A Boolean theory of signatures for tonal scales

Bruno Simeone(*), Gilbert Nouno(^), Malik Mezzadri(°), Isabella Lari(*)

(*) Dipartimento di Scienze Statistiche, Sapienza Università Roma

(^) IRCAM, Paris, France

(°) Musician and composer

Abstract

We explore the concept of tonal signatures developed and put into musical practice by one of us (Mezzadri). A tonal signature of a scale S is a minimal subset of notes within S that is not contained in any scale S' different from S. We present a set covering model to find a smallest signature. We also show that the signatures of a scale are the prime implicants of a suitable monotone Boolean function represented by a CNF. On this ground, we introduce a more general notion of Boolean signature, depending on a Boolean operator. The computational machinery for generating Boolean signatures remains essentially the same. The richness and variety of Boolean signatures has a great potential for the development of new paradigms in polytonal harmony.

1. Basic definitions and notation

Let us denote by **N** = {C, C#, …, B♭, B} the set of the 12 notes, ordered in the stated order. A musical scale will always be regarded as a subset of **N** . All musical scales consist of 7 notes. Every scale S can be represented by a binary vector with 12 components, the *characteristic vector* ch(S) of S, defined as follows:

$$ch_i(S) = \begin{cases} 1, & \text{if } i \in S \\ 0, & \text{if } i \notin S \end{cases} \quad , \quad i = 1,\ldots,12$$

The characteristic vectors of the three scales in C (major, minor, (minor) harmonic) are displayed in Table 1. Those of the other major, minor, harmonic scales may be obtained from the characteristic vectors of the C, Cm, Ch scale, respectively, by cyclic permutations of their components. Thus, all in all, there are 36 scales.

Ital	Engl	Do Major	Do minor	Do (minor) harmonic
		C	C m	C h
Do	C	1	1	1
Do #	C #	0	0	0
Re	D	1	1	1
Mi ♭	E ♭	0	1	1
Mi	E	1	0	0
Fa	F	1	1	1
Fa #	F #	0	0	0
Sol	G	1	1	1
La ♭	A ♭	0	0	1
La	A	1	1	0
Si ♭	B ♭	0	0	0
Si	B	1	1	1

Table 1

The *scale-note* matrix is the 36×12 binary matrix A = [a_{ij}], whose columns correspond to the 12 notes (in the above order) and whose rows correspond to the 36 scales, in the order C, **C#** , … , **B♭,** B**;** C m, **C# m** , … , B♭ m, B m**;** C h, **C#** h, … , **B ♭** h, B h. Let S_j be the j-th scale in this order. By definition,

$$a_{ij} = \begin{cases} 1, & \text{if note } j \text{ belong to scale } S_i \\ 0, & \text{else} \end{cases}$$

Thus the rows of A are the characteristic vectors of the corresponding scales.

Now let S_r be any given reference scale. A *typical* set T for S_r is any subset of S_r that is not contained in any scale $S_j \neq S_r$. Clearly, any set T' such that $T \subset T' \subseteq S_r$ is also typical. So, it makes sense to look for minimal typical sets. Such sets are called *(tonal) signatures*, and they are the main notion explored in these notes.

2. Finding a smallest signature: a set covering model

The (standard) *product matrix* P = [p_{ij}] is the 35×12 binary matrix defined as follows: for each i = 1, …., 12; j≠r, j =1, …, 36, let

$$p_{ij} = \begin{cases} 1, & \text{if } j \in S_r, j \notin S_i \\ 0, & \text{else} \end{cases}$$

So, in the matrix $P \equiv P_r$ there is one row for each scale S_i , i≠r. In our notation, the row retains the same index i as S_i . So, for instance, if the reference scale is the C major one (row 1 in A), the rows of P are labelled 2, 3, … , 36, in this order.

Tables 2 and 3 show the two product matrices relative to the C major and the C# major scales, resp.

The *set covering* problem *SC(r)* associated with the reference scale S_r is defined as follows:

$$\min \sum_{j=1}^{12} x_j \qquad (1.1)$$

$$SC(r) \qquad \sum_{j=1}^{12} p_{ij}\, x_j \geq 1, \quad i \neq r \qquad (1.2)$$

$$x_j = 0 \text{ or } 1, \quad j = 1, \dots, 12 \qquad (1.3)$$

Do major product matrix

	Note	Do	Do#	Re	Mi♭	Mi	Fa	Fa#	Sol	La♭	La	Si♭	Si
	Scale												
M	Do												
A	Do#	0	0	1	0	1	0	0	1	0	1	0	1
J	Re	1	0	0	0	0	1	0	0	0	0	0	0
O	Mi♭	0	0	0	0	1	0	0	0	0	1	0	1
R	Mi	1	0	1	0	0	1	0	1	0	0	0	0
	Fa	0	0	0	0	0	0	0	0	0	0	0	1
	Fa#	1	0	1	0	1	0	0	1	0	1	0	0
	Sol	0	0	0	0	0	1	0	0	0	0	0	0
	La♭	0	0	1	0	1	0	0	0	0	1	0	1
	La	1	0	0	0	0	1	0	1	0	0	0	0
	Si♭	0	0	0	0	1	0	0	0	0	0	0	1
	Si	1	0	1	0	0	1	0	1	0	1	0	0
M	Do	0	0	0	0	1	0	0	0	0	0	0	0
I	Do#	0	0	1	0	0	1	0	1	0	1	0	1
N	Re	1	0	0	0	0	0	0	0	0	0	0	0
O	Mi♭	0	0	0	0	1	0	0	1	0	1	0	1
R	Mi	1	0	1	0	0	1	0	0	0	0	0	0
	Fa	0	0	0	0	0	0	0	0	0	1	0	1
	Fa#	1	0	1	0	1	0	0	1	0	0	0	0
	Sol	0	0	0	0	0	1	0	0	0	0	0	1
	La♭	1	0	1	0	1	0	0	0	0	1	0	0
	La	0	0	0	0	0	1	0	1	0	0	0	0
	Si♭	0	0	1	0	1	0	0	0	0	0	0	1
	Si	1	0	0	0	0	1	0	1	0	1	0	0
H	Do	0	0	0	0	1	0	0	0	0	1	0	0
A	Do#	0	0	1	0	0	1	0	1	0	0	0	1
R	Re	1	0	0	0	0	0	0	0	0	0	0	1
M	Mi♭	1	0	0	0	1	0	0	1	0	1	0	0
O	Mi	0	0	1	0	0	1	0	0	0	0	0	0
N	Fa	0	0	1	0	0	0	0	0	0	1	0	1
I	Fa#	1	0	0	0	1	0	0	1	0	0	0	0
C	Sol	0	0	0	0	1	1	0	0	0	0	0	1
	La♭	1	0	1	0	0	1	0	0	0	1	0	0
	La	0	0	0	0	0	0	0	1	0	0	0	0
	Si♭	0	0	1	0	1	0	0	1	0	0	0	1
	Si	1	0	0	0	0	1	0	0	0	1	0	0

Table 2

Do # major product matrix

	Note / Scale	Do	Do#	Re	Mi♭	Mi	Fa	Fa#	Sol	La♭	La	Si♭	Si
M	Do	0	1	0	1	0	0	1	0	1	0	1	0
A	Do#												
J	Re	1	0	0	1	0	1	0	0	1	0	1	0
O	Mi♭	0	1	0	0	0	0	1	0	0	0	0	0
R	Mi	1	0	0	0	0	1	0	0	0	0	1	0
	Fa	0	1	0	1	0	0	1	0	1	0	0	0
	Fa#	1	0	0	0	0	0	0	0	0	0	0	0
	Sol	0	1	0	1	0	1	0	0	1	0	1	0
	La♭	0	0	0	0	0	0	1	0	0	0	0	0
	La	1	0	0	1	0	1	0	0	0	0	1	0
	Si♭	0	1	0	0	0	0	1	0	1	0	0	0
	Si	1	0	0	0	0	1	0	0	0	0	0	0
M	Do	0	1	0	0	0	0	1	0	1	0	1	0
I	Do#	0	0	0	0	0	1	0	0	0	0	0	0
N	Re	1	0	0	1	0	0	1	0	1	0	1	0
O	Mi♭	0	1	0	0	0	0	0	0	0	0	0	0
R	Mi	1	0	0	0	0	1	0	0	1	0	1	0
	Fa	0	1	0	1	0	0	1	0	0	0	0	0
	Fa#	1	0	0	0	0	0	0	0	0	0	1	0
	Sol	0	1	0	1	0	1	0	0	1	0	0	0
	La♭	1	0	0	0	0	0	1	0	0	0	0	0
	La	0	1	0	1	0	1	0	0	0	0	1	0
	Si♭	0	0	0	0	0	0	1	0	1	0	0	0
	Si	1	0	0	1	0	1	0	0	0	0	0	0
H	Do	0	1	0	0	0	0	1	0	0	0	1	0
A	Do#	0	0	0	0	0	1	0	0	0	0	1	0
R	Re	1	0	0	1	0	0	1	0	1	0	0	0
M	Mi♭	1	1	0	0	0	0	0	0	0	0	0	0
O	Mi	0	1	0	0	0	1	0	0	1	0	1	0
N	Fa	0	0	0	1	0	0	1	0	0	0	0	0
I	Fa#	1	0	0	1	0	0	0	0	0	0	1	0
C	Sol	0	1	0	0	0	1	0	0	1	0	0	0
	La♭	1	0	0	0	0	1	1	0	0	0	0	0
	La	0	1	0	1	0	0	1	0	0	0	1	0
	Si♭	0	0	0	0	0	0	0	0	1	0	0	0
	Si	1	0	0	1	0	1	0	0	1	0	0	0

Table 3

The set covering problem *SC(r)* belongs to the class of integer linear programming problems, and thus it may be solved by any specialized software for such class, e.g. CPLEX. It has 12 binary variables x_i and 35 set covering constraints.

The interpretation of *SC(r)* is as follows. In the optimal solution $x = (x_1, \ldots, x_{12})$ to *SC(r)*, each binary variable x_i takes the value 1 if the corresponding note i belongs to the smallest signature sought for, and 0 else; that is, the optimal solution x to *SC(r)* is the characteristic vector of a smallest signature of S_r .

For any given *i*, by the definition of the product matrix, the variables that appear on the l.h.s. of (1.2) are associated to those notes that are present in S_r but are absent from S_i. The constraint then has the form "the sum of a bunch of variables is at least 1". Since the variables are binary, an equivalent formulation is "at least one variable in the bunch takes the value 1". Summing up, the constraint (1.2) expresses the requirement that at least one note present in S_r must be absent from S_i . Since this requirement must hold for each $i \neq r$, we conclude that x must be the characteristic vector of some typical set. Furthermore, the objective function (1.1) is equal to the number of variables that take the value 1, that is, to the cardinality of the typical set. Since such objective function is minimized, the optimal solution to *SC(r)* provides a smallest typical set. Such set must be a minimal typical set, for otherwise by deleting some note from it we would get a smaller typical set, contradicting the optimality of the set. Hence the optimal solution to *SC(r)* is indeed the characteristic vector of a smallest signature.

3. A Boolean method for finding all signatures

In this section we describe a method for finding all the signatures of a given reference scale S_r .

For the explanation of the Boolean elementary notions dealt with here, the reader may consult wikipedia items such as 'Boolean Algebras', 'Boolean Functions', 'Truth Tables', 'Logic Gates', or, for details, [2].

Given a binary variable x, its *negation* or *complement* is defined to be $\overline{x} = 1 - x$. If x, y are two binary variables their *product* (or *conjunction*) is the ordinary product xy (also written $x \wedge y$), and their *union* is $x \vee y = \max\{x,y\} = x+y - xy$. Both the union and the product are associative, commutative, and idempotent operations; they are also distributive w.r.t. each other and satisfy the absorption laws:

$$x(x \vee y) = x, \quad x \vee xy = x.$$

More generally, if A and B are products of variables, one has

$$A \vee AB = A,$$

and one says that A *absorbs* AB (a special case occurs when B = 1).

A *literal* is either a variable or its negation. A *term* is a product of literals; a *clause* is a union of literals.

We may notice that the constraint (1.2) in the set covering model *SC(r)* may be re-written as

$$\bigvee_{j=1}^{12} p_{ij} x_j = 1, \quad i \neq r \qquad (2)$$

In fact, both (1.2) and (2) express the requirement that at least one of the variables surviving on their l.h.s. must take the value 1.

The system of equations (2) is equivalent to the single equation

$$\bigwedge_{i \neq r} \bigvee_{j=1}^{12} p_{ij} x_j = 1 \qquad (3)$$

Let us denote by $\varphi_r(x)$ the Boolean function defined by the *conjunctive normal form* (CNF) on the l.h.s. of (3). That is, $\varphi_r(x)$ is written as a product of clauses. Since no variable in this CNF is negated, the Boolean function $\varphi_r(x)$ is *monotone*: $x \le y \Rightarrow \varphi_r(x) \le \varphi_r(y)$. This function will be called the *signature function* of S_r.

Equation (3) is also called a *Satisfiability* (SAT) problem. Any binary vector x such that $\varphi_r(x) = 1$, that is, any solution to the SAT problem (3), is called *a true point* (or a *truth assignment*) for $\varphi_r(x)$. If, instead, $\varphi_r(x) = 0$ then x is called a *false point*. From the above discussion, one sees that the true points of $\varphi_r(x)$ are the characteristic vectors of the typical sets of the scale S_r. A true point is said to be *minimal* if it becomes a false point whenever any of its components with value 1 is decreased to 0. Hence the characteristic vectors of the signatures of S_r are precisely the minimal true points of $\varphi_r(x)$.

From a well-known theorem on Boolean functions, it follows that there is a one-to-one correspondence between the minimal true points of $\varphi_r(x)$ and the prime implicants of such function. The prime implicants are actually the terms of the (unique) irredundant disjunctive normal form (DNF) of $\varphi_r(x)$. While a CNF is a product of unions, a DNF is a union of products, and "irredundant" means that no term in the DNF is absorbed by some other term. The one-to-one correspondence is such that, if the variable x_i appears in a prime implicant, then it must be equal to 1 in the associated minimal true point, else it must be equal to 0 in that point.

Thus, finding all signatures of the reference scale S_r amounts to re-writing the Boolean function $\varphi_r(x)$, naturally represented by a CNF, as a DNF, and "cleaning up" in the resulting expression all terms that are absorbed by some other term.

A small artificial example will illustrate the above procedure. Let $\varphi_r(x)$ be given by the CNF $(x_1 \vee x_2)(x_1 \vee x_3 \vee x_4)(x_2 \vee x_4)$. Then

$$\begin{aligned}\varphi_r(x) &= x_1 x_2 \vee x_1 x_4 \vee x_1 x_2 x_3 \vee x_1 x_3 x_4 \vee x_1 x_2 x_4 \\ &\quad \vee x_1 x_4 \vee x_1 x_2 \vee x_1 x_2 x_4 \vee x_2 x_3 \vee x_2 x_3 x_4 \vee x_2 x_4 \vee x_2 x_4 \\ &= x_1 x_2 \vee x_1 x_4 \vee x_2 x_3 \vee x_2 x_4\end{aligned}$$

- the irredundant DNF of $\varphi_r(x)$ - the latter identity following from absorption. Hence the minimal true points of $\varphi_r(x)$ are the 4 binary vectors (1,1,0,0), (1,0,0,1), (0,1,1,0), (0,1,0,1).

4. Boolean signatures

Let us go back to the definition of product matrix. An equivalent way to define such matrix is the following:

$$p_{ij} = a_{rj} * a_{ij}, \quad i \neq r; \; j = 1, \ldots, 12, \tag{4}$$

where $*$ is the Boolean operator defined by the truth table

x	**y**	**x * y**
0	0	0
0	1	0
1	0	1
1	1	0

That is, $x * y = x\,\overline{y}$ = x AND (NOT y).

Relation (4) suggests the following generalization of signatures. Just replace in (4) the above $x\,\overline{y}$ operator by an arbitrary Boolean operator *. The product matrix now depends both on the reference scale S_r and on the operator *: P = P(r,*). The rest of the procedures defined in the previous sections is unchanged: from such product matrix P(r,*) one builds the set covering problem *SC*(r, *) given by (1.1), (1.2), (1.3) and the signature function $\varphi(x; r, *)$ given by the l.h.s. of (2).

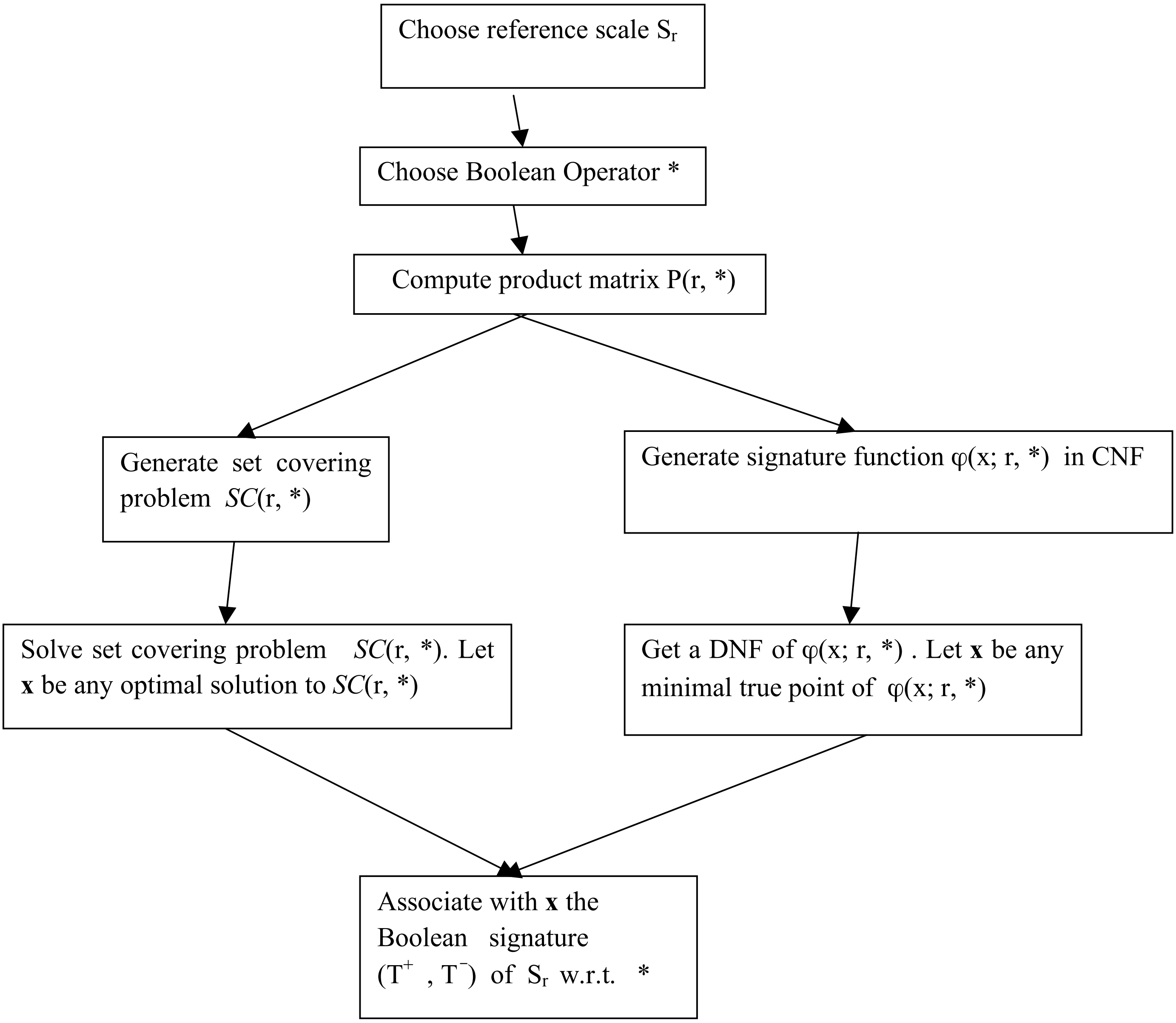
Let x be any minimal true point of $\varphi(x; r, *)$ (in particular, any optimal solution to *SC*(r, *)).

With the binary vector x we associate two subsets of N, namely,

$$T^+ = \{j: x_j = 1, \; j \in S_r\}, \quad T^- = \{j: x_j = 1, \; j \notin S_r\}. \tag{5}$$

The sets T^+ and T^- are called the *positive* and the *negative signature*, respectively, and the pair (T^+, T^-) the *Boolean signature*, of the scale S_r w.r.t. the Boolean operator *. The musical interpretation is as follows: T^+ is the set of notes inside S_r that *must* be played, while T^- is the set of notes outside S_r that *must not* be played; all the remaining notes might or might not be played, preference being given to the notes in $S_r \backslash T^+$. When there exist, for a given reference scale, several signaures, all or some of them may be sequentially performed, in a suitable order, during the execution of the piece.

The following diagram summarizes the overall procedure.



All in all, there are 16 different truth tables, giving rise to 16 different Boolean operators; of course, some of them are musically more interesting than others.

Examples

1) The 0-constant and the 1-constant operators, defined by the truth tables

x	**y**	**x * y**
0	0	0
0	1	0
1	0	0
1	1	0

and

x	**y**	**x * y**
0	0	1
0	1	1
1	0	1
1	1	1

respectively, give rise to product matrices consisting of all 0's and all 1's, respectively.

In the former case, the associated set covering problem has no solution; in the latter case, any single note corresponds to an optimal solution. Hence, these two Boolean operators are trivial and they may be ignored.

2) When * is the NAND operator,

x	**y**	**x * y**
0	0	1
0	1	0
1	0	0
1	1	0

and the reference scale is either major or minor, the set covering problem has no feasible solution. Thus also the NAND operator may be discarded unless the reference scale is harmonic.

3) When $x * y = x\,\overline{y}$, the set T^- is empty and T^+ is an ordinary signature.
4) Vice versa, when $x * y = \overline{x}\,y$ = (NOT x) AND y, the set T^+ is empty and T^- is the set of notes outside S_r that must not be played; in this case T^- will be called an *antisignature*.

5) When $x * y = x \oplus y$ = x XOR y, the truth table is

x	y	x * y
0	0	0
0	1	1
1	0	1
1	1	0

In this case, any Boolean signature (T^+, T^-) has the following property: for any scale S_j different from S_r, there exists some note that either belongs to T^+ (and hence is present in S_r), but is absent from S_j; or belongs to T^- (and hence is absent from S_r) but is present in S_j. Such (T^+, T^-) is called a *conflict signature.*

The meaning of the Boolean signatures for the different 13 nontrivial Boolean operators is explicitly given below. Each such operator is labelled f_1 f_2 f_3 f_4, where f_1, f_2, f_3, f_4 are the values of p*q for (p,q) = (0,0), (0,1), (1,0) and (1,1), respectively; and the 13 operators are sorted in reverse lexicographic order: F1 = 0001, F2 =0010, F3 = 0011, ... Thus, e.g., F6 =0110 is the XOR operator. The following short-hand notation is used: R denotes the reference scale; S denotes any other scale.

F1: 0001 – R AND S; (*shared signature*)

For each S, the signature contains at least one note shared by R and S;

F2: 0010 – R AND (NOT S) ; (*ordinary signature*)

For each S, the signature contains at least one note present in R and absent from S;

F3: 0011 – R

The signature contains (at least) one note of R;

F4: 0100 – (NOT R) AND S (*antisignature*)

For each S, the signature contains at least one note absent from R and present in S;

F5: 0101 – S (*universal signature*)

For each S, the signature contains at least one note of S;

F6: 0110 – R XOR S (*conflict signature*)

For each S, the signature contains at least one note present either in R or in S, but not in both;

F7: 0111 – R OR S
For each S, the signature contains at least one note present either in R, or in S, or in both;

F8: 1000 – R NOR S (possible only when the reference scale is harmonic)

For each S, the signature contains at least one note absent from both R and S;

F9: 1001 – R XNOR S

For each S, the signature contains at least one note that is present both in R and S, or absent from both R and S;

F10: 1010 – NOT (S) (*defective signature*)

For each S, the signature contains at least one note that is absent from S;

F11: 1011 – S IMPLIES R

For each S, the signature contains one note that cannot be absent from R when it is present in S (i.e., the note is either present both in R and S or absent from S);

F12: 1100 – NOT (R)

The signature contains (at least) one note absent from R;

F13: 1101 – R IMPLIES S (*induced signature*)

For each S, the signature contains one note that cannot be absent from S when it is present in R (i.e., the note is either present both in R and S or absent from R);

F14: 1110 – R NAND S

For each S, the signature contains at least one note that is not shared by R and S;

In any case, the signature must be *minimal* with respect to the stated property, i.e., it looses such property whenever an arbitrary note is taken off from the signature.

Remark 1. The interest of a Boolean operator depends not only on its meaning, but also on the richness of the corresponding signatures (see Catalogue in the Appendix). For example, the signatures for F3 and F12 depend on R, but not on S, and they consist of single notes, so they look uninteresting; also the signatures for F11 consist of single notes for all major and minor scales, so their possible use is confined to harmonic ones.

5. Invariance

Let us compare the two product matrices P and P' given in Tables 2 and 3, respectively.

A careful look at them shows that, once P is at hand, P' can be easily computed from P according to the following simple rule:

$$p'_{t+[h+1],[j+1]} = p_{t+1,\,j}\,, \qquad t = 0 \text{ (major scales), } 12 \text{ (minor scales), } 24 \text{ (harmonic scales), } h = 2, \ldots, 12;\ ,\ j = 1, \ldots 12\ ; \qquad (6)$$

where [p+q] is equal to (p+q) mod 12.

For example, consider the entries $p_{3,6} = p_{Re, Fa}$ in Table 2 and $p'_{4,7} = p'_{Mib, Fa\#}$ in Table 3: both are equal to 1. Or, consider the entries $p_{18,5} = p_{Fa\ m, Mi}$ in Table 2 and $p'_{19,6} = p'_{Fa\#\ m, Fa}$ in Table 3: both are equal to 0. Or, consider the entries $p_{27,12} = p_{Re\ h, Si}$ in Table 2 and $p'_{28,1} = p'_{Mib\ h, Do}$ in Table 3: both are equal to 1.
Identity (6) is but a special case of a much more general identity.

Theorem 1: For any given Boolean operator *, all the product matrices relative to a major (minor, harmonic) scale can be obtained from those relative to the C major (minor, harmonic, resp.) scale through cyclic permutations of the rows and of the columns.
More precisely, if $P^{(t,k)}$ is the product matrix relative to the scale $S_r \equiv S_{t+k}$ (t =0, 12, 24 ; k= 2, …,12) and to the Boolean operator *, the following identity holds:

$$p^{(t,k)}_{s+[h+k-1],[j+k-1]} = p^{(t,1)}_{s+h,j} \quad , \quad s = 0, 12, 24 \ ; \ h, j = 1, \ldots, 12; \qquad (s,h) \neq (t,k) \qquad (7)$$

Proof. Setting $r \equiv t+k$ and $i \equiv s + h$, let us re-write the definition (4) under the equivalent form

$$p^{(t,k)}_{s+h,j} = a_{t+k,j} * a_{s+h,j}, \quad s = 0, 12, 24 \ ; \ h, j = 1, \ldots, 12; \ (s,h) \neq (t,k). \qquad (8)$$

The cyclicity of all major (minor, harmonic) scales implies that, for each d = 1, …, 12 one has $a_{s+h,\, j} = a_{s+[h+d],[j+d]}$. Hence, by (8),

$$p^{(t,k)}_{s+[h+k-1],[j+k-1]}$$
$$= a_{t+k,[j+k-1]} * a_{s+[h+k-1],[j+k-1]}$$
$$= a_{t+1,j} * a_{s+h,j} = p^{(t,1)}_{s+h,j}$$

□

Identity (7) says the following: if the product matrix P relative to the C major (minor, harmonic) scale is at hand, in order to get the product matrix P' relative to any other major (minor, harmonic) scale whose tonic is k-1 notes (semitones) higher than C, simply copy each entry of P into the cell of P' lying k -1 positions to the right and k-1 positions below, where the positions are counted modulo 12.

Corollary 1 (*Invariance*) : The set covering problem and the signature function are the same for all major (minor, harmonic) scales, up to cyclic variable renaming.

The above Invariance property implies that, in order to get a full catalogue of all boolean signatures, it is enough to find the prime implicants of 40 (13×2+14, since F8 makes sense only for harmonic scales) Boolean functions with 12 variables and 35 clauses, that is, to find the irredundant DNF's of such functions. In order to get these DNF's, a variety of dualization algorithms is available in the literature (see [2]). The one we have chosen to implement is conceptually very simple, but it has a good performance in practice (0.08 seconds on a 2.8 GHz processor were enough to compute the full catalogue of Boolean signatures). Here is a formal description.

Write the input CNF as

$$\varphi\ (x, r, *) \ = \ \bigwedge_{i \neq r} C_i$$

where C_i *is the l.h.s. of* (2).

D := 1;

for each i = 1, …, 36, $i \neq r$

$F_i := D \wedge C_i$;

Use, if needed, distributivity and absorption to write F_i *in DNF.*

Let D be the resulting DNF;

Throughout the 'for' loop the current D maintains the property of being a DNF. At the end of the loop D yields the required DNF of φ(x; r, *) .

Along the same lines, in order to get smallest signatures for all tonal scales and all Boolean operators, it is enough to solve 40 set covering problems with 12 variables and 35 constraints each.

What is even nicer, whereas arbitrary set covering problems are NP-complete [1], and hence computationally hard to solve, all such 40 set covering problems are easily solvable by a combination of standard preprocessing rules and small-size linear programs, without any need of enumeration at all. This issue is discussed at length in a companion paper [3], where actually the following results are reported:

1) For all the 13 Boolean operators but F1, F5, F9, and F10, four simple preprocessing rules are enough to fully solve the set covering problems.
2) For the remaining four operators, (with the single exception of F5 when the reference scale is harmonic) a smallest signature can be obtained through the solution of a small-size linear program, obtained from the set covering problem by the addition of the single constraint 'sum of all variables ≥ 3'; moreover, the restriction that the variables must take the values 0 or 1 is relaxed into the requirement that their value must lie in the interval [0,1]. It turns out that, although the resulting linear program may admit fractional optimal solutions, the widely used CPLEX 10.0 LP solver always finds a binary optimal solution, thus yielding a smallest signature.
3) The exceptional case when the operator is F5 and the scale is harmonic is the only one where a smallest signature consists only of two notes. In this case it is easy to explicitly pinpoint the unique smallest signature – e.g., A and B♭ – for the C h scale.

6. The inverse problem

We have seen that with any reference scale S_r and any Boolean operator * (except when * is the operator F8 and S_r is a major or minor scale) one can associate one or more signatures. One may also look at the *inverse problem*: given any Boolean operator * and a binary vector **s** = (s_1, …, s_{12}), find all scales S_r, if any, such that **s** is (the characteristic vector of) one of the signatures associated with * and S_r.

We are going to describe a very simple procedure for solving the above inverse problem. First of all, we need to extend to arbitrary Boolean operators, in a straightforward way, the notion of ordinary typical set, given in Sec. 1.

Def. : A *typical vector* w.r.t. the reference scale S_r and the Boolean operator * is any binary feasible solution to the set covering problem $SC(S_r, *)$ (see Sec. 2); or, equivalently, any true point of the signature function $\varphi(x; r, *)$ (see Sec. 4).

When * is F2, a typical vector is but the characteristic vector of a typical set, as defined in Sec. 1.

Let

$A = [a_{ij}]$, i = 1, … , 36; j = 1, …, 12; be the scale-node matrix defined in Sec. 1;

$W = [w_{hij}]$, where $w_{hij} = a_{hj} * a_{ij}$, h = 1, … , 36; i = 1, … , 36; j = 1, …, 12;

$Q = [q_{hi}]$, h = 1, … , 36; i = 1, … , 36, where

$$q_{hi} = \sum_{j=1}^{12} w_{hij}\, s_j \,,$$

m = $[m_i]$ where m_i is the smallest off-diagonal entry of the i-th column of Q, i = 1, … , 36. Notice that each m_i is a nonnegative integer.

Lemma 1: Vector **s** is a typical vector w.r.t. the reference scale S_r and the Boolean operator * if and only if m_r is strictly positive.

Proof. Let us set, for each $i \neq r$, and for all j = 1, …, 12,

$$p_{ij} \equiv w_{rij} \,. \qquad (9)$$

Then $P = [p_{ij}]$ coincides with the product matrix P(r,*) w.r.t. the reference scale S_r and the Boolean operator *. Hence the condition $m_r > 0$ is equivalent to the system of inequalities

$$\sum_{j=1}^{12} p_{ij}\, s_j \;\geq\; 1, \quad i \neq r \quad . \qquad (10)$$

□

As a consequence, the condition $m_r = 0$ characterizes those binary vectors **s** that *are not* typical vectors w.r.t. the reference scale S_r and the Boolean operator *.

An inequality (10) will be said to be *active* if it holds as an equality.

Lemma 2: For each $i \neq r$, the following three statements are equivalent:

- the i-th inequality (10) is active;
- there exists a unique index j such that $p_{ij} = s_j = 1$;
- the i-th inequality (10) holds, but there exists an index j such that if s_j is decreased from 1 to 0, then such inequality no longer holds.

Proof. The equivalence of the above three statements is a direct consequence of the fact that both the p_{ij}'s and the s_j's are binary. □

Theorem 5: Vector **s** is a signature w.r.t. the reference scale S_r and the Boolean operator * if and only if the following two conditions hold:

(i) $m_r = 1$ (that is, at least one of the inequalities (10) is active) ;

(ii) Let $P = [p_{ij}]$ be the matrix defined by (9); and let H be the submatrix of P formed by the rows $i \neq r$ such that $q_{ir} = 1$ and by the columns j such that $s_j = 1$. Then (a) each column of H must contain at least one 1, and (b) each row of H must contain exactly one 1.

Proof. *If)* If $m_r = 1$ then all the inequalities (10) hold. By Lemma 1, vector **s** is a typical vector w.r.t. the reference scale S_r and the Boolean operator *. Consider now the submatrix H in (ii). Such submatrix must have at least one row since $m_r = 1$ and at least one column since **s** cannot be the null vector. Now let j be any index such that $s_j = 1$. By condition (ii) – (a) there is a row $i \neq r$ such that $p_{ij} = 1$. By (ii) – (b) all the other entries of this row are 0. Hence by Lemma 2 when s_j is decreased from 1 to 0 the i-th inequality (10) no longer holds. This implies that **s** is a minimal typical vector, i.e., a signature.

Only if) Conversely, if **s** is a signature, the inequalities (10) must hold and at least one of them must be active, else **s** would not be a minimal typical vector. Hence one must have $m_r = 1$. Consider the submatrix H defined in (ii). Again H must have at least one row and at least column. For each row i of H, one has $q_{ir} = 1$ and thus the corresponding inequality (10) is active. Hence by Lemma 2 row i has exactly one 1 and (ii) – (b) holds. On the other hand, for all the rows i of P, but not of H, one has $q_{ir} > 1$ and thus the corresponding inequalities (10) are not active. Therefore H cannot have a column j entirely made of 0's, since then one could decrease s_j from 1 to 0 without violating any inequality (10), against the minimality of **s**. Hence also (ii) – (a) must hold. □

The above Theorem 5 yields the following simple direct procedure for the solution of the inverse problem:

Compute the 3-*dimensional array* $W = [w_{hij}]$;
Compute the square matrix $Q = [q_{hi}]$;
Compute the vector $\mathbf{m} = [m_i]$;
Let U *be the set of those* i *for which* $m_i = 1$ (the scales i in U are candidates for being the reference scales sought for);
For each r *in* U, *check whether conditions* (ii) – (a) and (ii) – (b) *of Theorem. 5 apply; if so*, r *is one of the required reference scales.*

Acknowledgements
We are indebted to Dr. Roberto Giaccio for his implementation of the dualization code.

References

[1] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman and co., New York, 1979.

[2] P.L. Hammer and Y. Crama, *Boolean Functions* (Cambridge University Press, 2010) forthcoming.

[3] I. Lari, B. Simeone: "Boolean harmonies, and a tractable class of set covering problems", Technical Report n. 14, 2010, Department of Statistical Sciences, La Sapienza University, Rome

Appendix

A full catalogue of Boolean signatures for the C major scale and the 13 nontrivial Boolean operators

F1: 0001
1- CDB
2- CEB
3- CFB
4- CGB
5- CAB
6- CEF
7- DEF
8- EFG
9- EFA

F2: 0010
1- CEFGB

F3: 0011
1- C
2- D
3- E
4- F
5- G
6- A
7- B

F4: 0100
1- C#D#F#G#A#

F5: 0101
1- CDB
2- CEB
3- CFB
4- CGB
5- CAB
6- CA#B
7- CC#D#
8- CC#F
9- CC#F#
10- CC#G#
11- CC#A#
12- CC#B
13- CC#D
14- C#DE
15- C#DF#
16- C#DG
17- C#DA
18- C#DB
19- C#FGB
20- C#EGA#
21- CDD#
22- C#DD#
23- DD#F
24- DD#G
25- DD#G#
26- DD#A#
27- C#D#E
28- DD#E
29- D#EF#
30- D#EG#
31- D#EA
32- D#EB
33- D#GB
34- CD#F#A
35- D#FAB
36- C#D#GA
37- CEF
38- DEF
39- D#EF
40- EFG
41- EFA
42- EFA#
43- C#FA
44- CEF#A#
45- C#FF#
46- D#FF#
47- EFF#
48- FF#G#
49- FF#A#
50- FF#B
51- CF#G
52- DF#G
53- EF#G
54- FF#G
55- F#GA
56- F#GB
57- DF#A#
58- C#D#F#G#A#
59- DEG#A#
60- CDF#G#
61- DFG#B
62- CEG#
63- CGG#
64- C#GG#
65- D#GG#
66- FGG#
67- F#GG#
68- GG#A#
69- C#G#A
70- DG#A
71- EG#A
72- F#G#A
73- GG#A
74- G#AA#
75- G#AB
76- CAA#
77- DAA#
78- D#AA#
79- FAA#
80- GAA#
81- C#A#B
82- D#A#B
83- EA#B
84- F#A#B
85- G#A#B
86- AA#B

F6: 0110
1- CD#F#GA#
2- CD#F#G#A#
3- C#D#F#GA#
4- C#D#F#G#A#
5- CEF#GA#
6- CEF#G#A#
7- C#EF#GA#
8- C#EF#G#A#
9- CD#F#GB
10- CD#F#G#B
11- C#D#F#GB
12- C#D#F#G#B
13- CEF#GB
14- CEF#G#B
15- C#EF#GB
16- C#EF#G#B
17- CD#FGA#
18- CD#FG#A#
19- C#D#FGA#
20- C#D#FG#A#
21- CEFGA#
22- CEFG#A#
23- C#EFGA#
24- C#EFG#A#
25- CD#FGB
26- CD#FG#B
27- C#D#FGB
28- C#D#FG#B
29- CEFGB
30- CEFG#B
31- C#EFGB
32- C#EFG#B

F7: 0111
1- C
2- D
3- E
4- F
5- C#D#F#G#A#
6- G
7- A
8- B

F9: 1001
1- CDB
2- CD#B
3- CEB
4- CFB
5- CF#B
6- CGB
7- CG#B
8- CAB
9- CEF
10- C#EF
11- DEF
12- EFG
13- EFG#
14- EFA

F10: 1010
1- CC#D
2- C#DD#
3- DD#FF#A
4- DD#F#AB
5- C#DEFG#
6- C#DFG#A#
7- C#DFF#A#
8- C#DGG#
9- CDEF#G#A#
10- CC#D#EG
11- CC#EGA
12- CD#EG#B
13- DD#E
14- CD#EGA#
15- D#EF#GA#
16- CC#EFA
17- D#EF
18- EFF#
19- C#EFG#B
20- EFGG#B
21- CD#EGG#
22- D#EAA#
23- CC#F#G
24- DD#F#GB
25- FF#G
26- F#GG#
27- C#D#F#GA#
28- C#F#GAA#
29- CD#G#AB
30- CC#FG#A
31- CD#FG#A
32- DD#G#A
33- C#D#FGAB
34- CDFF#A
35- CFF#G#A
36- C#EFG#A
37- GG#A
38- CC#EAA#
39- C#DF#AA#
40- C#EF#AA#
41- DFF#AA#
42- G#AA#
43- CC#B
44- CDD#F#B
45- CD#F#G#B
46- CFF#B
47- CA#B
48- CEFGB
49- EFA#B
50- CEGG#B
51- DEGG#B
52- C#DFA#B
53- DD#GA#B
54- DFGA#B
55- DGG#A#B
56- D#F#GA#B
57- AA#B

F11: 1011
1- C
2- D
3- E
4- F
5- G
6- A
7- B

F12: 1100
1- C#
2- D#
3- F#
4- G#
5- A#

F13: 1101
1- CDB
2- CEB
3- CFB
4- CGB
5- CAB
6- C#
7- D#
8- CEF
9- DEF
10- EFG
11- EFA
12- F#
13- G#
14- A#

F14: 1110
1- C#
2- D#
3- F#
4- G#
5- A#
6- CEFGB