,F,G,A,B♭}, ..., {C♯,D,E,F♯,G,A,B}, {C,D,E,F♯,G,A,B}, {C,D,E,F,G,A,B}, ... Each pcset on this list is a Cohn-flip of the set that precedes it, and also of the set that follows it. Harmonic triads, as Cohn sets mod 12, give rise to shorter, less exhaustive Cohn cycles, e.g.... {C,E,G}, {C,E♭,G}, {C,E♭,A♭}, {B,D♯,G♯}, {B,E,G♯}, {B,E,G}, {C,E,G}, ...

Part 2: Cohn functions; definitions and examples

From here on I shall present my own work, one generalization of Cohn's (and by no means the only one of interest).

Pcsets—and other related set-theoretic configurations, as we shall see—can be represented by certain *functions,* functions which have as arguments the numbers mod 12, or more generally mod N. The C major diatonic set mod 12, for example, can be represented by the function f whose values are f(0)=yes, f(1)=no, f(2)=yes, f(3)=no, f(4)=yes, f(5)= yes, f(6)=no, f(7)=yes, f(8)=no, f(9)=yes, f(10)=no, f(11)=yes. This function assigns the value "yes" to each pc-number that is a member of the designated pcset, and the value "no" to each pc-number that is not. We shall represent the above function by the symbol (ynynyynynyny).

The 7-35 set class can also be represented by a function that has numbers mod 7 as arguments, and numbers mod 12 as values. The function assigns to each number j mod 7 the j-th interval of the diatonic scale. So f(0)=2: the 0th interval of the scale is 2 semitones. f(1)=2: the next interval of the scale is 2 semitones. f(2)=1: the next interval of the scale is 1 semitone. And so forth: f(3)=2, f(4)=2, f(5)=2, f(6)=1. We represent this function by the symbol (2212221).

In like fashion the C major triad, as a set in the mod 12 universe, can be represented by the mod 12 function (ynnnynnynnn). And the transpositional set class of "major triad" can be represented by the mod 3 function (435), where the numbers 4, 3, and 5 are intervals mod 12.

Cohn-properties of these various pcsets (or set classes) are reflected in certain properties of the mod N functions that represent the various pcsets (or set classes).

The concept of Cohn flipping can be extended to functions with arguments mod N as follows.

DEFINITION 2.1: A mod N function is *Cohn flipped* when an exchange of different (NB) values at adjacent (NB) arguments gives rise to a rotated retrograde of the original function.

For example, consider the mod 12 function (ynynnnnnnnny), which represents the pcset {B,C,D}. Let us exchange the different (NB) values y and n at the adjacent (NB) arguments 0 and 1. We obtain the function (*ny*ynnnnnnnny), which represents {B,C♯,D}. The exchanged function values are italicized. If we start at the italicized *y* and read the new function circularly backwards therefrom, we obtain (*yn*ynnnnnnnny), which (ignoring italics) is the original function. So (ynynnnnnnnny) has been Cohn-flipped into (nyynnnnnnnny), by the above definition.

Or consider the mod 3 function (129), where the numbers 1, 2, and 9 are mod 12 intervals. This function represents the transpositional set class of {B,C,D}. If we exchange the different (NB) values 1 and 2 at the adjacent (NB) arguments 0 and 1, we obtain the new function (*21*9), which represents the transpositional set class of {B,C♯,D}. If we start at the italicized *1* and read the new function circularly backwards therefrom, we obtain (*12*9), which (ignoring italics) is the original function.

We can now define "Cohn functions."

DEFINITION 2.2: A *Cohn function* is a function with arguments mod N that can be Cohn-flipped in two different ways—i.e. at two different adjacent-argument locations.

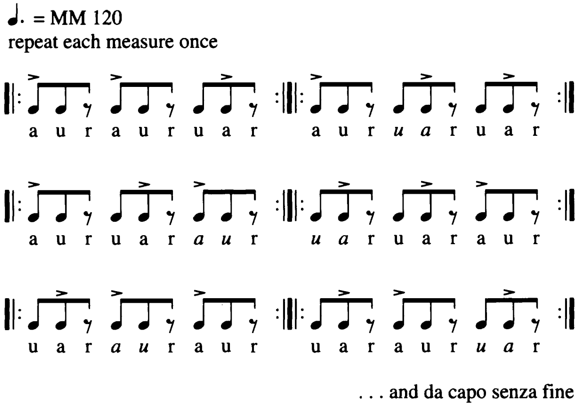
For a Cohn function, N must be greater than 2. (A function mod 2 has only *one* adjacent-argument location.) For the rest of this article, in considering functions mod N, we shall assume that N is 3 or greater.

Functions that represent Cohn sets are Cohn functions in a natural way. For instance, let us consider the function (ynynynnynynn), which represents the Cohn set {C,D,E,G,A}. An exchange of differing (NB) values at the adjacent (NB) arguments 4 and 5 (mod 12) yields the new function (ynyn*n*ynynynn), which represents the pcset {C,D,F,G,A}. If we start with the rightmost y of the new function as written, and read the new function circularly backwards therefrom, we obtain (ynyn*yn*nynynn), which is the original function (ignoring italics). So the orginal function, (ynynynnynynn), can be Cohn-flipped at arguments 4 and 5. It can also be Cohn-flipped at arguments 11 and 0: the exchange of differing values at those two adjacent arguments produces the new function (*n*nynynnynyny), which represents the pcset {B,D,E,G,A}. Reading the new function backwards as it stands on the page—we do not have to do any rotating here—we obtain (*y*nynynnynyn*n*), which is the original function (ignoring italics). So our function can be Cohn-flipped in two different ways—at arguments 4 and 5, or at arguments 11 and 0. It is a Cohn function.

If we represent the transpositional set class of {C,D,E,G,A} by the function (22323), an interval-valued function mod 5, then that function is a Cohn function. It Cohn-flips at arguments 1 and 2: the exchange of values here produces (2*3*223), which, when readbackwards and suitably rotated, produces (22*3*23), the original function. It also Cohn-flips at arguments 4 and 0 (mod 5): the exchange of values there produces (*3*232*2*), which, when readbackwards, produces (*2*232*3*), the original function.

The study of Cohn functions is thus one way to generalize the study of Cohn sets. (It is by no means the only way!) As we shall see, we can catalogue all possible Cohn functions, so the generalization is fruitful. It is also useful because we have made no stipulations whatever upon the things that may be the *values* of a Cohn function. The values may be the symbols "yes" and "no"; they may be various intervals mod 12; they may be various dynamics, or register-numbers, or characteristic markings like {tenuto, staccato, slurred, détaché, jété}, or a myriad of other such musically pertinent sorts of things.

For one example, let us use the symbols a, u, and r to denote "accented," "unaccented," and "rest." A repeating accent-pattern in compound triple meter can then be represented by the mod 9 function (auraururar). This is a Cohn function. It Cohn-flips at arguments 0 and 1: (*ua*rauruar) retrograde-rotates as (*au*rauruar). It also Cohn-flips at arguments 3 and 4: (aur*ua*ruar) retrograde-rotates as (aur*au*ruar). Example 1 shows a Cohn cycle that reflects this structure, realizing it as a passage for drum.[2]



Example 1

. . . D] F a C e G b D [F . . .
M m M m M m p

Example 2

Here is another Cohn function that goes beyond normal pcset theory. It involves pitch-class structures in just intonation. Let the symbols M, m, and p represent respectively a just major third (5:4), a just minor third (6:5), and a Pythagorean minor third (32:27, a syntonic comma less than a just minor third). The exact numerical ratios are not the point here; the point is that these categories correspond conceptually to aspects of just scales as promulgated widely during the 19th Century, beginning with Moritz Hauptmann.[3] The top row of Example 2 shows the pitch classes of the just C major scale as Hauptmann lays them out.

The bottom row of the example shows the succession of intervals for the series of pitch classes. Upper-case pcs relate among themselves by fifths; so do lower-case pcs. Upper-case pcs relate to adjacent lower-case pcs by just thirds. The interval dF would be a just minor third, but lower-case d is not in Hauptmann's C-major system; it is pre-empted by upper-case D, the fifth of G. Instead of dF, the system includes the dyad DF, which spans a Pythagorean minor third, reflecting the construction of D as three fifths from F.

The function we wish to consider is the interval-series (MmMmMmp), a mod 7 function. It is a Cohn function. It Cohn-flips at arguments 0 and 6: (*p*mMmMm*M*) retrogrades as (*M*mMmMm*p*), the original function (ignoring italics). We shall call this "the M/p flip." The function also

(a)	. . . D]	F		a		C		e		G		b		D		[F . . .
			M		m		M		m		M		m		p	
			p		m		M		m		M		m		M	
	. . . D]	f♯		a		C		e		G		b		D		[f♯ . . .
(b)	. . . D]	F		a		C		e		G		b		D		[F . . .
			M		m		M		m		M		m		p	
			M		m		M		m		M		p		m	
	. . . d]	F		a		C		e		G		b		d		[F . . .

Example 3

Cohn-flips at arguments 5 and 6: (MmMmM*pm*) retrograde-rotates as (MmMmM*mp*). We shall call this "the m/p flip."

Cohn-flips of the interval-series correspond to modulations of the just C-major system. Example 3a shows how the system becomes a just e-minor system by the M/p flip. (e, a, and b triads are all just in this system.) And example 3b shows how the system becomes a just a-minor system by the m/p flip. (a, d, and e triads are all just in this system.) From this phenomenon we can construct an "infinite Cohn-cycle"—better, a "Cohn-chain"—of modulating just major and minor keys: . . . dm, FM, am, CM, em, GM, bm, DM, f♯m, ... Every pair of adjacent keys in this chain is related via a pertinent Cohn-flip of the interval-series involved.

Here is yet another 3-valued Cohn function of musical interest. It is the mod 7 function (xyzxzyz). It Cohn-flips at arguments 1 and 2: (x*zy*xzyz) retrograde-rotates into (x*yz*xzyz). It also Cohn-flips at arguments 0 and 6: (*z*yzxzy*x*) is the retrograde of (*x*yzxzy*z*).

If we use the arguments mod 7 to label successive pitch classes of the C major scale, then the arguments that receive the value x, for the function (xyzxzyz), are arguments 0 and 3, the numbers that label pcs C and F. The arguments that receive the value y are 1 and 5, the numbers that label pcs D and A; while the arguments that receive the value z are 2, 4, and 6, the numbers that label pcs E, G, and B. The Cohn function can thus represent a partition of the scale into three pcsets: X={C,F}, Y={D,A}, and Z={E,G,B}. Under this representation, the Cohn flips of the function correspond to "Cohn flips" of the partition as a whole. By exchanging the adjacent pcs D and E (arguments 1 and 2 mod 7), one transforms the XYZ partition into the diatonically-inverted partition X′={C,F}, Y′={E,A}, Z′={D,G,B}. And by exchanging the adjacent pcs C and B (arguments 0 and 6 mod 7), one transforms the XYZ partition into the diatonically-inverted partition X″={B,F}, Y″={D,A}, Z″={E,G,C}.

We shall call such a partition a *Cohn partition* (of its mod N "scale").

The Cohn structure of the function induces a Cohn cycle of partitions, as displayed in example 4.

Other interesting applications can be found for 3-valued Cohn functions. David Clampitt (in an unpublished letter) pointed out that the interval series for many normally-ordered pcsets mod 12 are 3-valued Cohn functions. So, for example, pentachords of Forte-types 5-3, 5-6, and 5-7 have respective interval-series (11217), (11316), and (11415); each series can be represented by the mod 5 Cohn function (xxyxz). The Cohn structuring gives rise to "modulating" cycles among prime and inverted forms of the sets. Example 5 shows how this works in the case of Forte-set 5-7.

The transition from each stage to the next, in example 5, replaces the "plagal" division of some dyad (e.g. C♯-D-F♯) by a corresponding "authentic" division (e.g. C♯-F-F♯), or vice-versa. The cycle of the example transposes each pcset down a semitone at every alternate stage, so it will cycle through all 24 unordered forms of set 5-7. Other such cycles will be shorter, e.g. that shown in example 6 for some sets of class 5-32.

	{C,F} {D,A} {E,G,B}	(exch. D with E) ⟶	{C,F} {E,A} {D,G,B}	(exch. F with G) ⟶	{C,G} {E,A} {D,F,B}
(exch. A with B) ⟶	{C,G} {E,B} {D,F,A}	(exch. C with D) ⟶	{D,G} {E,B} {C,F,A}	(exch. E with F) ⟶	{D,G} {F,B} {C,E,A}
(exch. G with A) ⟶	{D,A} {F,B} {C,E,G}	(exch. B with C) ⟶	{D,A} {F,C} {B,E,G}	(where we started)	

Example 4

	{C,C♯,D,F♯,G} 1 1 4 1 5	⟶	{C,C♯,F,F♯,G} 1 4 1 1 5	⟶	{C,C♯,F,F♯,B} 1 4 1 5 1				
⟶	{C,E,F,F♯,B} 4 1 1 5 1	⟶	{C,E,F,B♭,B} 4 1 5 1 1	⟶	and so forth				

Example 5

{C,C♯,E,F♯,A} → {C,C♯,E,G,A} → {C,D♯,E,G,A}
1 3 2 3 3   1 3 3 2 3   3 1 3 2 3

→ {C,D♯,E,G,B♭} → {C,D♯,F♯,G,B♭} → and so forth
3 1 3 3 2   3 3 1 3 2

Example 6

{B,C,D} $\xrightarrow{\text{re-dividing the m3}}$ {B,C♯,D} $\xrightarrow{\text{re-dividing the semitone}}$
1 2 9   2 1 9

{E,C♯,D} $\xrightarrow{\text{re-dividing the m3}}$ {E,C♯,D♯} $\xrightarrow{\text{re-dividing the semitone}}$
9 1 2   9 2 1

{E,F♯,D♯} → and so forth
2 9 1

Example 7

Each pcset in the cycle of example 6 is transformed by T3 two stages later, so the entire cycle will enchain eight forms of set 5-32—four prime and four inverted.

Clampitt, in his letter, listed exhaustively the many pcset classes that have interval series representable by 3-valued Cohn functions. All trichords whose interval-series have three distinct values are such. Example 7 demonstrates for the interval series (129).

In the example, the first transition corresponds to a Cohn flip of the pcsets involved—C♯ is adjacent to C as a pitch class. But the second transition is not a Cohn flip of pcsets—E is not adjacent to B as a pitch class. Nevertheless the transition *is* generated by an exchange of distinct adjacent values in the *interval series*, 2 and 9 being adjacent values for the mod 3 function (219).

Indeed any 3-valued function mod 3 is de facto a Cohn function. (xyz), in all generality, Cohn-flips at arguments 0 and 1: (*yx*z) is a rotated retrograde of (*xy*z). (xyz) also Cohn-flips at arguments 0 and 2: (*z*y*x*)is the retrograde of (*x*y*z*). Furthermore, (xyz) *also* Cohn-flips at arguments 1 and 2: (x*zy*) is a rotated retrograde of (x*yz*). (xyz) thus Cohn-flips at *three* different adjacent-argument pairs.

Any 2-valued function mod 3 is also de facto a Cohn function. (xxy), in all generality, Cohn-flips at arguments 1 and 2; it also Cohn-flips at arguments 2 and 0. It does not, however, Cohn-flip at arguments 0 and 1, because the values of the function there are not (NB) *distinct*.

It follows that the interval series for *any* trichord mod 12, the augmented triad excepted, is a Cohn function mod 3. So any trichord, except for the augmented triad, can be cycled in the manner of example 7.

Example 8 illustrates this for a 2-valued interval series.

Since the 3-valued Cohn function (xyz) Cohn-flips about *any* pair of arguments, example 7, whose trichord has a 3-valued interval series, can be expanded to a much more complex graph. Example 9 shows a portion of the hexagonal graph; the interval series Cohn-flips one way vertically, another way diagonally up-right-down-left, and yet another way up-left-down-right.[4] The reader is invited to explore how the graph is surfed by consecutive forms of Forte-set 3-2 in J.S. Bach's chorale prelude on "Durch Adam's Fall," or in the f♯m fugue from Book I, WTC.

Cohn functions can have more than three distinct values. Indeed there is no upper limit on the number of distinct values a Cohn function may

{C,C♯,D} → {C,C♯,B} → {C,B♭,B} → {A,B♭,B}
1 1 10   1 10 1   10 1 1   1 1 10

→ and so forth

Example 8

{E,F,D}
{E♭,F,D} {E,C♯,D}
{E♭,E,C♯} {E♭,C,D} {B,C♯,D}
{E♭,C,C♯} {B,C,D} {B,C♯,A♯}
{B♭,C,C♯} {B,C,A} {B,G♯,A♯}
{B♭,B,C♯} {B♭,C,A} {B,G♯,A} {G,G♯,A♯}
{B♭,G,A} {F♯,G♯,A} {G,G♯,F}
{F♯,G,A} {F♯,G♯,F}

connection re-divides the semitone.

connection re-divides the minor third.

connection re-divides the whole tone.

Example 9

have. But Cohn functions with more than three distinct values come with a very special structure.

To illustrate, let us consider the two-valued mod 3 "root function" f1 = (yyn), where y and n stand for "yes" and "no" respectively. From f1 we derive a two-valued mod 6 Cohn function f2 = (yn yn ny). The blanks have no mathematical significance; they are only there to structure the visual field suggestively. The consecutive values of f2 are the values of the root function f1, alternating with their respective antitheses. f2 can thus be regarded as a series of three "antithesis pairs."

To proceed farther, we next select some palindromic string of values—call it P. P may be any palindrome whatsoever. For instance P might be null; or P might consist of the lone symbol y, or the lone symbol n, or the lone symbol (say) r, where r is a new symbol. Or P might consist of the palindrome nn, or yy, or rr. Or P might consist of the palindrome qxvryynnnabcbannnyyrvxq. And so forth. Let L be the length of P.

From f2 we now form f3 = (yn P yn P ny P). f3 is a function mod(6+3L). f3 is a Cohn function. Specifically, the exchange of distinct values for f3 at arguments 0 and 1 yields (*ny* P yn P ny P), which is a rotated retrograde of (*yn* P yn P ny P), which is f3 itself, ignoring italics. And the exchange of distinct values for f3 at the adjacent arguments 2+L and 3+L yields (yn P *ny* P ny P), which is a rotated retrograde of (yn P *yn* P ny P), which is f3 itself, ignoring italics.

In case the palindrome P is void (L=0), f3 is simply f2. Reinterpreting y as "low" and n as "high," we can use f2 = (low high low high high low) to model thematic aspects of the Hauptmotiv for the finale of Schoenberg's *Third Quartet* Op. 30. The motiv, heard in the first violin at measure 1, presents pitches E5, D♯6, E5, D♯6, C♯6, D5 as four eighths and two quarters, bowed in successive pairs. The violin presents another version of the same contour motiv during measure 2.

In case P is the palindrome comprising the one symbol r, f3 is the mod 9 Cohn function (yn r yn r ny r). If we interpret y to mean "accented" and n to mean "unaccented," and r to mean "rest," then f3 has the same structure as the function (aurauruar), studied earlier in connection with example 1 (p. 185).

In case P is the palindrome comprising the symbol-string nn, then f3 is the mod 12 function (yn nn yn nn ny nn). Reading the symbols as "yes" and "no" in the familiar way mod 12, we find that f3 here can represent the Cohn set {C,E,A}. And so forth.

We could also start with the two-valued mod 5 "root function" f1= (yynyn). We derive the two-valued mod 10 Cohn function f2 by alternating f1's values with their respective antitheses: f2=(yn yn ny yn ny). And then, P being any palindrome of length L, we trope each antithesis-pair of f2 by an inserted P, thereby obtaining the Cohn function f3=(yn P yn P ny P yn P ny P). f3 is a Cohn function mod (10+5L).

In general, if we start from any of certain "root functions" mod K and any palindrome P of length L, we can derive Cohn functions mod (K (2+L)) in this fashion. The mathematical niceties will be provided later. As we shall note, K must be odd. An appropriate "root function" f1 is not necessarily itself a Cohn function mod K, but it can be derived from a very specific function f0 mod K in a way we shall examine later. f0 is a Cohn function and much more.

Part 3: prolegomena

In this and the following parts of the present essay, we shall construct and catalogue all possible Cohn functions. There are only a few generic types of them, and we have already studied specific examples of each possible type.

In parts 4 through 7 we shall construct each generic type of Cohn function. Part 8 will then explain in what sense every possible Cohn function must be one of the already-studied types.

To simplify matters we shall consider only those Cohn functions which are "0/1 situated." A Cohn function is such when one of its Cohn flips occurs at the arguments 0 and 1, mod N. It is clear that Cohn functions, in general, are the various rotations of 0/1 situated Cohn functions.

Part 4: Two-valued *generator-type* Cohn functions

The first type of Cohn function we shall construct is what we shall call the "two-valued generator type." Example 10 illustrates a particular specimen.

The top row of example 10(a) lists the numbers mod 12 as multiples of 5. 5 is the "generator" here, in a technical sense to be described later. The bottom row of example 10(a) assigns to each argument j mod 12 the value f(j)=x or f(j)=y, depending on where j comes in the list of arguments that lie above. If j lies between 5 and 1 inclusive on the list, the function assigns to argument j the value f(j)=x. If j lies between 6 and 0 inclusive on the list, the function assigns to argument j the value f(j)=y. A vertical divider on example 10(a) emphasizes that structure.

Example 10(b) "unscrambles" (a), rewriting the arguments j in their usual order, and permuting the list of corresponding (underlying) function values accordingly.

In the form of example 10(b) we can recognize f as a Cohn function. The exchange of distinct values at arguments 0 and 1 yields (*xy*yxyx yyxyxy), which retrogrades into (*yx*yxyxyyxyxy), which is f. And the exchange of distinct values at arguments 5 and 6 yields (yxyxy*yx*yxyxy), which retrogrades into (yxyxy*xy*yxyxy), which is f.[5]

If we interpret y as "yes" and x as "no," then f can represent the G a)

j =	5	10	3	8	1	6	11	4	9	2	7	0
f(j) =	x	x	x	x	x	y	y	y	y	y	y	y

b)

j =	0	1	2	3	4	5	6	7	8	9	10	11
f(j) =	y	x	y	x	y	x	y	y	x	y	x	y

Example 10

major diatonic set {C,D,E,F♯,G,A,B}, a Cohn set mod 12. (Replacing C by C♯, or F♯ by F, produces a suitable inverted diatonic set.)

The construction just studied, using the generator 5 mod 12, can be generalized to an analogous construction using any generator G mod N. We shall need to define, first, what is meant by a "generator mod N."

DEFINITION 4.1: By a *generator mod N* we shall understand an integer G, between 1 and N-1 inclusive, whose multiples mod N exhaust all the numbers mod N. That is, the set {G,2G,3G, . . . ,(N-1)G,NG=0}, when we reduce each of its numbers mod N, includes each number mod N.

Since there are N listed multiples of G above, and N distinct numbers mod N, each number mod N will appear once and only once on the list.

The generators mod 12 are 1, 5, 7, and 11.

A generator G mod N necessarily possesses several other interesting properties we shall now survey. Any number mod N which possesses any one of these properties will be a generator.

DEFINITION 4.2: The *[additive] order* of a number j mod N is the smallest positive integer K for which Kj = 0 mod N (i.e. for which N divides Kj).

Since Nj=0 mod N, the order K of j is always either equal to N or smaller than N.

For G to be a generator mod N, it is both necessary and sufficient that the order of G be N itself (and not any smaller number). For those interested, a proof will soon be supplied.

Also: for G to be a generator mod N, it is both necessary and sufficient that the integers G and N be relatively prime (have no common divisor greater than 1).

DEFINITION 4.3: A *unit* mod N is a number u mod N such that 1 is some multiple of u, mod N. That is, there is some v mod N satisfying uv=1 mod N. So, for example, 7 is a unit mod 10 (since 7 times 3 is 1 mod 10).

The numbers 1, 5, 7, and 11 are all units mod 12—each of those numbers, multiplied by itself, produces 1 mod 12.

For G to be a generator mod N, it is both necessary and sufficient that G be a unit.

We sum up our observations so far in a tighter form.

a) <G, 2G, . . . , MG=1, 1+G, 1+2G, . . . ,(N-1)G, NG=0>
The list is called "<FROM G TO 0 STEP G>"

b) <G, 2G, . . . , MG=1>
The list is called "<FROM G TO 1 STEP G>"

c) <1+G, 1+2G, . . . , (N-1)G, NG=0>
The list is called "<FROM 1+G TO 0 STEP G>"

Example 11

THEOREM 4.4: Conditions (a) through (d) below are all logically equivalent. Each condition implies all the others.

(a) G is a generator mod N.
(b) The order of G, mod N, is N.
(c) As integers, G and N are relatively prime (have no common divisor bigger than 1).
(d) G is a unit mod N.

We shall accept this theorem as valid. For readers who are curious, Appendix A (p. 213) gives a proof.

Having devoted some study to the properties of generators mod N, we now continue with the study of two-valued generator-type Cohn functions.

In example 10, we began by listing the numbers mod 12 as consecutive multiples of the generator 5. In the general case where G is a generator mod N, we shall begin analogously, listing the numbers mod N as consecutive multiples of the generator G, starting with G and ending with 0=NG. We obtain the list of example 11(a).

Here M is the number such that MG=1. Then (M+1)G = MG+G = 1+G, (M+2)G = MG+2G = 1+2G, and so forth.

The numbers on the list add an increment of G, mod N, at each successive entry. As indicated on example 11(a), we shall call the list "<FROM G TO 0 STEP G>," understanding "mod N" where that does not need to be specified.

We shall be devoting special attention to that part of the list which extends from G through 1 inclusive. This sublist is shown in example 11(b); it will be called "<FROM G TO 1 STEP G>." We shall also be devoting special attention to that part of the big list which extends from 1+G through 0 inclusive. This sublist is shown in example 11(c); it will be called "<FROM 1+G TO 0 STEP G>."

A special note is in order. Example 11 shows entries of 2G, 1+2G, and the like in its visual field. In some cases, these numbers may not lie within the indicated spans. The notation "<G, 2G, . . . , MG=1>" means:

"Start with G, and keep adding G's until you get to 1; then stop." Now the number G=1 is always a generator mod N; in that case M=1 and the list <FROM G TO 1 STEP G> comprises G alone—the number 2 will not be on the list. Similarly, G=(N-1) is always a generator mod N; in that case 1+G = 0 mod N and the list <FROM 1+G TO 0 STEP G> comprises 0 alone—the number 1+2G will not be on the list, nor will the number (N-1)G. For purposes of reasoning and/or computing, the *logical* definitions of the sublists should be used, not their visual appearances on example 11. *Logically*, <FROM G TO 1 STEP G> is that series which lists all consecutive multiples of G between 1G=G and MG=1 inclusive. When M=1 (if G=1), the list will contain only one entry. If M=2, the list will contain exactly two entries, namely 1G=G and 2G=MG=1. The following logical definitions of the lists at issue will be useful later, for precise reasoning.

DEFINITIONS 4.5: Given G a unit mod N, and the integer M between 1 and N-1 inclusive such that MG=1 mod N, then

The list <FROM G TO 1 STEP G> arranges in order those multiples mG of G such that the integer m lies between 1 and M inclusive.

The list <FROM 1+G TO 0 STEP G> arranges in order those multiples nG of G such that the integer n lies between M+1 and N inclusive.

The list <FROM G TO 0 STEP G> arranges in order those multiples nG of G such that the integer n lies between 1 and N inclusive.

The reader is reminded that we are always supposing N greater than 2, in our present investigations.

Example 12(a) copies over example 10(a). Reviewing it, we see that the Cohn function there was obtained by assigning the value x to every argument mod 12 <FROM 5 TO 1 STEP 5>, and assigning the value y to every argument mod 12 <FROM 6 TO 0 STEP 5>. The construction generalizes for any generator G mod N, as displayed in example 12(b). (This example supposes that the lengths of the pertinent sublists are sufficiently large to admit the numbers 2G etc. mod N.) That is, the function displayed in example 12(b) is a Cohn function. More formally, using the logical definitions of the sublists given in Definitions 4.2 above, we have the following theorem.

THEOREM 4.6: Given a generator G mod N, define the mod N function f as follows: for every argument j on the list <FROM G TO 1 STEP G>, set f(j)=x; for every argument j on the list <FROM 1+G TO 0 STEP G>, set f(j)=y.
Then f is a Cohn function. It Cohn-flips at arguments 0 and 1, and it Cohn-flips at arguments G and 1+G.

The theorem will not be proved here. An earlier draft of this article, in which I included formal proofs of this and other subsequent theorems, began stretching out to well over a hundred pages of typescript, most of

a)	j =	5	10	3	8	1	6	11	4	9	2	7	0
	f(j) =	x	x	x	x	x	y	y	y	y	y	y	y

b)	j =	G	2G . . .	(M-1)G	MG=1	1+G	1+2G . . .	(N-1)G	NG=0
	f(j) =	x	x . . .	x	x	y	y . . .	y	y

Example 12

them purely mathematical. I decided the readers of this journal, by and large, would not be happy with such text.

We next turn our attention to an interesting consequence of the theorem. We note first that a two-valued generator-type Cohn function can always be rotated into itself, after a suitable single exchange of distinct values at numerically adjacent arguments.

THEOREM 4.7: Given the situation of Theorem 4.6, let f′ be the function which exchanges the values of f at the adjacent arguments 0 and 1. Then f′ is a rotation of f (as well as a rotated-retrograde). Specifically, for all arguments j, f′(j) = f(j+G).

As above, a formal proof is not given here. The converse of Theorem 4.7 is also true.

THEOREM 4.8: Let f be a function mod N, with f(0) and f(1) distinct values; let f′ be defined by f′(0)=f(1); f′(1)=f(0); f′(j)=f(j) for other arguments j. Suppose that f′ is some rotation of f. Suppose specifically that G is the order-interval of rotation, so that f′(j)=f(j+G) for all arguments j mod N. Then G is a generator mod N, and f is a two-valued generator-type Cohn function. f Cohn-flips, specifically, at arguments 0-and-1, and also at arguments G-and-(1+G).

A formal proof is not provided here. We can combine Theorems 4.7 and 4.8 into a single theorem:

THEOREM 4.9: The following categories of function mod N are the same:

(a) a two-valued generator-type Cohn function, and
(b) a function which can be rotated into itself, after distinct values have been exchanged at some pair of numerically adjacent arguments.

Part 5: Three-valued generator type Cohn functions

Let N be an *odd* number. Let G be a generator mod N; let M be the positive integer less than N such that MG=1 mod N. Either M or N-M must be even, and only one of the two is even. (If both were odd, or if both were even, then N would be even, contrary to supposition.)

If N-M is even we can replace the generator G with the new generator

j:	G	2G	3G	...	1-2G	1-G	1	1+G	...	(N-1)G=-G	NG=0
f(j):	x	z	x	...	z	x	z	y	...	y	y

Example 13

G′ = N-G = -G mod N. Setting M′ = N-M, we see that M′G′ = (N-M) (-G) = -NG + MG = 1 mod N. And M′ is even. So, to consider the generic situation, it suffices to consider the case where M is even.

Example 13 is analogous to example 12(b) before. It displays a function f with three distinct values, x, y, and z. Like the function of example 12(b), f here takes on the value y for all arguments on the right side of the example, <FROM 1+G TO 0 STEP G>. The current f takes on two distinct values for G-multiples on the left side of the example. For *odd* integers m between 1 and M-1 inclusive, f assigns the argument mG the value x; for *even* integers m between 2 and M inclusive, f assigns the argument mG the value z. Since M itself is even, f does assume all three of the values: f(0) = y, f(G) = x, and f(MG) = f(1) = z. (M, which is a unit, cannot be 0; it must be 2 or greater.)

THEOREM 5.1: The function f of example 13 is a Cohn function. It Cohn-flips at arguments 0-and-1, and it Cohn-flips at arguments G-and-(1+G).

The proof is not given.

DEFINITION 5.2: A function of the sort just constructed will be called a *three-valued generator-type Cohn function.*

The function is "three-valued" because we are supposing x distinct from z. The machinery of the theorem would work perfectly well if z and x were equal, but in that case we would simply have a two-valued generator-type Cohn function of the sort examined in Part 4: example 13 would collapse into example 12(b).

The most general three-valued generator-type Cohn function is any rotation or retrograde-rotation of the generic Cohn function displayed in example 13.

NOTE 5.3. It will be well to stress once more that three-valued generator-type Cohn functions can be found or constructed *only* when N is odd. They can *always* be found when N is odd (and greater than 1), because in that case, given any generator G with MG = 1, either M or (N-M) must be even—and if (N-M) is even, it can be considered as an M' for the generator G′ = (N-1)G. These matters were discussed toward the beginning of Part 5 above.

For an example, let us consider N=9. Since 7 and 9 have no common

(a)

j:	7	5	3	1	8	6	4	2	0
f(j):	x	z	x	z	y	y	y	y	y

(b)

j:	0	1	2	3	4	5	6	7	8
f(j):	y	z	y	x	y	z	y	x	y

Example 14

divisor greater than 1, G=7 is a generator mod 9. Here M=4, since 4 times 7 = 1 modulo 9. Example 14(a) lists the numbers mod 9 in the format of example 13, and constructs the corresponding Cohn-function f as indicated. Example 14(b) "unscrambles" the numbers mod 9, putting them in their usual order, and writes beneath each j the corresponding value of f that lies below j on example 14(a). Reading off f-values in the order of example 14(b) we can denote f in our earlier symbolism as (yzyxyzyxy). We can then check directly that f is a Cohn function: (*zy*yxyzyxy) retrograde-rotates into(*yz*yxyzyxy), which is f, ignoring italics. And (yzyxyzy*yx*) retrograde-rotates into (yzyxyzy*xy*), which is f, ignoring italics. As our abstract work has told us, f Cohn-flips at 0-and-1, and also at 7-and-8.

As another example, let us take N=7 (remember, N must be odd) and G=2. (Since 2 and 7 are relatively prime, 2 is a generator mod 7.) In this case M=4, which is even, so we can construct the generic three-valued Cohn function in this particular case as follows: for j = 2, 4, 6, and 1, f(j) = x, z, x, and z respectively; for j = 3, 5, and 0, f(j) = y. Thus for j = 0, 1, 2, 3, 4, 5, and 6, f(j) = respectively y, z, x, y, z, y, and x. In our earlier symbolism, f is the function (yzxyzyx). f can model a partition of the C major scale into sets X, Y, and Z: X comprises just those notes j whose f-values are x, and so forth. Thus X = {E,B}, Y = {C,F,A}, and Z = {D,G}. This is a form of the "Cohn-partition" studied earlier, a partition of the scale into a triad and two fifths. Cohn-flipping the function at arguments 0-and-1 models exchanging notes 0-and-1 (C and D) between pertinent sets of the partition: this yields the new partition X′ = {E,B}, Y′ = {D,F,A}, Z′ = {C,G}, a diatonic inversion of the original partition. Cohn-flipping the function at arguments G-and-(1+G), i.e. at arguments 2-and-3, models exchanging notes 2-and-3 (E and F) between pertinent sets of the partition: this yields the new partition X″ = {F,B}, Y″ = {C,E,A}, Z″ = {D,G}, a diatonic inversion of the original partition.

Not every three-valued Cohn function is of generator type. Some are of the type we shall study next, which we shall call the "antithesis-palindrome type." The rhythmic function of example 1, which was three-valued, was of that type. We discussed this informally at the end of Part 2. There, we noted as well that Cohn functions of the antithesis-palindrome type may have an indefinitely large number of distinct values. In Parts 6 and 7, which follow, we shall explore that. Before studying the full antithesis-palindrome type of function, in Part 7, we shall first study the pure antithesis-pair type in Part 6. Here the palindrome is null, and this type of function has only two values.

Part 6: The (two-valued) antithesis-pair type of Cohn function

We shall construct here the most general Cohn function f, of antithesis-pair type, that Cohn-flips at arguments 0-and-1, and that has symbolic values x and y, with f(0)=y and f(1)=x.[6] We start with an *odd* number K, greater than or equal to 3.

Example 15, stage 0, displays a certain function f0 mod K. The arguments for f0 are listed as the multiples of 2, mod K, in order <FROM 2 TO 0 STEP 2>. The integer K being odd, the integer K-1 will be even; a vertical divider is placed on the example after K-1. Then, since K-1 is the same as -1 mod K, the next entry on the argument list of the example is (-1)+2 or 1 mod K. Successive increments of 2 then continue the list through the odd integers from 1 to K inclusive, where of course K=0 mod K. The values of f0 are given as f(k)=x or f(k)=y, depending on whether the integer k, lying between 1 and K inclusive, is even or odd.

The reader's attention is drawn once more to the restriction *that K be odd*. Since 2 and K have no common divisors greater than 1, 2 is a generator mod K, and the format of the example lists *all* the numbers mod K. (If K were even, the multiples of 2 mod K would list only half the numbers mod K—just the even numbers mod K.)

Attention is also drawn to the fact that the function f0 has (K-1)/2 arguments with x-values and (K+1)/2 arguments with y-values. It thus assumes the value y exactly one more time, than it assumes the value x.[7]

Stage 1 of example 15 keeps the function values from stage 0 as they lie; it multiplies all the arguments of stage 0 by a unit (any unit) u mod K. A new function f1 mod K is thereby constructed. The arguments of f1 are here listed as consecutive multiples of the generator 2u mod K. (Since 2 and u are both units, so is 2u.) Arguments on the list <FROM 2u TO -u STEP 2u> have x-values; arguments on the list <FROM u TO 0 STEP 2u> have y-values. The function f1 will be called the *root function* for the Cohn function eventually generated. f1 itself need not necessarily be a Cohn function itself; it is of course generated in a very specific way from the very specific (Cohn) function f0. The root function f1, like the funcstage 0: f0 assigns the even integers, between 1 and K inclusive, the value x; it assigns the odd such integers the value y.

$$\begin{array}{cccc|ccccc} 2 & 4 & \dots & K\text{-}1 & 1 & 3 & \dots & K\text{-}2 & K{=}0 \\ x & x & \dots & x & y & y & \dots & y & y \end{array}$$

stage 1: The *root function* f1 multiplies the arguments of f0 by the unit u mod K.

$$\begin{array}{cccc|ccccc} 2u & 4u & \dots & \text{-}u & u & 3u & \dots & (K\text{-}2)u & 0 \\ x & x & \dots & x & y & y & \dots & y & y \end{array}$$

stage 2: f2 is defined on even arguments mod 2K, replacing u mod K in the root function by v=2u mod 2K. (E.g: if K=7 and u=5 mod 7 then v=10 mod 14.)

$$\begin{array}{cccc|ccccc} 2v & 4v & \dots & \text{-}v & v & 3v & \dots & (K\text{-}2)v & 0 \\ x & x & \dots & x & y & y & \dots & y & y \end{array}$$

stage 3: for each even argument j mod 2K in stage 2, assign the odd argument 1+j the antithetic f2-value (opposite to j's value).

$$\begin{array}{cccc|ccccc} 2v & 4v & \dots & \text{-}v & v & 3v & \dots & (K\text{-}2)v & 0 \\ 1+2v & 1+4v & \dots & 1\text{-}v & 1+v & 1+3v & \dots & 1\text{-}2v & 1 \\ x & x & \dots & x & y & y & \dots & y & y \\ y & y & \dots & y & x & x & \dots & x & x \end{array}$$

Example 15

tion f0, has (K-1)/2 arguments with x-values and (K+1)/2 arguments with y-values; it assumes the value y exactly one more time, than it assumes the value x.

Stage 2 of example 15 keeps the function values from stage 1 as they lie; it replaces the number u mod K by the number v=2u mod 2K. (For example: if K is 7 and u is 5 mod 7, v is 10 mod 14.) This process begins to construct a new function f2 mod 2K. The even arguments for f2 are here listed as consecutive multiples of the number 2v mod 2K. Arguments on the list <FROM 2v TO -v STEP 2v> have x-values; arguments on the list <FROM v TO 0 STEP 2v> have y-values. f2 is not yet completely defined: we know its values so far only for even arguments mod 2K.

Stage 3 of example 15 finishes the definition of function f2, by providing its arguments for odd values mod 2K. For each even argument j mod 2K, the odd argument 1+j is assigned the "antithetic" value, from the value assigned at argument j. That is, f2(1+j) is defined as y or x, according to whether f2(j) is x or y respectively.

stage 0: f0 assigns the even integers, between 1 and 7 inclusive, the value x; it assigns the odd such integers the value y.

2	4	6		1	3	5	7=0
x	x	x		y	y	y	y

stage 1: The *root function* f1 multiplies the arguments of f0 by the unit u=5 mod K=7. (3, 6, 2, 5, 1, 4, and 0 are respectively 5 times 2, 4, 6, 1, 3, 5, and 0 mod 7.)

3	6	2		5	1	4	7=0
x	x	x		y	y	y	y

stage 2: f2 is defined on even arguments mod 2K=14, replacing 5 mod 7 in the root function by 10 mod 14. All argument numbers mod 7 in stage 1 get multiplied by 2 mod 14 in stage 2.

6	12	4		10	2	8	14=0
x	x	x		y	y	y	y

stage 3: for each even argument j mod 14 in stage 2, assign the odd argument 1+j the antithetic f2-value (opposite to j's value).

6		12		4			10		2		8		0	
	7		13		5			11		3		9		1
x		x		x			y		y		y		y	
	y		y		y			x		x		x		x

Example 16

THEOREM 6.1: f2, as defined at stage 3 of example 15, is a Cohn function. It Cohn-flips, specifically, at arguments 0-and-1, and it also Cohn-flips at arguments v-and-(1+v).

For example, let us take K=7 and u = 5 mod 7, so that v = 10 mod 14. Stages 0 through 3 of example 16 then provide us with the antithesis-type Cohn function f2 mod 14 displayed at stage 3. We can write out f2 in the conventional notation used earlier, as (yx yx xy xy yx yx xy). The blanks here have no mathematical significance; they are only used to isolate the antithesis-pairs at even-and-succesive-odd arguments mod 14. By our construction, the Cohn function f2 will Cohn-flip at arguments 0-and-1, and also at arguments v-and-(1+v), i.e. at arguments 10-and-11. We can check that out by inspection: (*xy* yx xy xy yx yx xy) retrograde-rotates into (*yx* yx xy xy yx yx xy) ; while (yx yx xy xy yx *xy* xy) retrograde-rotates into (yx yx xy xy yx *yx* xy).

Part 7: antithesis-palindrome type Cohn functions.

Let us consider the function f2 just discussed in connection with example 16 at the end of Part 6 above. If P is any palindrome of any length, involving any symbols whatsoever, the pertinent adjacent-argument exchanges and retrograde-rotation behavior of f2 will not be changed in essence, should we insert the string P between successive antithesis-pairs. That is, the function f3 = (yx P yx P xy P xy P yx P yx P xy P) will be a Cohn function. By inspection: (*xy* P yx P xy P xy P yx P yx P xy P) retrograde-rotates into (*yx* P yx P xy P xy P yx P yx P xy P); while (yx P yx P xy P xy P yx P *xy* P xy P) retrograde-rotates into (yx P yx P xy P xy P yx P *yx* P xy P).

If the palindrome is of length L, then the expression above articulates into segments of length 2+L, each segment consisting either of "xy P" or of "yx P." There will be K such segments. So the "N" of the enlarged function "mod N" will be K times 2+L.

The idea is intuitively clear enough. Working out the specifics of the number-juggling involved is a bit fussy. The niceties will be handled in this part of the present essay.

We start with an odd number K and a unit u mod K, as before. Let P be any palindrome; let L be its length. If L=0 (so the palindrome is null), we have the case just discussed in Part 6 above, so we shall suppose here that L is 1 or greater. If L is 1, we shall denote the sole symbol of the palindrome by p1. If L is 2 or more, we shall use the symbols p1, p2, . . . to denote the successive symbols of the palindrome P = p1 p2 . . . p(L-1) pL, so that p1=pL, p2=p(L-1) and so forth. Let J = 2+L. Let N be K times J.

Stages 0 and 1 of example 17 proceed exactly as did stages 0 and 1 for example 15 earlier. At stage 2, where example 15 replaced u by v=2u mod 2K, example 17 more generally replaces u by v=Ju mod JK. This amounts to multiplying all the argument numbers mod K of stage 1 by J, to obtain the arguments mod JK listed in stage 2.

At stage 2, the function f2 is defined only for those arguments mod N=JK that are multiples of J. Stage 3, like stage 3 of example 15, assigns "antithesis" values to f2 for those arguments mod N that are 1-more-than-some-multiple-of-J.

We have yet to assign f2-values to arguments mod N that are neither multiples of J, nor 1 more than multiples of J. Stage 4 of example 17 assigns such values. "smoJ" stands for "some multiple of J." Thus if a number mod N is (smoJ)+0, its f2-value has already been assigned at stage 3. The same is the case for a number that is (smoJ)+1. If a number n is (smoJ)+2 (i.e. 2 more than some multiple of J), stage 4 assigns f2(n) the value p1, where p1 is the first symbol of the palindrome P. Since P is a palindrome, p1 is the same symbol as pL, the last symbol of the palinstage 0: f0 assigns the even integers, between 1 and K inclusive, the value x; it assigns the odd such integers the value y.

2	4	...	K-1	1	3	...	K-2	K=0
x	x	...	x	y	y	...	y	y

stage 1: The *root function* f1 multiplies the arguments of f0 by the unit u mod K.

2u	4u	...	-u	u	3u	...	(K-2)u	0
x	x	...	x	y	y	...	y	y

stage 2: f2 is defined on arguments that are multiples of J mod JK, replacing u mod K in the root function by v=Ju mod JK. (E.g: if K=7 and u=5 mod 7 and J=10, then v=50 mod 70.)

2v	4v	...	-v	v	3v	...	(K-2)v	0
x	x	...	x	y	y	...	y	y

stage 3: for each argument n mod JK in stage 2, assign the argument 1+n the antithetic f2-value (opposite to n's value).

2v		4v		... -v		v		3v		...(K-2)v		0	
	1+2v		1+4v		... 1-v		1+v		1+3v		...		1-2v	1
x		x		... x		y		y		... y		y	
	y		y		... y		x		x		...		x	x

stage 4: "smoJ" means "some multiple of J." The palindrome P, of length L = J-2, has symbols p1 (= pL), p2 (= p(L-1)), . . . , p(L-1) (= p2), and pL (= p1).

if the number mod N is	the f2-value there is
(smoJ) + 0	as assigned in stage 3
(smoJ) + 1	as assigned in stage 3
(smoJ) + 2	p1 (= pL)
(smoJ) + 3	p2 (= p(L-1))
.	.
.	.
.	.
(smoJ) + (J-2)	p(L-1) (= p2)
(smoJ) + (J-1)	p(L) (= p1)

Example 17

drome. The fact is noted on the visual display for stage 4. If a number n is (smoJ)+3 (i.e. 3 more than some multiple of J, should there be any such), stage 4 assigns f2(n) the value p2, where p2 is the second symbol of the palindrome P. Since P is a palindrome, p2 is the same symbol as p(L-1), the penultimate symbol of the palindrome. The fact is noted on the visual display for stage 4. And so forth: if a number n is (smoJ)+(r+1), where the integer r is between 1 and L inclusive, stage 4 assigns f2(n) the value pr, the r-th symbol of the palindrome. Since P is a palindrome, pr is the same symbol as p(L+1-r).

We see from the visual display that an argument mod N which is (smoJ)+2 is assigned the same f2-value, as is an argument which is (smoJ)+(J-1); an argument which is (smoJ)+3 (should there be any) is assigned the same f2-value, as an argument which is (smoJ)+(J-2); and so forth.

THEOREM 7.1: f2, as fully defined at stage 4, is a Cohn function. It Cohn-flips at arguments 0-and-1; it also Cohn-flips at arguments v-and-(1+v).

The proof is not given.

The function f2, with values as at stage 4 of example 17, can be constructed for any N, with a free choice of unit u mod K, provided only that K be an odd divisor of N (greater than 1). The function can therefore be constructed for any N which has a proper odd divisor, which is to say for any N that is neither a power of 2 nor a prime number.

Example 18 provides a specific case of the generic structure. Stages 0 and 1 copy over the same stages from example 16, taking K=7 and u=5 mod 7. Now we take a palindrome P of length L=4 —let's give it the symbols pqqp. Then J = L+2 = 6, and N = JK = 42.

Stage 2 of example 18 multiplies the numbers mod 7 of stage 1 by J=6 mod 42. Stage 3 thus assigns f2-values to all numbers that are some multiple of 6 (smo6) mod 42. Stage 3 then assigns f2-values to all numbers that are (smo6)+1 mod 42. If n is smo6 mod 42, stage 3 assigns 1+n the f2-value antithetic to n's value.

Stage 4 of example 18 completes the assignment of f2-values by assigning numbers that are (smo6)+2, or (smo6)+3, or (smo6)+4, or (smo6)+5, the f2-values p or q or q or p respectively, i.e. the consecutive symbols of the palindrome p q q p. On the lower right side of stage 4, the palindrome can be read downwards.

All the values of f2, for every argument mod 42, can now be read off stages 3 and 4 of example 18. f2 can be represented in our standard notation by (yx pqqp yx pqqp xy pqqp xy pqqp yx pqqp yx pqqp xy pqqp). As earlier, the blanks in the notation have no mathematical significance; they are only there to articulate in the visual layout the antithesis-cum-palindrome structuring.

stage 0: f0 assigns the even integers, between 1 and 7 inclusive, the value x; it assigns the odd such integers the value y.

2	4	6	1	3	5	7=0
x	x	x	y	y	y	y

stage 1: The *root function* f1 multiplies the arguments of f0 by the unit u=5 mod K=7. (3, 6, 2, 5, 1, 4, and 0 are respectively 5 times 2, 4, 6, 1, 3, 5, and 0 mod 7.)

3	6	2	5	1	4	7=0
x	x	x	y	y	y	y

stage 2: f2 is defined on arguments that are some multiple of J=6 mod 42, replacing u=5 mod 7 in the root function by Ju=30 mod 42. All argument numbers mod 7 in stage 1 get multiplied by 6 mod 42 in stage 2.

18	36	12	30	6	24	42=0
x	x	x	y	y	y	y

stage 3: for each argument n that is some multiple of 6 (smo6) mod 42 in stage 2, assign the argument 1+n, that is (smo6)+1, the antithetic f2-value (opposite to n's value).

18		36		12		30		6		24		0	
	19		37		13		31		7		25		1
x		x		x		y		y		y		y	
	y		y		y		x		x		x		x

stage 4: "smo6" means "some multiple of 6." The palindrome P, of length 4 = J-2, has consecutive symbols p, q, q, and p.

if the number mod 42 is	the f2-value there is
(smo6) + 0	as assigned in stage 3
(smo6) + 1	as assigned in stage 3
(smo6) + 2	p
(smo6) + 3	q
(smo6) + 4	q
(smo6) + 5	p

Example 18

Part 8: The above types of Cohn function are exhaustive.

We have now explored several generic sorts of Cohn functions (that Cohn flip at arguments 0 and 1). In Parts 4 and 5 we explored two-valued and three-valued generator-type Cohn functios. In Part 7 we explored antithesis-palindrome type Cohn functions, a species that subsumes the functions of Part 6 as special cases.

The following fact is true: Any Cohn function must be either of generator type (two-valued or three-valued), or of antithesis-palindrome type. We have therefore now constructed generic examples for all possible types of Cohn functions.

There will be no formal proof of that fact here—as I stated earlier, I am eschewing such proofs in this paper, to conserve space and to avoid wearying readers who have limited tolerance for formal mathematical prose. Still, it will be useful to break the assertion down into two more specific theorems.

THEOREM 8.1: Let f be a Cohn function mod N that Cohn-flips at arguments 0-and-1, and that also Cohn-flips at arguments G-and-(1+G). Suppose that G is a generator mod N. Then f is a generator-type Cohn function, as specifically constructed in Parts 4 and 5 earlier. If N is even, f must be two-valued. If N is odd, f may be two-valued or three-valued.

THEOREM 8.2: Let f be a Cohn function mod N that Cohn-flips at arguments 0-and-1, and that also Cohn-flips at arguments v-and-(1+v). Suppose that v is *not* a generator mod N. Then f is an antithesis-palindrome type Cohn function, as specifically constructed in Part 7 earlier. The number K of that construction is the order of v—defined earlier as the smallest positive integer satisfying Kv=0 mod N. (Since v is not a generator, K will be strictly less than N.) K must be odd. K divides N properly, and the multiples of v, mod N, are exactly the multiples of J mod N, where J = (N/K). Should J be greater than 2, then there is a palindrome P, of length J-2, such that consecutive values of f at kv+2, kv+3, . . . , kv+(J-2), and kv+(J-1) are the consecutive symbols of the palindrome—k being any integer from 0 to K-1 inclusive. The number u = (v/J) mod K is a unit mod K, and the stages of example 18 can be read backwards, to arrive at the root function f1 mod K, and thence to arrive at the underlying function f0 mod K, which simply assigns the value f(1) to the even integers mod K, from 2 to K-1 inclusive, and the value f(0) to the odd integers mod K, from 1 to K=0 inclusive.

The theorems are given without proof.[8]

Part 9: Some directions for further work (I)

One way of expanding the preceding work would be to study "order w" Cohn-flipping of functions mod N, where w is some number mod N.

holding {F,G♯}; moving B and D♯ by semitones:

F G♯ B D♯	F G♯ B D♯
F G♯ C D	F G♯ B♭ D

holding {G♯,B}; moving F and D♯ by semitones:

F G♯ B D♯	F G♯ B D♯
F♯ G♯ B D	E G♯ B D

holding {F,B}; moving G♯ and D♯ by semitones:

F G♯ B D♯	F G♯ B D♯
F A B D	F G B D

holding {B,D♯}; moving F and G♯ by semitones:

F G♯ B D♯
F♯ A B D♯

holding {D♯,F}; moving G♯ and B by semitones:

F G♯ B D♯
F A C D♯

holding {G♯,D♯}; moving F and B by semitones:

F G♯ B D♯
F♯ G♯ B♯ D♯

Example 19

A function has an "order w Cohn flip" at arguments Q-and-(w+Q) if an exchange of distinct (NB) function values at those two w-separated (NB) arguments gives a rotated retrograde of the original function. A function is an "order w Cohn function" if it can be order w Cohn-flipped in two different ways.

If w is a unit mod N, then the order w Cohn functions are simply the "circle of w transforms" of ordinary (order 1) Cohn functions. The situation is trickier if w is not a unit mod N, but the matter may still be studied by techniques similar to those used to perform the constructions and prove the theorems of Parts 4 through 8 above.

Another way of expanding the work arises from Douthett's voluminous but as yet unpublished work on various pcset relationships suggested by Cohn's original work. We can say that pcsets X and Y are in the "Douthett relation of degree 1" (DOUTH1) if Y can be obtained by discarding some member pitch class x of X, and then picking up some new pitch class y one semitone away from the discarded x.[9] X is then a Cohn set if it is in the DOUTH1 relation to two distinct inverted forms of itself. According to Theorem 4.9, X is in the DOUTH1 relation to some properly transposed form of itself if, and only if, it is a Cohn set arising from some two-valued generator-type Cohn function.

Douthett studied many generalizations of the DOUTH1 idea. For instance, we can say that pcsets X and Y are in the "Douthett relation of degree 2" (DOUTH2) if Y can be obtained by discarding two member pitch classes x1 and x2 of X, and then picking up two new pitch classes y1 and y2, where y1 lies one semitone away from the discarded x1 and y2 lies one semitone away from the discarded x2. In general, we can study Douthett relations of arbitrarily high degree D, within suitably large "scales" mod N. But DOUTH2 is already of ample interest, in the mod 12 context; it warrants extended study in its own right. For instance, as Douthett points out in private correspondence, any pcset forming a Tristan Chord or a dominant-seventh chord has a vast panoply of DOUTH2-relations with transposed and inverted forms of itself. Example 19 illustrates these, starting from the chord {F,G♯,B,D♯}.

As we see from the left-hand column, the chord is in DOUTH2-relation with its own transpositions at levels T3, T6, and T9, pivoting about its own ic3 and ic6 dyads.[10] As we see from the right-hand column, the chord is also in DOUTH2-relation with six of its inverted forms, namely the dominant-seventh chords built on F, G♯, B, E, G, and B♭.[11]

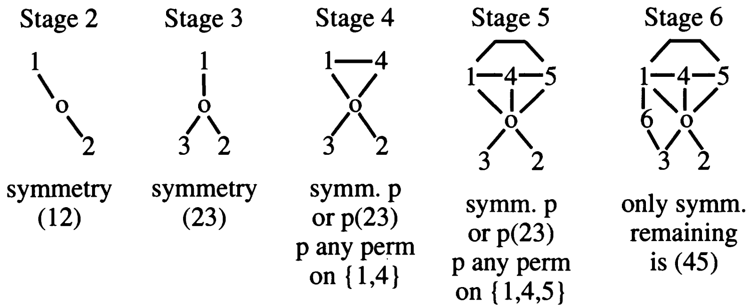
Douthett is particularly interested in features of the abstract graphs generated by such transitional possibilities. Beyond abstract graphs, contextual graphs that arise from specific passages can be of analytic interest. Let us consider, for instance, those Tristan and dominant-seventh chords that occur withing the first three phrases of *Tristan*. They can be listed as follows.

Chord 1: {F,B,D♯,G♯}
Chord 2: {E,G♯,D,B}
Chord 3: {G♯,D,F♯,B}
Chord 4: {G,B,F,D}
Chord 5: {C,F,A♭,D}
Chord 6: {B,D♯,A,F♯}.

The following DOUTH2-relationships obtain.

Chord 1 to any of Chords 2,3,4,5,6; and vice-versa.
Chord 2 to Chords 4 and 5; and vice-versa.
Chord 3 to Chords 4,5, and 6; and vice-versa.
Chord 4 to Chord 5; and vice-versa.

In this scheme, Chord 1, which is after all *the* Tristan Chord, has a special and unique position: it is DOUTH2-related to all five of the other



Example 20

Chords. Chord 2 has three DOUTH2-links (to Chords 1,4,5); Chords 3,4, and 5 each have four DOUTH2-links. Chord 6 is also special and unique in that it has the fewest number of DOUTH2-links in the context; it is linked only to Chords 1 and 3. The two most "tonally functional" Chords of the passage, Chords 2 and 6, thus have the fewest DOUTH2-links (three and two respectively).

Example 20 presents a suggestive way of "unfolding" the various DOUTH2 links over five successive temporal stages. The numbers on the example stand for their respective Chords; lines connect Chords that are in the DOUTH2 relation.

At Stage 2 (after we hear Chord 2), there is a perfect graphic symmetry: the line-structure of the graph is not affected if nodes 1 and 2 are permuted. At Stage 3 (after we hear Chord 3), the symmetry of nodes 1 and 2 is destroyed, while a new symmetry arises: the line-structure of the graph is not affected if nodes 2 and 3 are permuted, but other permutations of the nodes would change the graphic structure. (For instance, permuting nodes 1 and 2 would remove the line between node 1 and node 3, and would introduce a line between node 2 and node 3.) Stage 4 preserves and amplifies the (23) symmetry. It introduces a new symmetry between nodes 1 and 4. If p is any permutation on the symbols 1 and 4, then p and p(23) are symmetries of the graph. Stage 5 preserves and amplifies these symmetries: the line-structure of the Stage 5 graph is preserved by permutations p and p(23) if p is any permutation on the sybols 1,4, and 5. Stage 6, which strongly prepares a tonal cadence, spectacularly disrupts the highly symmetrical structure of the Stage 5 graph, along with the process of amplifying symmetry. At Stage 6, no symmetries remain except for (45).

Similar sorts of analytic commentary can be produced for other suitable Wagnerian passages. The curious reader may construct such a commentary, as practice, for the opening of the Immolation Scene (starting at

"Molto largamente e più lento"), where the following pertinent Chords are heard.

Chord 1: A C♯ E G
Chord 2: B♭ D♭ F G
Chord 3: B♭ C E G
Chord 4: B♭ D♭ F♭ A♭
Chord 5: B♭ D♭ E♭ G
Chord 6: B C♯ E G
Chord 7: B C♯ E♯ G♯

Chord 7, after being presented in the above DOUTH2 context, is then treated as a functional dominant of F# major, "resolving" to the Fire motive in that key. The reader who works out the exercise will find that the highly asymmetrical graph at Stage 7 shatters the highly symmetical graph at Stage 6, very much as the final stage of the *Tristan* analysis produced a relatively asymmetrical graph that shattered a highly symmetrical graph at the penultimate Stage there. In both cases the final Stages prepare big tonal cadences, sounding like "functional dominants" in that connection.[12]

An interesting complex of DOUTH2-like relationships involves forms of the hexachord 023468, a hexachord much used by Schoenberg—e.g. in the Wind Quintet, in *Von Heute auf Morgen*, and in the Violin Fantasy. Instead of DOUTH2 itself, we want to invoke here the analogous relation that involves fouths-or-fifths instead of semitones, that is DOUTH2 subjected to the "circle-of-fifths" transformation. We shall call the new relation DOUTHCF2. Sets X and Y are in the DOUTHCF2 relation if one can discard pitch classes x1 and x2 from X, and pick up new pitch classes y1 and y2 to form Y, in such a way that y1 is a fourth or fifth from x1, and y2 is a fourth or fifth from x2.

So, in example 21(a), the hexachord H = 023468 is in the relation to its 2-transpose H2. As suggested by the vertical underlay of the example, we can discard pcs 0 and 3 from H, and pick up new pcs 5 (a fifth away from pc 0) and t (a fifth away from pc 3), so as to obtain H2. In similar fashion, H2 is in DOUTHCF2-relation with H4, and H4 with H6, and so on. As one reads down the list of example 21(a), one passes through a complete T2-cycle of hexachord forms. This cycle is similar in structure to a Cohn cycle.

We are not yet through with example 21(a). It enables us to see that H is in the DOUTHCF2 relation with *every one* of its evenly-transposed forms. For instance, compare the H rank of the matrix to the H4 rank. We see that H4 can be obtained from H if we discard pcs 2 and 3 from H and pick up new pcs 7 and 8, a fifth away from pcs 2 and 3 respectively. Likewise we can obtain H6 from H if we discard pcs 3 and 4 from H and pick

(a)	H:	0 2 3 4 6 8
	H2:	5 2 t 4 6 8
	H4:	0 7 t 4 6 8
	H6:	0 2 t 9 6 8
	H8:	0 2 t 4 e 8
	Ht:	0 2 t 4 6 1
(and back to	H:	0 2 3 4 6 8)

(b)

H: 023468	H: 023468	H: 023468	H: 023468
h0: 09t468	h2: 02te68	h4: 02t148	ht: 72t468

(c)

H: 023468
discard 3; pick up t: 02t468
discard 8; pick up 3 again: 02t463, obtaining h6.

(d)

H: 023468
discard 3; pick up t: 02t468
discard t; pick up 5: 025468, obtaining h8.

Example 21

up new pcs t and 9, a fifth away from pcs 3 and 4 respectively. And so forth. So the DOUTHCF2-graph for the six forms of the hexachord is a complete graph. (Any pair of the hexachord-forms is connected by a line on the graph.)

Example 21(b) shows how H is also DOUTHCF2-related to four of its evenly-inverted forms. The inverted form "h0" inverts H about the pc 0; the forms h2, h4, etc. are h0 transposed by T2, T4, etc. respectively.

It seems a shame that H is not DOUTHCF2-related, as well, to h6 and h8. However, we can define an "extended DOUTHCF2 relation," EXT DOUTHCF2, which will in fact relate H with h6 and h8 as well. The extended relation works as follows. Given a pcset X, suppose that the algorithm below can be executed.

(i) Discard a member pc x of X.
(ii) (If possible,) pick up a fresh pc y1 from the "pack" (i.e. the complement of X), that lies a fifth away from the discarded x. Call the new pcset Z.
(iii) Put x back into the "pack" (i.e. the complement of Z, whence it might be picked up again at a later stage of the algorithm).
(iv) Discard a member pc z of Z. (We allow the possibility of choosing z=y1, of discarding the pc selected at stage (ii) above.)
(v) (If possible,) pick up a fresh pc y2 from the "pack" (i.e. the

complement of Z), that lies a fith away from the discarded z. Call the new pcset Y.

If the algorithm can be executed successfully, we shall say that pcsets X and Y are in the extended DOUTHCF2 relation EXTDOUTHCF2.[13]

It is easily verified that DOUTHCF2-related sets must be EXT DOUTHCF2-related.[14] So EXTDOUTHCF2 does indeed "extend" DOUTHCF2. Furthermore, any pcset X—except for the total chromatic—is EXTDOUTHCF2-related to itself.[15]

Example 21(c) now shows us how hexachord H is in the EXT DOUTHCF2 relation to its evenly-inverted form h6. And example 21(d) shows us how H is in the EXTDOUTHCF2 relation to its evenly-inverted form h8. So H is in the EXTDOUTHCF2 relation to *all* its evenly-inverted forms (via example 21 (b)(c)(d)), as well as all its evenly-transposed forms (via example 21(a)).

I have yet to find a pithy short passage for analysis, from any of Schoenberg's various works involving hexachord H, that works as neatly as the Wagner analyses above. But I am fairly sure that some such passages exist (I should guess most likely in the Wind Quintet). Or possibly, the DOUTHCF2 or EXTDOUTHCF2 relations might be suggestive in analyzing larger formal features of the works, features that might involve a variety of excerpts from temporally non-contiguous locations in the music.

Part 10: Some directions for further work (II)

Douthett's work, as I partially described it and elaborated it in Part 9 above, is only one direction in which Cohn's orginal impetus can lead onward. In the context of the present paper, I want to note that the DOUTH2 and other such relations among *pcsets* can be studied and generalized by using the *function* model I earlier used to study and generalize Cohn's initial ideas about pcsets.

We can specifically extend the notion of a DOUTH2 relation between pcsets, to a notion of a DOUTH2 relation between functions mod N, as follows.

DEFINITION 10.1: Let f and f′ be functions mod N. We shall say that f and f′ are "DOUTH2-related" when we can find numbers m and n mod N such that criteria (a) through (c) below all obtain.

(a) The four numbers m, m+1, n, and n+1 are all distinct mod N.
(b) f(m) ne f(m+1); f(n) ne f(n+1).
(c) f′(m) = f(m+1); f′(m+1) = f(m)
f′(n) = f(n+1); f′(n+1) = f(n)
f′(j) = f(j) for all other arguments j mod N.

We will then be particularly interested in functions f that are DOUTH2-related to various of their rotated ("transposed") and/or rotated-retrograde ("inverted") forms. Such functions can be investigated using techniques like those used to explore the necessary structure of Cohn functions, in the body of this paper.

The "extended DOUTH2 relation" can be defined among functions as follows.

DEFINITION 10,2: Let f be a function mod N. Suppose that the following algorithm can be executed.

(a) Find m (if possible) so that f(m) ne f(m+1).

(b) Define $f'(m) = f(m+1)$; $f'(m+1) = f(m)$;
$f'(j) = f(j)$ for all other j mod N.

(c) Select n (possibly = m) such that
$f'(n)$ ne $f'(n+1)$.

(d) Define $f''(n) = f'(n+1)$; $f''(n+1) = f(n)$;
$f''(j) = f'(j)$ for all other j mod N.

In that case, we shall say that the functions f and f″ are"EXTDOUTH2-related."

As was the case with the corresponding relations in Part 9 above, we can show that any pair of functions that is DOUTH2-related is also EXTDOUTH2-related. (So EXTDOUTH2 does indeed "extend" DOUTH2.) And we can show that any function f is necessarily EXT DOUTH2-related to itself, so long as f has more than one value.

We will be particularly interested in functions f that are EXT DOUTH2-related to various of their rotated ("transposed") and/or rotated-retrograde ("inverted") forms. Such functions can be investigated using techniques like those used to explore the necessary structure of Cohn functions, in the body of this paper.

And so forth. Though Cohn's original ideas, and many of their extensions by others, are formulated so as to concern pcsets mod N, much of the work can be generalized so as to pertain to functions mod N. Instead of talking about transposed or inverted pcsets, we then talk about rotated or rotated-retrograde functions. The conceptual focus changes: instead of regarding the numbers mod N as *objects*, either set-theoretic or embedded in a mod N "scale," we regard the numbers mod N as *order-positions* in a *serial* context where the "objects" arranged in series can be anything whatsoever—they are now simply the values of functions. So far as the formal work is concerned, we need only be able to distinguish situations like "$f(j) = f(j+1)$," from situations like "f(j) ne f(j+1)." For the formal work, we need not concern ourselves any more with what f(j) and f(j+1) "are." This leaves the functions free to model a great many musical phenomena, including pitch-class phenomena but also including many others.

APPENDIX A

We provide here a formal proof for Theorem 4.4, which is restated below.

THEOREM 4.4: Conditions (a) through (d) below are all logically equivalent. Each condition implies all the others.

(a) G is a generator mod N.

(b) The order of G, mod N, is N.

(c) As integers, G and N have no common divisor bigger than 1.

(d) G is a unit mod N.

We divide the proof into four sections. First we shall prove that condition (a) implies condition (b). Then we shall prove that (b) implies (c). Then we shall prove that (d) implies (a). And finally we shall prove that (c) implies (d). These steps entail the complete logical equivalence of the four conditions.

Proof that (a) implies (b): Let K be the order of G mod N. K, as noted earlier, is either equal to N or smaller than N. The list of G-multiples mod N, after commencing with G,2G,3G, . . . ,KG=0, cycles around and repeats with (K+1)G = KG+G = G, (K+2)G = KG+2G = 2G, and so forth. So the set of G-multiples mod N comprises at most K distinct numbers mod N. By the assumption of condition (a), K must equal N.

Proof that (b) implies (c): Let d be a positive whole number that divides both G and N. We write N = N′d; G =dG′. Then N′G = N′dG′ = NG′ = 0 mod N. Since N′G = 0 mod N, N′ is greater than or equal to the order of G. But by the assumption of (b), the order of G is N. So N′, which divides N, is also greater than or equal to N. It follows that N′ must equal N, whence d must equal 1.

Proof that (d) implies (a): Since G is a unit, there is some number v such that vG=1 mod N. Given any number j mod N, the number jv is such that (jv)G = j(vG) = j mod N.Thus any number mod N is some multiple of G, as required.

The proof that (c) implies (d) is the most complex. To prove it, we shall first prove the following lemma.

LEMMA: Let S be any additive subgroup of the integers (positive, negative, and zero). Assume S does not consist of zero alone. Let d be the smallest positive number contained in S. Then S comprises exactly the various multiples of d.

Proof of the lemma: Since S contains d, S contains d, d+d, d+d+d, and so forth. S, bring a subgroup, also contains the negatives of all those numbers. So S contains every multiple of d. We must show now that every member of S is a multiple of d. S being a subgroup, it suffices to show this for every positive member of S. Let s be a positive member of S. Let jd be the largest multiple of d that is less than or equal to s. Then s = jd + r, where r is some number lying between 0 and d-1 inclusive. Now s is a member of S, and jd—being a multiple of d—is a member of S. S being a subgroup, the number s-jd is a member of S. But that number is r. Thus r is a member of S, and r lies between 0 and d-1 inclusive. Since d is the smallest positive member of S, then r cannot be positive. Hence r must equal 0. And so s = jd. As desired, the sample member s of S is indeed some multiple of d.

We can now proceed to the proof that (c) of 4.1 implies (d) of 4.1. Given that G and N have no common divisor bigger than 1, consider the family S of integers comprising those integers that can be written in the form jG+kN, where j and k range through all integers. The sum or difference of any two such integers is such an integer, so S is an additive subgroup of integers. S contains both G (which = 1G+0N) and N (which = 0G+1N).

Let d be the smallest positive member of S. Then, according to the lemma, S consists precisely of all the multiples of d. So G and N, both members of S, are both multiples of d. By the assumption of condition (c), d must equal 1. So 1 is a member of S.

That means—according to the way S was defined—that the integer 1 can be written as jG+kN, for some integers j and k. So 1 differs from jG by a multiple of N. Hence jG = 1 mod N. And so G is a unit mod N, as asserted.

NOTES

1. Attending the symposium were Norman Carey, David Clampitt, John Clough—who organized the event, Richard Cohn, Jack Douthett, Daniel Harrison, Martha Hyde, Carol Krumhansl, David Lewin, and Charles Smith. Some of Cohn's initial ideas have been synopsized in his article on "Maximally Smooth Cycles, Hexatonic Systems, and the Analysis of Late-Romantic Triadic Progressions," *music analysis* 15, (March, 1996), 9–40.
2. I am grateful to Cohn for showing me (in an unpublished email letter) how the symbols a, u, and r can be used in this way, to represent certain rhythmic ideas by 3-valued Cohn functions.
3. Moritz Hauptmann, *Die Natur der Harmonik und der Metrik* (Leipzig: Breitkopf & Härtel, 1853). Trans. William Edward Heathcote, *The Nature of Harmony and Metre*, reprinted (New York: Da Capo Press, 1989).
4. The analogous graph for [the interval-series of] harmonic triads would illustrate, in its three different directional bondings, Riemann's relations of our relative major/minor, of our parallel major/minor, and of his Leittonwechsel.
5. It is no accident here that 5, which is one of the "arguments 5 and 6," is the generator. More specifically, the generator 5 is the *order-interval* between arguments 0-and-1, and arguments 5-and-6, the two adjacent-argument-pairs at which f Cohn-flips. We shall discuss this later on in generality.
6. This seems a bit perverse, but the assignment of y and x values will correlate better with the structure of earlier examples for generator-type functions.
7. f0 is a (two-valued generator-type) Cohn function. It Cohn-flips at arguments 1-and-2, and also at arguments (K-1)-and-0. This is not of much special interest in the present context, for f0 is much more radically constrained by its specific construction here: It must be the *specific* function displayed in the example.
8. Those who want to essay their own proofs for the theorems are advised to start by proving the following lemma: Let f be any function mod N (not necessarily a Cohn function) which Cohn-flips at arguments Q-and-(1+Q). (It does not matter whether Q is a generator or not.) Let f′ be the function defined by f′(Q)=f(1+Q); f′(1+Q)=f(Q); f′(j)=f(j) for all other values of J mod N. Then f′(j) = f(1+2Q-j) for all arguments j mod N. That is, the index of rotation for the pertinent rotated-retrogression of f, once the values f(Q) and f(1+Q) have been exchanged, is 1+2Q, the sum of Q and 1+Q.
9. The reader is referred to the article on "Pitch-Class Poker" by Lora Gingerich in the *Journal of Music Theory Pedagogy* 5.2 (Fall 1991), pp. 161–78, for a prescient approach to these sorts of ideas in a pedagogical setting.
10. The structural involvement of the ic3-cycle in this context vividly evokes aspects of Benjamin Boretz's *Meta-Variations: Studies in the Foundations of Musical Thought*, Ph.D. dissertation, Princeton University, 1970. A revised version of the dissertation appears in *Perspectives of New Music*, volumes 8–11 (Fall-Winter 1969 through Spring-Summer 1973). Particularly to the point are the analytic remarks on *Tristan* in volume 11, number 1 (Fall-Winter 1972), 159–217.
11. The connection with Boretz's work is again striking.
12. This sort of tension bewteen "tonal" and "atonal" meanings for the semidiminished and dominant seventh chords seems emblematic for Wagnerian dramaturgy from 1858 on. There are two worlds in *Tristan*. One is the world of social and

political obligation, responsibility, and action. In this world Tristan is a knight, a subject of King Mark's, a trusted vassal and a relative on an important political mission. Isolde is a Queen affianced to a foreign King in a political arrangement. The drama here is a drama of *action*, much in the spirit of Scribe (though Wagner no doubt preferred to think of his model as Schiller). There are purposes to be fulfilled, ends to be attained, and people must *act* to attain those ends. In this world, harmony is tonally functional in a more or less traditional sense: dissonances must be explained; they must be resolved to one key (end) or another, and they push on to do so, achieving teleological purposes. The Tristan Chord {F,G♯,B,D♯}, in particular, which disrupts the tonally functional teleology, must resolve in one key or another. (It tries to resolve in almost every key, I should say, somewhere or other in the drama.) The other world of *Tristan* is the world of passion, of oceanic feeling, of the *Liebestod*, the love relationship between Tristan and Isolde. This world knows no political or social obligations or purposes; it seeks only to generate itself indefinitely in all directions. Harmony, in this world, works "atonally," generating intervallic patterns and relationships that exfoliate ceaselessly. In this harmonic world, the would-be "keys" into which the Tristan Chord "resolves" are naught but neighboring or passing inflections of the Chord itself (or whatever of its T/I forms may be locally prominent). So the behavior of the dominant seventh chord at Stage 6 of example 20, shattering the transformational symmetries of Stage 5 as it pushes on to a functional tonal cadence, is allegorically emblematic in this connection. So, likewise, is the behavior of the dominant seventh chord at Stage 7 of the Immolation Scene exercise, shattering the transformational symmetries of Stage 6 as it pushes on to a functional tonal cadence. Here, Brünnhilde's thoughts, having explored her love relation with Siegfried, turn toward action and purpose.

13. Gingerich's "Pitch-Class Poker" seems especially prescient in the context of the algorithm.
14. Take x of the algorithm = x1 as per the DOUTHCF2 criterion; take y1 according to the same criterion; take z of the algorithm = x2 as per the DOUTHCF2 criterion, and select y2 as per the DOUTHCF2 criterion.
15. Find x for stage (i) of the algorithm so that some y1, fifth-related to x, is not a member of X. Take that y1 for stage (ii). Then, at stage (iv), discard z=y1. And then, at stage (v), pick up x from the pack as "y2." After executing the algorithm in that fashion, one ends up with pcset Y the same as the original pcset X.