A Binomial Representation of Pitch for Computer Processing of Musical Data
Author(s): Alexander R. Brinkman
Source: *Music Theory Spectrum*, Vol. 8 (Spring, 1986), pp. 44-57
Published by: University of California Press on behalf of the Society for Music Theory
Stable URL: http://www.jstor.org/stable/746069
Accessed: 15/02/2010 10:23

A Binomial Representation of Pitch for Computer Processing of Musical Data

Alexander R. Brinkman

A number of coding languages have been devised to represent musical information for computer processing, but whatever convention is used to code the music, operations on the data are usually done arithmetically within the computer. Therefore, note names or positions on the musical staff are usually translated into numerical values before processing. The use of integers, e.g. 0–11, to represent pitch classes (pcs) is often adequate, particularly for atonal or serial music. However, in many instances accuracy requires that a distinction be made between enharmonically equivalent notes and intervals. This is true, for example, in applications dealing with tonal music, music printing, computer assisted instruction, and even in synthesis of music in tuning systems other than equal temperament.

This article describes a representation for pitch structures that can be manipulated arithmetically, but which distinguishes between enharmonic notes and intervals while retaining octave and pc information. The system is based on extensions to the pc concept as used extensively by scholars of twentieth century music.[1]

The basic technique described here is not new. I have been using it and teaching it to my students for a number of years, and I am certain that others have used it as well. However, there has been no previous formalization of the system.[2]

The pitch-class system. In the system of pcs, class membership is defined by *octave equivalence* and *enharmonic equivalence*. Thus a pc includes all possible alternate spellings and registral dispositions of a given pitch. For the present purpose it will be useful to adopt a fixed correspondence between integer notation and staff notation with C = 0, C♯/D♭ = 1, D = 2, . . ., B = 11.

Integer notation avoids the distinctions of specific spelling and register that are inherent in notated music. In addition, it simplifies musical operations such as interval calculation, transposition, and inversion by reducing them to arithmetic operations.

Though the degree of generality embodied in the pc system is ideally suited to certain problems in twentieth-century music,

An earlier version of this paper was presented in a "Symposium on Future Computer Applications in Music Theory" at the annual conference of the Society for Music Theory held at the University of Southern California, Los Angeles, October 1981. I wish to thank Robert D. Morris for many helpful suggestions while preparing the original paper.

[1]The terms *pitch class* and *pitch-class set* were introduced by Milton Babbitt in the early 1960s. The pc system and its applications have been formalized in Allen Forte, *The Structure of Atonal Music* (New Haven: Yale University Press, 1973), and in John Rahn, *Basic Atonal Theory* (New York: Longman, 1980), as well in many articles in the literature.

[2]A similar notation is mentioned in Terry Winograd, "Linguistics and Computer Analysis of Tonal Harmony," *Journal of Music Theory* 12 (1968): 2–49. The coding scheme is mentioned briefly on p. 32.

it may be a hindrance when studying other types of music. Nevertheless, the ease with which this system can be manipulated arithmetically is useful, especially in computer-assisted studies. Many of the desirable qualities of the pc system can be adapted to problems requiring more specific information on the notation used by the composer.

The name-class system. Using the pc system as a model, we can devise a system that more accurately represents musical notation. The letter names which correspond to staff positions in musical notation are cyclic just as pitch classes are, repeating after seven notes. Each letter name can be considered a *name class* (nc), which represents all instances of a given note name regardless of register or chromatic inflection. Thus, in the nc system class membership is defined by *octave equivalence* and *diatonic equivalence.* In this system D, D♯, and D♭ are equivalent, but C♯ and D♭ are not, since they are spelled using different letter names and occupy different staff positions in the same octave.

Just as the twelve pcs can be represented by the integers 0 through 11 within a modulo 12 system, the elements of the letter-name scale can be represented by the integers 0 through 6 within a modulo 7 system. Using a fixed correspondence between integers and note names, 0 can represent any C, 1 any D, 4 any G, and so forth.[3]

Intervals in this system are analogous to intervals in the pc system; i.e., they can be calculated as the mod 7 difference between nc integers. Nc intervals are generic rather than specific. All thirds, whether major, minor, augmented, or diminished, are represented by a difference of 2; all fourths by a difference of 3. Interval inversion is calculated as the mod 7 complement (or inverse) of an interval. Table 1 shows intervals and their integer notation in the nc system. Because of inversional equivalence there are only four interval classes; these are represented by the integers 0–3.

Table 1. Complementary Intervals in the name-class system

name	nc	nc	name
prime	0	7/0	octave
second	1	6	seventh
third	2	5	sixth
fourth	3	4	fifth

Each operation or procedure that is possible in the pc system has an analogue in the nc system. We can create nc sets, transpose them, calculate inversions, represent their generic interval content by interval vectors, and reduce them to prime forms.[4]

Table 2 shows all possible prime forms in the name-class system. This table recognizes transpositional and inversional equivalence. Note that the nc interval-class vector uses only three digits indicating, from left to right, the number of seconds and sevenths (ic 1), the number of thirds and sixths (ic 2), and the number of fourths and fifths (ic 3). The table also lists a possible traditional interpretation for each set. The nc in boldface is the most likely candidate for the traditional "root" of the set.

The binomial representation. The pc and nc systems are complementary, with each making up for deficiencies of the other. The pc designates the exact pitch, while the nc indicates the spelling. The advantages of both systems can be realized concurrently if notes are represented by integer couples in the form <pc, nc> with a common point of origin <0, 0> representing C. Enharmonic spellings of the same note have the

[3]As a mnemonic, name-class integers are easy to remember if related to scale steps in the scale on D. Name-class intervals are always one less than the standard name: prime = 0, second = 1, third = 2, etc.

[4]A more comprehensive discussion of diatonic sets may be found in John Clough, "Aspects of Diatonic Sets," *Journal of Music Theory* 23 (1979): 45–61.

Table 2. Prime forms in the name-class system

nc set	ic vector	possible interpretation	nc set	ic vector	possible interpretation
null	000	null set	**0**,1,2,3,4,5,6	777	scale
0	000	note	**0**,1,2,3,4,5	555	scale segment
0,1	100	second/seventh	**0**,1,2,3,4	433	scale segment
0,2	010	third/sixth	0,**1**,2,3,5	343	9th chord
0,**3**	001	fourth/fifth	**0**,1,2,4,5	334	pentatonic
0,1,2	210	scale segment	**0**,1,2,3	321	scale segment
0,**1**,3	111	inc. 7th chord	**0**,1,2,4	222	inc. 9th chord
0,1,4	102	quartal/quintal	0,1,**3**,4	213	quartal/quintal
0,2,4	021	triad	0,**1**,3,5	132	7th chord

same pc but different name classes, e.g. C♯ = <1, 0> and D♭ = <1, 1>; chromatic alterations of the same note share the same nc but have a different pc component, e.g. C = <0, 0> and C♭ = <11, 0>.

There are 84 (= 7 • 12) binomial pcs in the system. Of these 35 are in common use, representing notes up to and including double accidentals. Table 3 shows the binomial representation for these 35 notes. Enharmonic spellings (those with the same pitch class) occur in the same row, while notes spelled with the same letter name occur in the same column. The system is extensible within reasonable limits. Triple, quadruple, even quintuple accidentals are theoretically possible, though unlikely in practice. Ambiguities result for more than five accidentals, although they could be resolved contextually. (Is <6, 0> C-sextuple-sharp or C-sextuple-flat?)

The binomial system can also represent intervals. The first integer specifies the interval size in half steps and the second integer designates the generic interval size. Thus, a minor third

Table 3. Binomial representation of notes

Pitch Class	Name Class						
	0	1	2	3	4	5	6
0	C	D♭♭					B♯
1	C♯	D♭					Bx
2	C×	D	E♭♭				
3		D♯	E♭	F♭♭			
4		D×	E	F♭			
5			E♯	F	G♭♭		
6			E×	F♯	G♭		
7				F×	G	A♭♭	
8					G♯	A♭	
9					G×	A	B♭♭
10	C♭♭					A♯	B♭
11	C♭					A×	B

is represented by the couple $<3, 2>$, major third by $<4, 2>$, augmented third by $<5,2>$, and diminished third by $<2, 2>$. The system is unambiguous through quintuply augmented or diminished intervals, which should be sufficient for most practical applications. Table 4 shows the binomial representation of intervals through augmented and diminished intervals. Intervals of the same generic size occur in the same column; enharmonically equivalent intervals in the same row.

Musical operations in the binomial system. All operations that are possible on pcs and ncs are possible in the binomial system. The operations are performed separately on the ncs in mod 7 and on the pcs in mod 12, as follows:

1. Transposition is addition. To transpose a note $<a, b>$ by interval $<c, d>$ do the following:

$$<a, b> + <c, d> = <(a + c) \bmod 12, (b + d) \bmod 7>$$

Table 4. Binomial representation of intervals

Pitch Class	Name Class						
	0	1	2	3	4	5	6
0	P1	D2					A7
1	A1	m2					
2		M2	d3				
3		A2	m3				
4			M3	d4			
5			A3	P4			
6				A4	d5		
7					P5	d6	
8					A5	m6	
9						M6	d7
10						A6	m7
11	d1						M7

Example: transpose D up a major 3rd.

$$<2, 1> + <4, 2> = <6, 3>$$
$$(\quad D \quad + \quad M3 \quad = \quad F\sharp \quad)$$

Note: the transposition interval may be negative; e.g., to transpose down by a M3 add $<-4, -2>$. This is equivalent to adding the complement (see 3 below).

2. Interval calculation is subtraction. To find the interval between two notes subtract the lower note from the higher note.
$$<a, b> - <c, d> = <(a-c) \bmod 12, (b-d) \bmod 7>$$

Example: the interval between E♭ and A.

$$<9, 5> - <3, 2> = <6,3>$$
$$(\quad A \quad - \quad E\flat \quad = \quad A4 \quad)$$

or

$$<3, 2> - <9, 5> = <6, 4>$$
$$(\quad E\flat \quad - \quad A \quad = \quad d5 \quad)$$

3. Interval inversion is complementation of the pc mod 12 and of the nc mod 7. The inversion of the interval $<a, b>$ is $<12, 7> - <a, b> = <(12-a) \bmod 12, (7-b) \bmod 7>$.

Example: the inversion of P4.

$$<12, 7> - <5, 3> = <7, 4>$$
$$(\qquad P4 \qquad P5 \quad)$$

4. Melodic inversion is subtraction of each note from a constant. To begin the inversion on the same note as the prime choose the constant two times the first note.

Example: Melodic inversion of D E F♯ A C.

(D	E	F♯	A	C)
$<2, 1>$	$<4, 2>$	$<6, 3>$	$<9, 5>$	$<0, 0>$

constant $= 2 \cdot <2, 1> = <4, 2>$

4,2	4,2	4,2	4,2	4,2
−2,1	−4,2	−6,3	−9,5	−0,0
<2,1>	<0,0>	<10,6>	<7,4>	<4,2>
(D	C	B♭	G	E)

Note: inversion can also be obtained by multiplying each binomial couple by <11,6> and transposing the result as desired.

5. Diatonic operations (within a specific scale) are obtained by doing the normal operation on the name class (mod 7) and referencing a table for the correct pitch class. The table (s) is an array indexed by the name classes, with the corresponding pc as the content of each element.

Example: diatonic sequence in G major.

s

0	1	2	3	4	5	6	ncs
0	2	4	6	7	9	11	pcs for G major.

Sequence G A B G F♯ a second higher.

	<7,4>	<9,5>	<11,6>	<7,4>	<6,3>
	+*,1	+*,1	+*,1	+*,1	+*,1
=	<s[5],5>	<s[6],6>	<s[0],0>	<s[5],5>	<s[4],4>
=	<9,5>	<11,6>	<0,0>	<9,5>	<7,4>
	(A)	(B)	(C)	(A)	(G)

Properties of the binomial system. The universal set U consists of 84 binomial pitch classes such that,

binomial pc = <pc,nc>

where $pc \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$
$nc \in \{0, 1, 2, 3, 4, 5, 6\}$

We define binary operations as follows:

+ : <a,b>+<c,d> = <(a+b) mod 12, (b+d) mod 7>
• : <a,b>•<c,d> = <(a • c) mod 12, (b • d) mod 7>

The following properties obtain:

1. For any binomial pcs A and B

$$(A + B) \in U \quad \text{and} \quad (A \bullet B) \in U.$$

2. Both addition and multiplication are commutative on the binomial system since for any two binomial pcs A and B,

$$A + B = B + A \quad \text{and} \quad A \bullet B = B \bullet A.$$

3. Both addition and multiplication are associative on the binomial system since for any three binomial pcs A, B, and C,

$$(A + B) + C = A + (B + C)$$
and
$$(A \bullet B) \bullet C = A \bullet (B \bullet C).$$

4. Multiplication is distributive over addition on the binomial system since for any three binomial pcs A, B, and C,

$$A \bullet (B + C) = A \bullet B + A \bullet C$$
and
$$(A + B) \bullet C = A \bullet C + B \bullet C.$$

5. There exists an identity element for addition which equals <0,0>, such that for any binomial pc A,

$$A + \langle 0, 0\rangle = A.$$

6. There exists an inverse A′ for every binomial pc A such that

$$A' + A = \langle 0, 0\rangle.$$

The inverse of <a,b> = <(12−a) mod 12, (7−b) mod 7>.

7. There exists an identity element for multiplication which equals <1, 1> such that for any binomial pc A,

$$A \bullet \langle 1, 1\rangle = A.$$

8. We define scalar multiplication such that for any positive integer n and any binomial pc $<a, b>$,

$$n \cdot <a, b> = <(n \cdot a) \bmod 12, (n \cdot b) \bmod 7>.$$

Properties one through seven are sufficient to show that the binomial system is a *commutative ring with unity*.

Binomial structures. Since each tone in any collection of notes can be expressed as an interval above the first note, the binomial system can also represent larger pitch structures. For example, the structure of a major triad can be expressed as a major third and a perfect fifth above the root. In the binomial system, the structure of any root position major triad can be represented by the couples $<0, 0>$ $<4, 2>$ $<7, 4>$. Each couple represents an interval above the point of origin, or root. In the same manner, the structure of any major scale can be represented as a series of intervals above the tonic.

Since all instances of a given structure are transpositionally equivalent, the correct spelling for any transposition can be calculated by adding the couple representing the desired root or tonic note to each element of the structure. Figure 1 illustrates the calculation of the notes in the G♭ major scale from the major-scale structure.

Figure 1. The major scale structure

$<0,0>$ $<2,1>$ $<4,2>$ $<5,3>$ $<7,4>$ $<9,5>$ $<11,6>$
G♭ Major scale is calculated by adding G♭ $<6,4>$ to each structure element:

$<0,0>$	$<2,1>$	$<4,2>$	$<5,3>$	$<7,4>$	$<9,5>$	$<11,6>$
+6,4	+6,4	+6,4	+6,4	+6,4	+6,4	+6,4
$<6,4>$	$<8,5>$	$<10,6>$	$<11,0>$	$<1,1>$	$<3,2>$	$<5,3>$
(G♭	A♭	B♭	C♭	D♭	E♭	F)

Binomial structures can be used advantageously in a number of programming situations. In an instructional program to drill students in spelling chords or scales, it is not necessary to store in memory every possible scale or chord—only the structure of one representative of each desired type. An analysis program might also use structures to quickly identify chordal types. (A "normal form" for the binomial system will be discussed later.) Table 5 shows the structure of some common entities in traditional tertian music.

A single-number representation of the binomial system. There are many possible methods of packing the nc and pc into a single integer. One simple method is shown below. The single-number representation (br) of a binomial couple $<pc, nc>$ can be obtained as follows:

$$br = pc \cdot 10 + nc$$

where pc is the pitch class and
nc is the name class.

The number can be decoded just as simply:

nc = br mod 10
pc = br div 10 [integer division, i.e. truncate]

This coding system has several desirable qualities. First, it is easy to read so it need not be decoded before the analyst can understand it: the last digit always indicates the spelling (nc) and the preceding digits give the pc:

br	pc	nc	note-name
116	11	6	B♮
31	3	1	D♯
32	3	2	E♭
1	0	1	D♭♭
10	1	0	C♯

Table 5. Binomial structure of some common pitch sets

Name	Structure
Triads	
Major	<0,0> <4,2> <7,4>
Minor	<0,0> <3,2> <7,4>
Diminished	<0,0> <3,2> <6,4>
Augmented	<0,0> <4,2> <8,4>
Seventh Chords	
Major	<0,0> <4,2> <7,4> <11,6>
Dominant	<0,0> <4,2> <7,4> <10,6>
Minor	<0,0> <3,2> <7,4> <10,6>
Half diminished	<0,0> <3,2> <6,4> <10,6>
Diminished	<0,0> <3,2> <6,4> < 9,6>
Scales	
Major	<0,0> <2,1> <4,2> <5,3> <7,4> <9,5> <11,6>
Natural Minor	<0,0> <2,1> <3,2> <5,3> <7,4> <8,5> <10,6>
Harmonic Minor	<0,0> <2,1> <3,2> <5,3> <7,4> <8,5> <11,6>
Mel. Min. Ascend.	<0,0> <2,1> <3,2> <5,3> <7,4> <9,5> <11,6>

A second advantage of the single-number br is that it is compact. The largest number required is 116. Even with the octave number (discussed below), 9116 or 10116 would be more than sufficient for the top range. Thus, the coded pc, nc, and octave number easily fit into a single 16-bit computer word. A final advantage is that the single-number representation greatly simplifies the problem of defining normal order, prime forms, interval vectors, and so on.

Mathematical operations on single-number brs. Mathematical operations on br-coded intervals and notes (brs) must take into account the decoding and recoding of the brs. We define the following operations on any two brs A and B:[5]

[5]The formulas given here assume that the mod function operates properly on *negative* integers, i.e., $(-3 \bmod 12) = 9$. This is frequently *not* the case since many computer installations define the mod function thus: a mod b : = a − ((a div b)•a); however the problem can be corrected by a simple Pascal function.

1. Addition (transposition):

$$A + B = 10 \cdot ((A \text{ div } 10 + B \text{ div } 10) \bmod 12) \quad [\text{pc part}]$$
$$+ ((A \bmod 10 + B \bmod 10) \bmod 7) \quad [\text{nc part}]$$

2. Subtraction (interval calculation or downward transposition):

$$A - B = 10 \cdot ((A \text{ div } 10 - B \text{ div } 10) \bmod 12) \quad [\text{pc part}]$$
$$+ ((A \bmod 10 - B \bmod 10) \bmod 7) \quad [\text{nc part}]$$

3. Complementation (interval inversion):

$$A' = 10 \cdot ((12 - (A \text{ div } 10)) \bmod 10) \quad [\text{pc part}]$$
$$+ ((7 - (A \bmod 10)) \bmod 7) \quad [\text{nc part}]$$

4. Multiplication (inversion through multiplication by 116):

$$A \cdot B = 10 (((A \text{ div } 10) \cdot (B \text{ div } 10)) \bmod 12) \quad [\text{pc part}]$$
$$+ (((A \bmod 10) \cdot (B \bmod 10)) \bmod 7) \quad [\text{nc part}]$$

5. Multiplication of a br-coded interval A by some positive integer n (interval projection):

$$n \cdot A = 10 \cdot (n \cdot (A \text{ div } 10) \bmod 12) \quad [\text{pc part}]$$
$$+ (n \cdot (A \bmod 10) \bmod 7) \quad [\text{nc part}]$$

Interval classes. A binomial interval class (ic) consists of the complementary pair of br-coded intervals (brs) such that $A = B'$ and $B = A'$ (as defined above). The ic is represented by the br with the smaller numeric value. For example, the ic consisting of the P4 (53) and the P5 (74) is represented by 53. The general rule is that if the br is greater than 63, the interval class is represented by its complement.

There are 84 intervals and 42 ics in the binomial system. Of these, 25 intervals (perfect, major, minor, diminished, and augmented intervals) and 12 ics are commonly used in music.

Table 6 shows all of the complementary interval pairs in the binomial system. The column titled "name" gives the traditional name for the interval (p1 = perfect prime; A4 = augmented fourth; m7 = minor seventh; etc). Where the modifier A or d is preceded by an integer, this further qualifies the interval (2d3 = doubly diminished third; 4A5 = quadruply augmented fifth). The column titled "index" will be explained presently.

Binomial pc-sets. Binomial pc-sets are sets of distinct br-coded notes. "Distinct" signifies that no two brs may be identical. Of course it is possible to have two or more enharmonically equivalent notes (C♯, D♭, B𝄪). It is also permissible for a set of binomial pcs to contain two or more nc-equivalent notes (C♯, C♭, C𝄪). Since there are 84 pcs in the binomial system, binomial pc-sets will contain from 0 (null set) to 84 binomial pcs (brs).

Interval-class vectors. It is possible to construct ic vectors in the binomial system, although the process is complicated by the fact that the coded binomial pcs do not form a contiguous set of integers, and by the fact that ics representing augmented primes have as their complements diminished octaves (or primes, by octave equivalence) which have the same nc component. Because of these factors it is necessary to define an index for each ic. The index is the ordinal position of each binomial ic in the ordered set of ics. The index for each ic in the system is shown in Table 6. The index for any binomial pc can be calculated by the pair of functions in the Pascal language, shown below. The first function defines the complement relation, and the second calculates the index:

```
function complement (br : integer) : integer;
 begin
  complement : = 10 * ((12 - (br div 10)) mod 12)
                    + ((7 - (br mod 10)) mod 7)
 end;

function index (br : integer) : integer;
 var
  t : integer        (* temporary value of index *)
```

```
begin
 if (br > = 04) and (br < = 06)
  then br : = br + 120;          (* force complement *)
 if br > 63
  then br : = complement (br);
 t : = 7 * (br div 10) + br mod 10;
 if br > = 10
  then index : = t - 3
  else index : = t
end;
```

Table 6. Complementary intervals in the binomial system

index	name	br	br	name
0	P1	00	127/00	P8
1	d2	01	126/06	A7
2	3d3	02	125/05	3A6
3	5d4	03	124/04	5A5
4	A1	10	117/110	d8/d1
5	m2	11	116	M7
6	2d3	12	115	2A6
7	4d4	13	114	4A5
8	6d5	14	113	6A4/6d4
9	4A6	15	112	4d3
10	2A7	16	111	2d2
11	2A1	20	107/100	2d8/2d1
12	M2	21	106	m7
13	d3	22	105	A6
14	3d4	23	104	3A5
15	5d5	24	103	5A4
16	5A6	25	102	5d3
17	3A7	26	101	3d2
18	3A1	30	97/90	3d8/3d1
19	A2	31	96	d7
20	m3	32	95	M6
21	2d4	33	94	2A5
22	4d5	34	93	4A4
23	5d6	35	92	5A3
24	4A7	36	91	4d2
25	4A1	40	87/80	4d8/4d1
26	2A2	41	86	2d7
27	M3	42	85	m6
28	d4	43	84	A5
29	3d5	44	83	3A4
30	4d6	45	82	4A3
31	5A7	46	81	5d2
32	5A1	50	77/70	5d8/5d1
33	3A2	51	76	3d7
34	A3	52	75	d6
35	P4	53	74	P5
36	2d5	54	73	2A4
37	3d6	55	72	3A3
38	5d7	56	71	5A2
39	6A1	60	67/60	6d8/6d1
40	4A2	61	66	4d7
41	2A3	62	65	2d6
42	A4	63	64	d5

Figure 2 shows a possible format for a binomial interval vector. The primary division of ics is according to the pc component of the coded note. The subdivisions give the spelling of each interval. The index is the position (1–42) in the vector.

Since duplicate brs are not permitted in binomial pc-sets, the interval class 00 has been omitted.

Figure 2. An interval vector for the binomial system

pc:	000	1111111	2222222	3333333	4444444	5555555	6666
nc:	123	0123456	0123456	0123456	0123456	0123456	0123
vector:	000	0000000	0000000	0000000	0000000	0000000	0000
index:	1 4	11	18	25	32	39	

The 42-element vector is somewhat cumbersome to construct and to read. A reasonable alternative for many applications is to use two vectors—the traditional six-element pc-vector and a three-element nc-vector. Examples follow:

pitches	br-set	pc/nc vectors
(c, e♭, g)	00, 32, 74	001110-021
(c, e, g)	00, 42, 74	001110-021
(c, e, g♯)	00, 42, 84	000300-021

This combination vector is sufficient for many purposes, and is much easier to read. It also shows certain types of relationships more directly: for example an inequality in the number of interval classes in the pitch and name domain indicates the existence of pc-equivalent or nc-equivalent note classes in the set. The double-vector system also makes it easy to group chords hierarchically in a kind of genus/species arrangement according to the pc/nc or nc/pc interval content. Thus all tertian triads could be considered to be of the same genus with the pc interval content determining distinctions within this group or, conversely, chords could be grouped according to their pc-interval content and differentiated according to their spelling.

The 42-element vector would be able to distinguish between the interval content of enharmonically similar structures (such as a dominant seventh chord and a German augmented sixth) while the two-vector system could not. However, this distinction can be made on the basis of normal forms.

A normal order for the binomial system. Using the single-number representation of binomial intervals, it is possible to define a normal order for a binomial pc-set. Since a primary use of such a system would be automated recognition of chord types, the normal order will be most useful if we recognize transpositional equivalence but exclude inversional equivalence. Except for this exclusion and the stipulation that all arithmetic operations be performed as defined above, the formulation of a normal order is almost identical to that defined for pc-sets by Forte.[6] The steps are as follows:

1. Sort the brs in ascending numerical order.
2. Find the circular permutation that gives the smallest interval between the first and last br. If two or more circular permutations have the same interval span, choose the one with the smallest intervals on the left.
3. Transpose the set to 00 by subtracting the first br from each br in the set.
4. Again sort the set in ascending numerical order.

Steps 1 and 2 give the normal order for the set at the specified pitch level. Steps 3 and 4 transpose the set to 00 to allow for easy comparison of transpositionally equivalent sets. Step 4 is necessary because at certain transposition levels pc-equivalent brs will become disordered with regard to numerical order. As an example, consider the pitch collection D, C♭♭, B♭, A♭:

[6]Forte, *The Structure of Atonal Music,* 3–5.

		(D	C♭♭	B♭	A♭)
binomial representation	:	21	100	106	85
1. Sort in ascending order	:	21	85	100	106
2. Find normal order	:	85	100	106	21
3. Transpose to 00	:	00	22	21	63
4. Sort in ascending order	:	00	21	22	63

Note that after step three pcs 22 and 21 were out of order. This can occur whenever the set contains pc-equivalent notes and the transposition level causes one of the nc-digits to cross the octave boundary (7 = 0).

A similar problem with reordering can occur with nc-equivalent notes. Consider the case of D♭ and D at t = 106 (m7):

(D♭	D)	
11	21	[in numerical order]
+106	+106	
110	00	[now disordered]
(C♭	C)	

Re-sorting after the transposition step will correct both of these problems.

It may be helpful to show the steps in finding normal order (from step 1 to step 2 above). The normal order is found by examining the adjacent interval pattern for the set. First the largest interval between successive brs is located (note that the interval calculation is done separately on the pcs in mod 12 and on the ncs in mod 7):

brs:	21	85	100	106	[21]
intervals:		64	22	06	42

Since the largest interval (64) occurs between 21 and 85, the best normal order (the circular permutation with the smallest span) begins with 85, i.e. 85, 100, 106, 21. If the largest interval had occurred in two or more positions, it would have been necessary to check successive intervals until the smallest interval after the large interval was found.

The normal-order/prime-form procedure shown above yields a unique prime form for each group of transpositionally equivalent binomial pc-sets. Because the spelling of the notes is taken into account, German augmented sixth chords can be differentiated from dominant seventh chords, and the root can be easily determined in pc-symmetrical structures like diminished seventh chords and augmented triads. Note that this system cannot distinguish between different modes of a scale (i.e., natural minor from major or Dorian). Since these are circular permutations of the same set, the tonic must be determined contextually.

The prime forms for some common musical structures are shown in Table 7. The interval in boldface is the traditional "root" of the sonority.

With 84 possible pcs in the binomial system, the number of set classes is formidable when compared with the number of possible pc-sets in the name or pitch system. It should be possible to construct a useful subset of the whole by using only those sets that can be generated by the thirty-four natural, sharped, flatted, double-flatted, and double-sharped notes. Certainly an analytical program could utilize a small table consisting of only those sets under investigation.

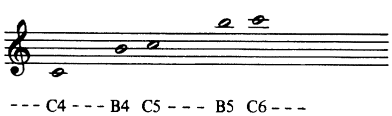
Adding octave information. The system described thus far is not capable of complete representation of pitch, since no provision has been made to distinguish between the various octaves. For our purpose, the most viable system of octave designation is that proposed by the Acoustical Society of America and accepted by the U.S.A. Standards Association. This system, shown in Figure 3, uses an integer subscript indicating the exact octave placement. The octave number refers to notes from the given C through the B notated a seventh above it.

Octave designations can be added to the binomial representation as a third term. C_4 is then designated $<4, 0, 0>$ and $C\sharp_5$ as $<5, 1, 0>$. In doing computations in this system, the octave

Table 7. Prime forms for some common sets in tonal music

Name	Prime Form
Triads	
It aug. 6th	00, 22, **64**
Diminished	**00,** 32, 64
Minor	**00,** 32, 74
Major	**00,** 42, 74
Augmented	**00,** 42, 84
Seventh chords	
Major	00, **11,** 53, 85
Half-dim.	00, **21,** 53, 85
Fr. aug. 6th	00, **21,** 63, 85
Diminished	00, **31,** 63, 95
Minor	00, 32, **53,** 85
Ger. aug. 6th	00, 32, **63,**85
Dominant	00, 32, 64, **85**
Scales	
Harm. min.	00, **11,** 32, 43, 64, 85, 96
Mel. min. asc.	00, **11,** 32, 43, 64, 85, 106
Major	00, **11,** 32, 53, 64, 85, 106
Whole-tone	**00,** 21, 42, 63, 84, 105
Pentatonic	**00,** 21, 42, 74, 95

Figure 3. Standard octave notation



number is mapped onto the pc and nc numbers, thus changing the cyclic pattern of integers into a continuous scale in both the pitch and name domains. A unique integer couple in the continuous scale is obtained as follows:

$$\langle cpc, cnc \rangle = \langle (oct \cdot 12) + pc, (oct \cdot 7) + nc \rangle$$

where pc is a pitch class
nc is a name class
oct is an octave designation

This modification makes it possible to represent any pitch in the standard notational system by a unique pair of integers. Examples are shown in Figure 4.

The pc, nc, and octave number can be calculated easily, as shown below:

$$\begin{aligned} pc &= cpc \bmod 12 \\ nc &= cnc \bmod 7 \\ oct &= cnc \text{ div } 7 \text{ [or } cpc \text{ div } 12] \end{aligned}$$

An example is shown in Figure 5.

Figure 4. Examples of pitch representation in the continuous scale

$$\begin{aligned} C\sharp_0 &= \langle (0 \cdot 12) + 1,\ (0 \cdot 7) + 0 \rangle &= \langle 1,0 \rangle \\ C\sharp_1 &= \langle (1 \cdot 12) + 1,\ (1 \cdot 7) + 0 \rangle &= \langle 13,7 \rangle \\ C\sharp_5 &= \langle (5 \cdot 12) + 1,\ (5 \cdot 7) + 0 \rangle &= \langle 61,35 \rangle \end{aligned}$$

Figure 5. Reduction of a note in the continuous scale

The binomial couple $\langle 63, 36 \rangle$ can be reduced as follows:

pitch class	= 63 mod 12 = 3
name class	= 36 mod 7 = 1
octave	= 36 div 7 or 63 div 12 = 5

therefore $\langle 63, 36 \rangle = D\sharp_5$.

While octave information could be carried by either the name or pitch number alone, mapping the octave onto both integers makes it possible to calculate the interval between two notes without first determining which note is higher. Since both pitch and name are represented on a continuous scale, the interval can be calculated either way as the absolute value of the difference between the respective pitch and name numbers. If the interval is greater than or equal to an octave, it can be converted to a simple interval by reducing the integers modulo 12 and modulo 7 respectively:

$$\begin{aligned} \text{interval} &= |<p1, n1> - <p2, n2>| \\ &= <|p1-p2| \bmod 12, |n1-n2| \bmod 7> \end{aligned}$$

Example: The interval between C_4 and G_4 is

$$\begin{aligned} &(\quad C_4 \qquad\qquad G_4 \quad) \\ &|<48, 28> - <55, 32>| \\ &= <|48-55|, |28-32|> \\ &= <7, 4> \quad (P5). \end{aligned}$$

The octave number can be a single-number continuous binary representation (cbr) as follows:

$$cbr = oct \cdot 1000 + br$$

where br is as defined above and
oct is an octave number (lowest octave = 0)

Decoding the above:

$$\begin{aligned} br &= cbr \bmod 1000 \\ oct &= cbr \operatorname{div} 1000 \end{aligned}$$

If we allow the cbr to be signed ($\pm$cbr), and provide operations to manipulate these numbers, we have a registral-space generalization of the binomial system. For pitches, the cbr designates octave, pc and nc (spelling). When the cbr represents an interval, the high-order digit allows for compound intervals, and the sign indicates direction. Signed cbrs are easily manipulated. For example, the function add (x, y) shown in Figure 6 adds any two cbr's in the form

$$\pm \text{opcn}$$

where $\pm$ is an optional sign
o is the standard octave number (0 or more digits)
pc is the pitch class (two digits: 00–11)
and n is the name class (one digit: 0–6).

The result is returned in the same form. Thus the function can be used to transpose or to calculate intervals between two pitches. In transposing, a positive or negative cbr-coded inter-

Figure 6. Function for addition of cbrs

```
Function adds two signed cbrs in the form (±) opcn
  where  o  is standard octave no. (0 or more digits)
         pc is pitch class (2 digits 00–11)
  and    n  is the name class (one digit 0–6)
  returns a signed cbr as function result

function add (cbr1, cbr2 : integer) : integer;
 var
  pitch, name, octave : integer;
  negative : boolean;
 begin
  pitch := abs (cpc (cbr1) + cpc (cbr2)) mod 12;
  name := cnc (cbr1) + cnc (cbr2);
  negative := name < 0;
  octave := abs (name) div 7;
  name := abs (name) mod 7;
  if negative
    then add := −((1000 * octave) + (10 * pitch) + name)
    else add :=   (1000 * octave) + (10 * pitch) + name
 end;
```

val B is added to a cbr-coded pitch A [add(A, B) or add (B, A)]. The function returns the new pitch in the correct octave. The directed interval between any two cbr-coded pitches X and Y is calculated by subtracting X from Y [add(Y, −X) or add (−X, Y)]. The result is a cbr-coded interval; the sign indicates direction (negative = down). Note that the octave part of the result is calculated in conjunction with the nc part; that is, as in standard notation the octave depends on spelling rather than chromatic inflection. Translations to and from standard pitch and interval designations are readily accomplished.

A set of procedures and functions in the Pascal language, for calculation of the normal order and prime form of binomial sets and for manipulation of signed cbrs, is available from the author upon request.