From Analysis to Surface: Generating the Surface of Milton Babbitt's *Sheer Pluck* from a Parsimonious Encoding of an Analysis of its Pitch Structure

Brian M. Bemman
David Meredith

Aalborg University
Dept. of Architecture, Design and Media Technology

MEI Conference, Charlottesville, Virginia, U.S.A
Thursday, 22 May, 2014

Intention

- Generate the surface of *Sheer Pluck* from a encoding of an analysis its pitch structure.

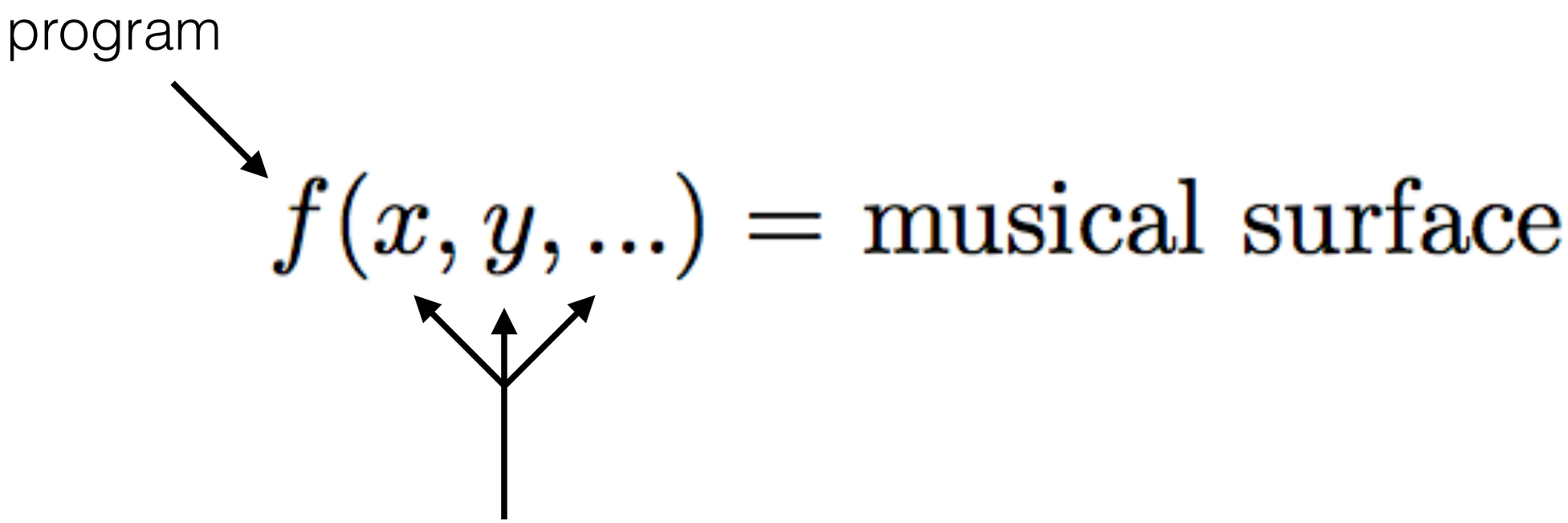
Analysis ⟶ Surface

compare with

Zip file ⟶ uncompressed file

- Generating the surface (“unziping” the analysis) proves no loss of information, i.e. the analysis contains all the information in the surface.

input and output



input: the smallest amount of information possible

Overview

1. Describe the basic properties of a six-part all-partition array.

2. Provide a template for and list of constraints on the organization of *Sheer Pluck's* pitch structure.

    2.1. Organize 48 twelve-tone rows into *hexachordally combinatorial* pairs.

3. Demonstrate the computational issues involved in automatically parsing *Sheer Pluck*.

    3.1. Parse the resultant structure into 58 distinct *integer compositions*.

1. Background

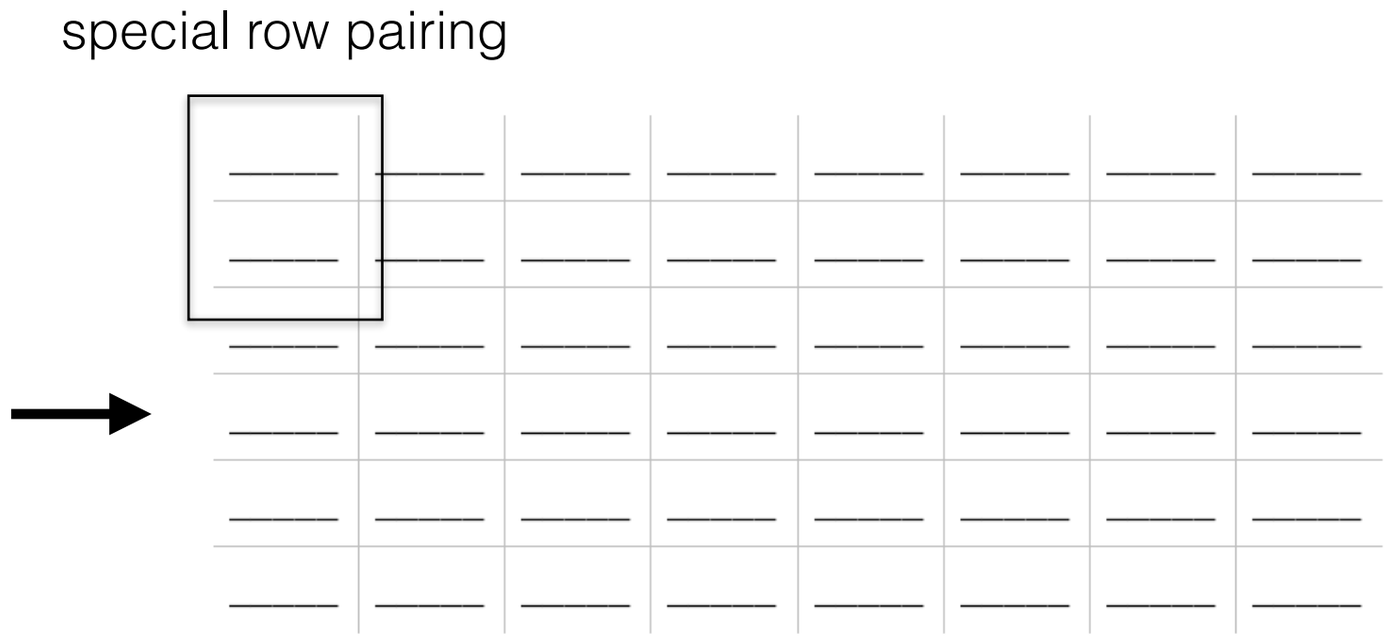
- What is an all-partition array?

Lyne $(x, \overline{x})$	2 / 3 10 11 9		9
Lyne $(\overline{x}, x)$	1 8		8
Lyne $(y, \overline{y})$	0	0 11 1 5	6
Lyne $(\overline{y}, y)$	7 /	10 9 4 2 8 3 6 7	7
Lyne $(z, \overline{z})$	6 5		5 0 10 4 11 2 3
Lyne $(\overline{z}, z)$	4 /		1
	$52^2 1^3$	84	71^5

…a twelve-tone structure organized into pairs of *hexachordally combinatorial* rows and then parsed into a sequence of discrete, vertical aggregates by distinct *integer compositions*.

Babbitt Square Row Organization

	I0	I5	I4	I6	I10	I11	I2	I9	I3	I1	I8	I7	
P0	0	5	4	6	10	11	2	9	3	1	8	7	R0
P7	7	0	11	1	5	6	9	4	10	8	3	2	R7
P8	8	1	0	2	6	7	10	5	11	9	4	3	R8
P6	6	11	10	0	4	5	8	3	9	7	2	1	R6
P2	2	7	6	8	0	1	4	11	5	3	10	9	R2
P1	1	6	5	7	11	0	3	10	4	2	9	8	R1
P10	10	3	2	4	8	9	0	7	1	11	6	5	R10
P3	3	8	7	9	1	2	5	0	6	4	11	10	R3
P9	9	2	1	3	7	8	11	6	0	10	5	4	R9
P11	11	4	3	5	9	10	1	8	2	0	7	6	R11
P4	4	9	8	10	2	3	6	1	7	5	0	11	R4
P5	5	10	9	11	3	4	7	2	8	6	1	0	R5
	RI0	RI5	RI4	RI6	RI10	RI11	RI2	RI9	RI3	RI1	RI8	RI7	



- 48 equivalent rows under P, I, R and RI organized into a **6**X8 grid = **six** part all-partition array
- Similarly, these 48 rows organized into a **4**X12 grid = **four**-part all-partition array

Row Pairing

$$\overbrace{\qquad\qquad\qquad\qquad}^{\{x\}}$$

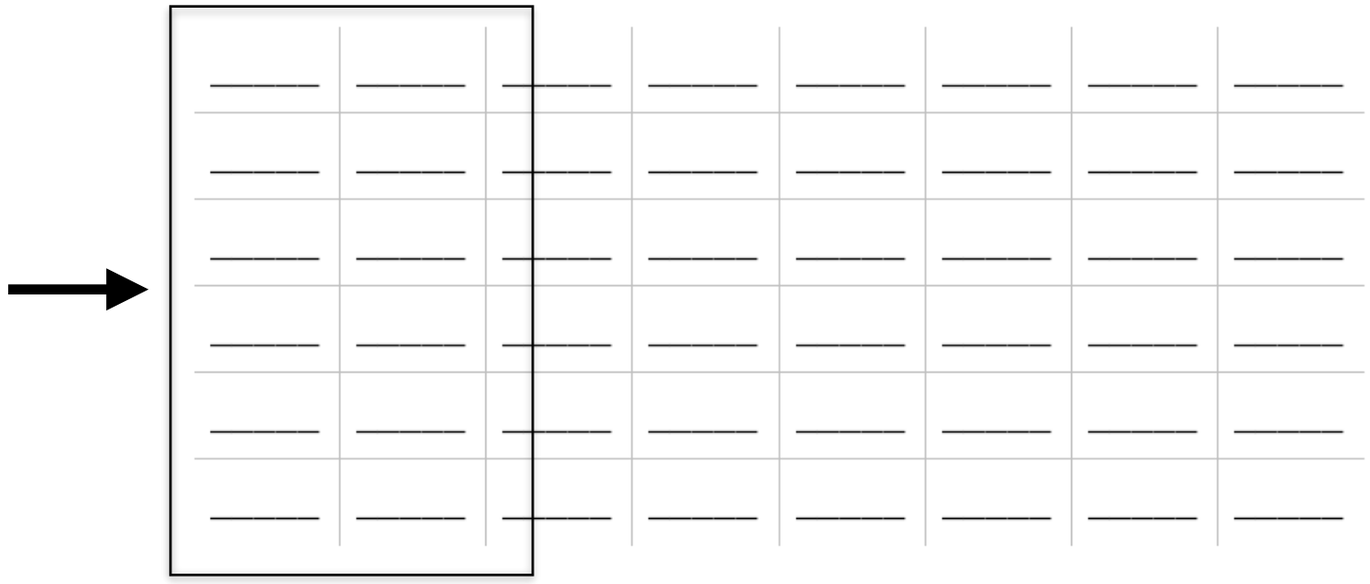
Row $(x, \overline{x})$	$(11, 4, 3, 5, 9, 10, 1, 8, 2, 0, 7, 6)$
Row $(\overline{x}, x)$	$(6, 7, 0, 2, 8, 1, 10, 9, 5, 3, 4, 11)$

$$\underbrace{\qquad\qquad\qquad\qquad}_{\{x\}}$$

- Based on the principle of *hexachordal combinatoriality*.
- *h*-related rows form both linear and vertical aggregates, four in total.

Babbitt Square Row Organization continued…

	I0	I5	I4	I6	I10	I11	I2	I9	I3	I1	I8	I7	
P0	0	5	4	6	10	11	2	9	3	1	8	7	R0
P7	7	0	11	1	5	6	9	4	10	8	3	2	R7
P8	8	1	0	2	6	7	10	5	11	9	4	3	R8
P6	6	11	10	0	4	5	8	3	9	7	2	1	R6
P2	2	7	6	8	0	1	4	11	5	3	10	9	R2
P1	1	6	5	7	11	0	3	10	4	2	9	8	R1
P10	10	3	2	4	8	9	0	7	1	11	6	5	R10
P3	3	8	7	9	1	2	5	0	6	4	11	10	R3
P9	9	2	1	3	7	8	11	6	0	10	5	4	R9
P11	11	4	3	5	9	10	1	8	2	0	7	6	R11
P4	4	9	8	10	2	3	6	1	7	5	0	11	R4
P5	5	10	9	11	3	4	7	2	8	6	1	0	R5
	RI0	RI5	RI4	RI6	RI10	RI11	RI2	RI9	RI3	RI1	RI8	RI7	



Blocks and Row Types

	Block I	Block II	
	$\overbrace{}^{\{x\}}$	$\overbrace{}^{\{x\}}$	
Lyne $(x, \overline{x})$	$(11, 4, 3, 5, 9, 10, 1, 8, 2, 0, 7, 6)$	$(5, 4, 11, 9, 3, 10, 1, 2, 6, 8, 7, 0)$	...
Lyne $(\overline{x}, x)$	$(6, 7, 0, 2, 8, 1, 10, 9, 5, 3, 4, 11)$	$(0, 7, 8, 6, 2, 1, 10, 3, 9, 11, 4, 5)$	
	$\underbrace{}_{\{y\}}$	$\underbrace{}_{\{y\}}$	
Lyne $(y, \overline{y})$	$(5, 6, 11, 1, 7, 0, 9, 8, 4, 2, 3, 10)$	$(1, 0, 7, 5, 11, 6, 9, 10, 2, 4, 3, 8)$	...
Lyne $(\overline{y}, y)$	$(2, 9, 10, 8, 4, 3, 0, 5, 11,1, 6, 7)$	$(4, 9, 8, 10, 2, 3, 6, 1, 7, 5, 0, 11)$	
	$\overbrace{}^{\{z\}}$	$\overbrace{}^{\{z\}}$	
Lyne $(z, \overline{z})$	$(0, 5, 4, 6, 10, 11, 2, 9, 3, 1, 8, 7)$	$(4, 5, 10, 0, 6, 11, 8, 7, 3, 1, 2, 9)$	...
Lyne $(\overline{z}, z)$	$(1, 8, 9, 7, 3, 2, 11, 4, 10, 0, 5, 6)$	$(3, 2, 9, 7, 1, 8, 11, 0, 4, 6, 5, 10)$	

- A *row type* refers to a row's hexachord content. Concatenated rows (often of the same type) are called **lynes**.

- The columns of the 6X8 grid are called **blocks**.

Overview

✓ 1. Describe the basic properties of a six-part all-partition array.

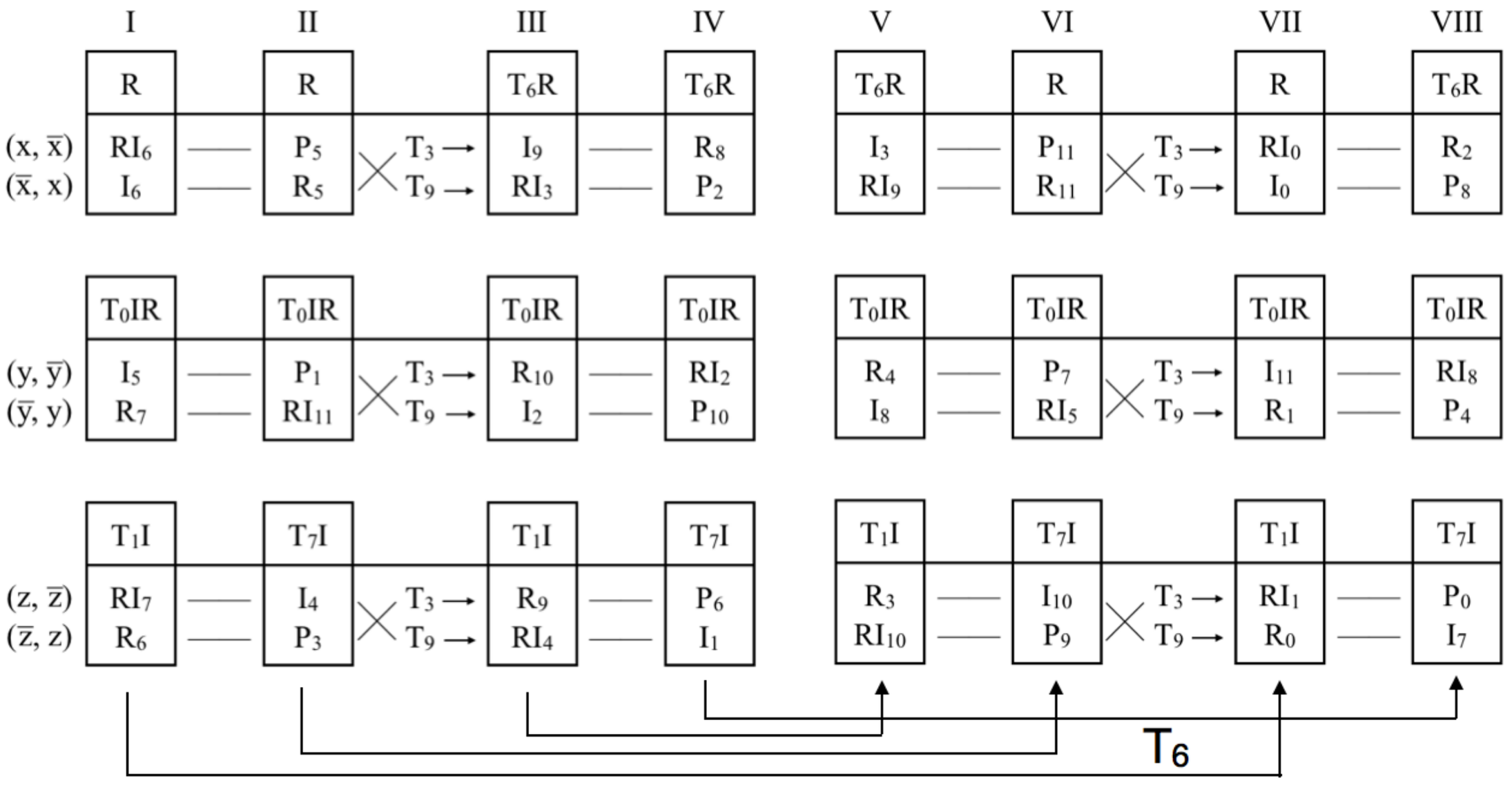
⇨ 2. Provide a template for and list of constraints on the organization of *Sheer Pluck's* pitch structure.

2.1. Organize 48 twelve-tone rows into hexachordally combinatorial pairs.

3. Demonstrate the computational issues involved in automatically parsing *Sheer Pluck*.

3.1. Parse the resultant structure into 58 distinct integer compositions.

2.1. Row Template of Babbitt's *Sheer Pluck*



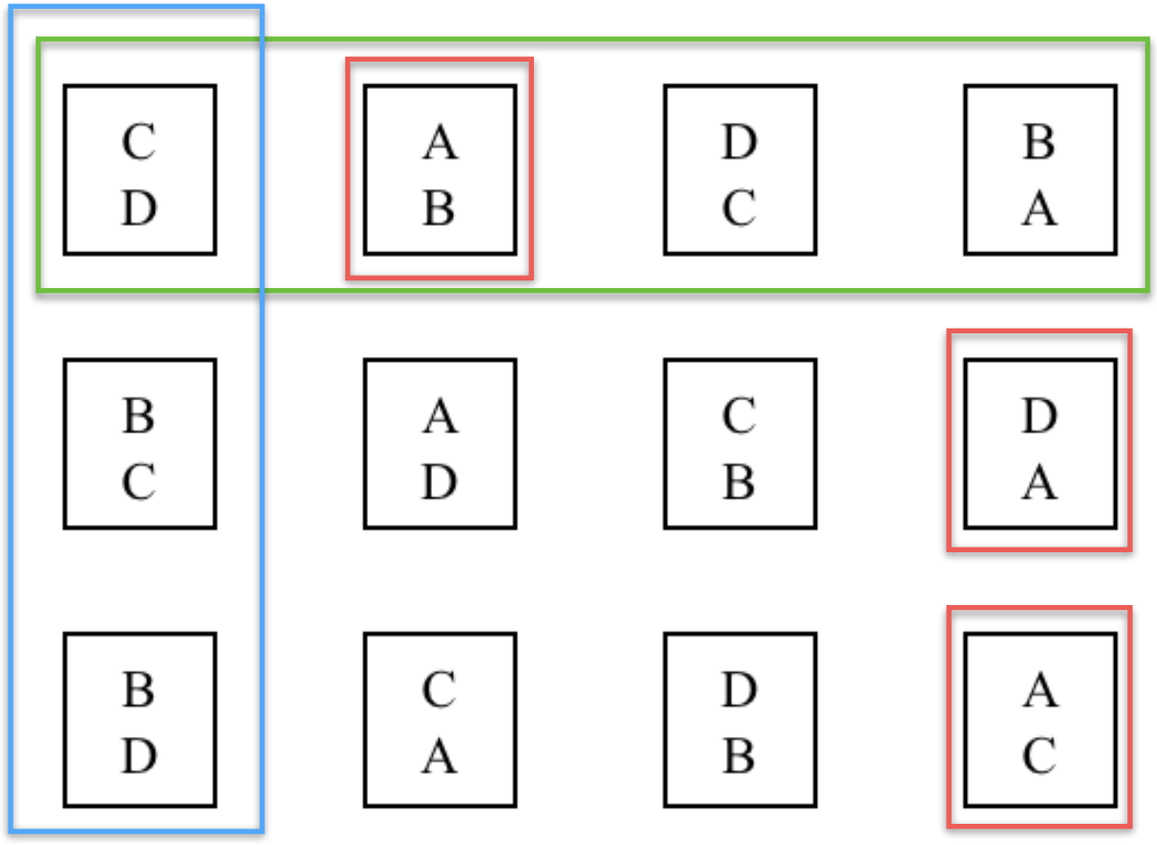
Complement Transformations, T_3 and T_9

Where

$$\{A, B, C, D\} = \{P, I, R, RI\}$$

Any given pair of *h*-related rows, x and y…

$$\{\{x, y\} : x \neq y \wedge \{x, y\} \subset \{A, B, C, D\}\}$$



Any given pair of h-related rows in a single lyne pair, x? and y?…

$$\{\{x_0, y_0\} : x_0 \neq y_0 \wedge (x_0 \in \{A, B\}, y_0 \in \{A, B\}) \cap (x_0 \in \{C, D\}, y_0 \in \{C, D\})\}$$

$$\{\{x_1, y_1\} : x_1 \neq y_1 \wedge (x_1 \in \{A, D\}, y_1 \in \{A, D\}) \cap (x_1 \in \{B, C\}, y_1 \in \{B, C\})\}$$

$$\{\{x_2, y_2\} : x_2 \neq y_2 \wedge (x_2 \in \{A, C\}, y_2 \in \{A, C\}) \cap (x_2 \in \{B, D\}, y_2 \in \{B, D\})\}$$

Any given block of *h*-related rows, p and q…

$$\{\{p, q\} : p \neq q \wedge p = \{x_0, y_0\} \cup \{x_1, y_1\} \cup \{x_2, y_2\} \not\ni A, q = \{x_0, y_0\} \cup \{x_1, y_1\} \cup \{x_2, y_2\}\}$$

Overview

✓ 1. Describe the basic properties of a six-part all-partition array.

2. Provide a template for and list of constraints on the organization of *Sheer Pluck's* pitch structure.

✓ 2.1. Organize 48 twelve-tone rows into hexachordally combinatorial pairs.

⇨ 3. Demonstrate the computational issues involved in automatically parsing *Sheer Pluck*.

3.1. Parse the resultant structure into 58 distinct *integer compositions*.

3.1. Parsing the Pitch-class Structure

- Automating the organization of pitch structure is relatively straightforward. Parsing it however, is not a computationally trivial problem to solve.

- Some definitions…

Integer Compositions

- In number theory, an **integer composition** is a way of representing an integer *n* as an *ordered* sum of positive integers.

When *n* = 12

$$3+3+2+2+1+1 \neq 2+3+2+1+3+1$$

- In an all-partition array, we must include zero in many integer compositions. We call such instances **weak integer compositions**.

When *n* = 12

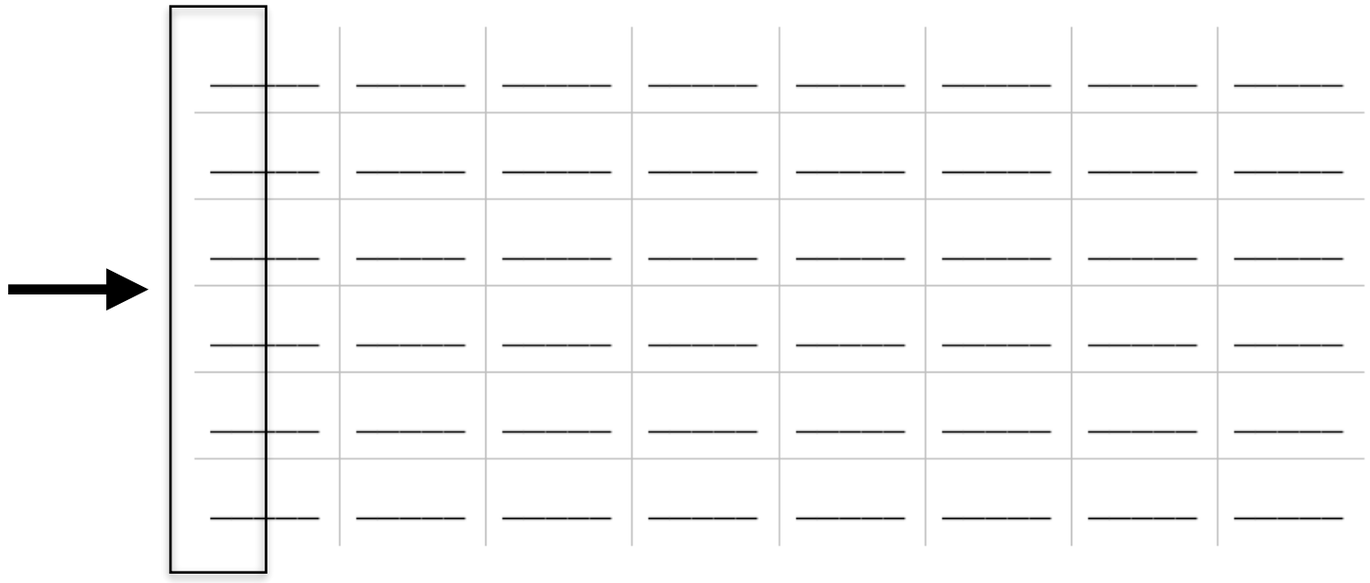
$$6+6+0+0+0+0 \neq 0+6+0+0+6+0$$

- There are 6,308 *ordered* ways of representing 12 in 6 parts, but only 58 un*ordered* ways.

- A six-part all-partition array is constructed of a distinct sequence of 58 compositions for which there are 58! possibilities.

some compositions in a block…

	I0	I5	I4	I6	I10	I11	I2	I9	I3	I1	I8	I7	
P0	0	5	4	6	10	11	2	9	3	1	8	7	R0
P7	7	0	11	1	5	6	9	4	10	8	3	2	R7
P8	8	1	0	2	6	7	10	5	11	9	4	3	R8
P6	6	11	10	0	4	5	8	3	9	7	2	1	R6
P2	2	7	6	8	0	1	4	11	5	3	10	9	R2
P1	1	6	5	7	11	0	3	10	4	2	9	8	R1
P10	10	3	2	4	8	9	0	7	1	11	6	5	R10
P3	3	8	7	9	1	2	5	0	6	4	11	10	R3
P9	9	2	1	3	7	8	11	6	0	10	5	4	R9
P11	11	4	3	5	9	10	1	8	2	0	7	6	R11
P4	4	9	8	10	2	3	6	1	7	5	0	11	R4
P5	5	10	9	11	3	4	7	2	8	6	1	0	R5
	RI0	RI5	RI4	RI6	RI10	RI11	RI2	RI9	RI3	RI1	RI8	RI7	



A portion of a single parsed block…

11 4 3 5		9 10 1 8 2	…
6 7	0 2		…
5	6 11 1	7	…
2 9	10 8 4 3	0 5 11	…
0	5	4 6	…
1 8	9 7	3	…

$$Compositions = \left| \begin{matrix} (4,2,1,2,1,2) \\ (0,2,3,4,1,2) \\ (5,0,1,3,2,1) \end{matrix} \right| \left| \begin{matrix} \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{matrix} \right| = 58\, compositions$$

A problem…

$$58 \cdot 12 = 696$$

$= 576\,pcs$

$+120\,pcs$

n distinct objects into k distinct boxes where *k* = 48 and *n* = 120

$$48^{120} = 5.61 \cdot 10^{201}$$

$$\approx 1 \cdot 10^{80}$$

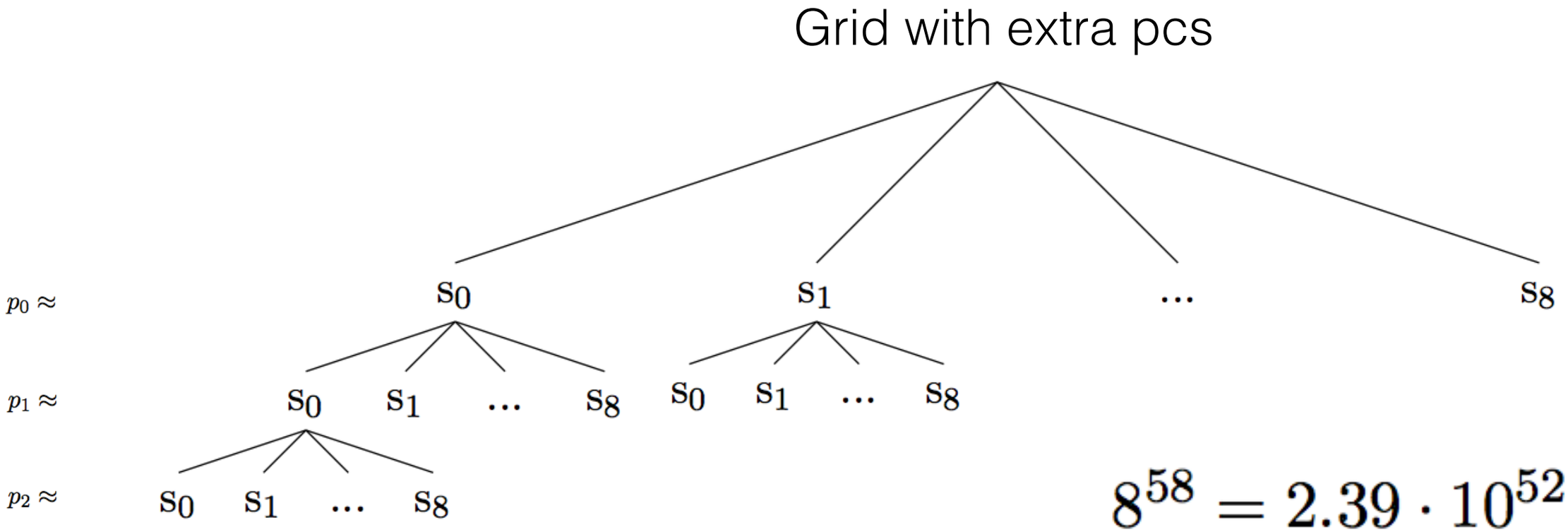
A portion of a single parsed block with extra pcs…

2 3 10 11 **9**		**9**	…	= 118
7 **8**		**8**	…	= 117
0	**0** 11 1 5	6	…	= 111
1	10 9 4 2 8 3 6 **7**	**7**	…	= 114
6 **5**		**5** 0 10 4 11 2 3	…	= 118
4		1	…	= 118

= 696

$(5, 2, 1, 1, 2, 1)$ $(0, 0, 4, 8, 0, 0)$ $(1, 1, 1, 1, 7, 1)$

Brute force search for possible successful integer compositions in *Sheer Pluck*



Where p_x is the ordinal position of a given composition and *s* is a successful integer composition.

Conclusion…

$$f(\text{musical surface}) = \text{musical surface}$$

- Alternatively, give as input the sequence of compositions.

Questions for Future Research

- Are there additional constraints in the pitch-class structure that will limit the number of calculations required in finding successful integer compositions?

  – Yes, there must be…Likely a relationship between repeated pcs and compositions. Greedy algorithm, heurstics?

- MEI – Database of not just Babbitt's works but analyses as well?

Acknowledgements

This talk was prepared while working as a Ph.D. fellow on a collaborative EU project entitled “Learning to Create” (Lrn2Cre8). The Lrn2Cre8 project acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859