Algorithmic Composition with Project One™:

An Introduction to Score Synthesis

Otto Laske PhD PsyD

Composer & Music Theorist

Course Agenda

- Esthetics of musical composition by computer (3-14)
- How Project One works (15-21)
- About Parameters (22-28)
- PR1 Menus (30-41)
- Design of Musical Form (42-49)
- Compositional Procedures (50-55)
- Output Examples (56-65)
- References (66)

Esthetics of Music Composition by Computer

What is Algorithmic Composition?

- ‘Algorithmic’ composition uses computer algorithms to create data for writing scores
- Scores so produced can be used in instrumental, vocal, Midi, and electroacoustic composition
- Algorithmic composition is not a tool for ‘producing junk fast’
- Rather, it requires reflection on the compositional process that might lead to creating a piece of music

History of Algorithmic Composition

- Algorithmic composition starts in the 13th century in the Ars Nova of France
- It continues in the 17th century with Bach
- It is revived in the 20th century with the ‘Second Vienna School’ (Schoenberg)
- It starts anew in the 1950s, with Xenakis & Koenig in Europe, and with Hiller and Babbitt in the U.S.

Why Compose with Algorithms?

- Rather than writing ‘bottom up,’ starting with local events, the composer can realize large scale designs ‘top down’
- The composer can work from a ‘deep structure,’ embodied in a ‘base score,’ thereby unifying all events in a composition
- The composer then works with variants of the base score creating a cohesive and developmental form

Deep vs. Surface Structure

- Music occurs in time
- Music evolving in time has an audible ‘surface structure’ sometimes expressed in notation
- Underneath the surface structure, there lies a ‘deep structure’ one cannot ‘see,’ made up of decisions about relationships between so-called ‘musical parameters’
- One and the same deep structure can manifest in different surface structures

What are Musical Parameters?

- Musical ‘parameters’ differ by composition
- There are certain basic or universal parameters, such as pitch (tone height), duration (tone length), time delay (interval between tone onsets), loudness (tone volume), and loud speaker location
- In algorithmic composition, we set up relationships between these parameters according to the constraints built into a particular program called a ‘score generator’

The Score Generator Project One

- There are many different score generators, depending on different notions of what is ‘composition’
- Project One by G.M. Koenig is a ‘classic’ program created in 1967 and evolved ever since
- Koenig’s program is unique in its focus on chordal structure that is to be ‘horizontalized’ in time by the composer according to his/her own esthetic principles

Project One is Problem Posing Device

- Project One was originally created to understand compositional decision making
- In defining input for the program, the composer is seen as defining an esthetic ‘hypothesis’ to be realized by computation
- Computed material has to be interpreted by the composer, either notationally or through sound, to become ‘music’
- Material can be interpreted for instruments, voices, Midi, or the electroacoustic medium

Where ‘Music’ Resides

- Using Project One makes it obvious that music resides in the mind, not in sound per se
- In algorithmic composition, ‘music’ resides in the interactive relationship between the human mind and the machine
- The computer thus becomes ‘the artist’s Alter Ego’ (Laske, 1990)
- We are using computer programs to learn about ourselves as musical minds

Using PR1 for Musical Thinking

- The ‘screen’ feature under ‘Output’ allows you to preview results (after having activated the CREATE button)
- Although it takes some time to develop a ‘feel’ for whether a particular output is what you want, you should use the screen feature as soon as possible
- A main idea in PR1 is to learn how to redefine your input to get the musical results you intend to produce
- Use Midi as a ‘sketch pad,’ to get a notion of what your score may sound like

Three Uses of PR1

- PR1 has two principal uses:
  - computing a score for transcription into conventional music notation
  - computing a score for making electroacoustic music (loudspeaker music) using the compositional languages called Csound or Kyma
  - Midi is only used as a ‘sketch pad’ and ‘debugging aid’
- In this course, we are only concerned with uses no. 1 and 3

Three Steps

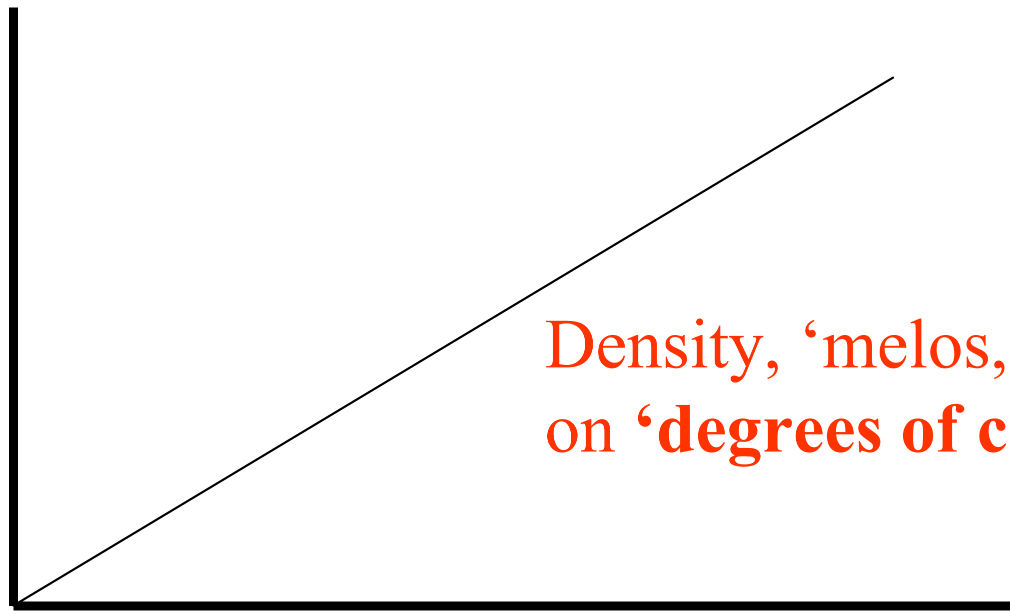
- Composing with PR1 entails three steps:
    - Input of numerical data encoding a compositional plan or hypothesis
    - Computation of the PR1 score
        - computation proper (‘Create’)
        - optional modification and further processing (‘recalculation’ and ‘horizontalization’)
    - Output of score data
        - symbolic (numerical) output
        - signal processing output

How Project One Works

PR1 Hypothesis

PR1 deconstructs 'music' into two main dimensions: 'harmony' based on chords, and time based on 'entry delays.' All other parameters flow from these two.

Harmony, defined intervallicly, and expressed in terms of **chords**



Density, 'melos,' texture, based on **'degrees of change'**

Time, defined in terms of time intervals called **'entry delays,'** giving rise to 'rhythm' through horizontalization of chords

In PR1, a 'pitch' is a chord of size 1

Degrees of Change

- PR1 conceives of music as being a process measurable in terms of degrees of underlying, ‘deep structure,’ change
- There are seven ‘system processes’ (P1 to P7) by which deep structural change is defined, from maximum change (P1) to relative stability (P7), with P4 as a mixture of the two
- The composer structures sections of movements or pieces in terms of *ideas about degree of change*
- Musical form is thus based on degrees of increasing and diminishing change of parameter values over time
- Such change is translated to linear time or ‘surface structure’ depending on choices made by the composer

Notion of ‘Base Score’

- The composer using PR1 defines an underlying score, or ‘base score’ -- a conceptual framework for many possible variants
- Through systematic changes to the base score and through horizontalization of chords, either manual or automated, the composer creates ‘score variants’
- Any number of score variants of a base score can be produced, and can be either sequenced or mixed
- The base score can be used to unify sections, movements, entire pieces, or whole series of pieces
- In this way, the base score acts as a unifying force for a multitude of variations, forming a DEEP STRUCTURE not necessarily immediately obvious to the listener

Musical Parameters

- Parameters are computable aspects of sound
- Score generators differ by the parameters they compute
- PR1 requires user input for five main parameters: tone color (instrument), entry delay (time delay, tempo), pitch, register, dynamics
- Pitch is a 'chord of size 1,' and is determined by specifying intervals that build chord structures, or 'harmony'
- In electrocoustic and Midi uses, these chord structures are 'read' by 'instruments' formatted according to particular sound synthesis languages
- In instrumental and vocal composition, chords are 'read' and 'horizontalized' note by note by the composer

Parametrical Composition

- Composition with PR1 is ‘parametrical’
- The composer specifies, not single tones or durations, but columns of parameter values (e.g. ‘entry delays’) in terms of degree of change over time
- By changing one or more parameters, we compute variants of an underlying base score, thereby creating motivic and harmonic relationships between different sections of a movement or piece

The Composer is in Charge

- The composer can specify up to 14 sections at a time
- PR1 leaves the sequencing of compositional sections to the composer
- The composer can also choose to compute a single section
- Sections differ in terms of the ranking of parameters in terms of seven ‘degrees of change’
- We design scores based on ideas of degree of change throughout a section, movement, or piece
- The change computed is deep structure change, and underlies all audible changes

About Parameters

‘Entry Delay’

- PR1 is based on the distinction between ‘entry delays’ (time delays) and durations *which are independent of each other*
- An ‘entry delay’ is the delay (time interval) between one tone onset and a subsequent one
- When not using horizontalization, only entry delays are computed, not durations
- Durations can cover the entire span of an entry delay, or just a part of it; in the latter case, a ‘silence’ or ‘rest’ is created; in manual composition, they can also extend beyond the entry delay

Entry Delays are Linked to Chord Size

- The composer defines up to 14 entry delays
- For each generic entry delay, he/she chooses the range of chord sizes (number of tones) that can be placed into a particular entry delay
- Chord size ranges from 1 to 6 tones
- The crucial relationship of entry delay to chord size is entirely open to compositional choice
- The shortest entry delay is the fraction 1/9, but there is no lower limit for decimal specification
- The actual entry delay depends on the tempo

Entry Delays Come in Three Flavors

- Entry delays can be specified as fractions, decimals, or ‘metrically’
- For symbolic representation in scores, we use fractions (0/0 to 999/1), to be converted to rhythms and meter by the composer; we may also use automated metrical representation dependent on choices of how to subdivide metric units (triplets, quintuplets, septuplets, nonuplets)
- For electroacoustic output, we use decimals
- In this course, we focus on instrumental-vocal music, using Midi as a sketch pad to test deep-structure frameworks

‘Harmony’ is Defined by Intervals

- In PR1, harmony is defined by choosing 4 intervals (2 interval pairs); pair no. 2 can be a restatement of the first
- Possible interval combinations (able to be transposed to yield 12-tone rows) are shown on line
- Intervals are indicated in terms of semi-tones from 1 to 11
- Negative numbers prevent inversion (upward & downward)
- Audible harmony is dependent on chord size, and whether horizontalization is used or not
- It also depends on choice of instruments, of course

‘Register’ is a Very General Notion

- Registers are not necessarily octave registers, as translating from PR1 to Midi or the electroacoustic medium
- In symbolic (instrumental/vocal) notation, they can be interpreted as arbitrary subdivisions of a <u>single instrument</u> (tone color regions, or even distinctions of playing mode)
- They can also be used to differentiate a single instrument <u>within a group of instruments</u> (say, ‘1’ = bass flute,’ ‘2’ = regular flute, ‘3’ = piccolo)
- In short, registers are regular or irregular ‘tone color subdivisions’ within a real or imagined acoustic continuum

‘Melody’ Depends on Register and Entry Delay

- In PR1, ‘melody’ (better: melos) is a result of combining 4 parameters: pitch, entry delay (duration), register, and tone color
- When thinking of melody in the conventional sense, we think of moving within a certain registral range (in terms of PR1: register defined by P7 or P6)
- Schoenberg introduced ‘klangfarbenmelodie’ (tone color melody) to indicate degrees of (ir-) regularity; such degrees may be distributed over many different registers (as defined by system processes P1 or P2)
- Highly diffuse ‘melos’ can be defined by system processes P1 to P3

Regularity of Meter is a Matter of Entry Delay

- In PR1, metric regularity is a result of defining entry delays (tone onsets) associated with chords
- Metric redundancy stems from redundancy of entry delay (ED defined by system process P5 to P7), and appears especially strongly when chords are not horizontalized and chordsize is high
- Regular rhythms can be reinforced by use of instruments (tone color) or by use of chords (intervals), or can be made to vanish by using system processes P1-P3

PR1 Menus

Overview of Menus for Input

- PR1 has one main and several subordinate screens
- On the main screen, the composer defines ‘input data’ that determine the structure of the score to be computed
- On an associated screen, the composer can ‘horizontalize’ the chordal scores, although this is not required for using Midi or electroacoustic instruments
- ‘Horizontalization’ entails making decisions about order and duration of single notes, as required in manual notation for instruments and voices, and can be either manual or automated (as for Midi, Csound, and Kyma)

Main Screen

- The main screen has four parts:
  – parameter value repertory
    - select the range of parameter values to be used
  – selection processes (‘branching table’)
    - specify how to select from the repertory, in terms of degree of change (‘system processes’ P1 to P7)
  – texture definition
    - define desired texture by specifying ‘time’ (entry delay) and ‘harmony’ (chord size and frequency of occurence)
  – presentation specification
    - fractional
    - decimal
    - metrical

Main Screen Visualized

Use the CREATE button to start generating score data and before saving.
CREATE lets you know about missing data and an inconsistent specification.

Define the range of values to be used for the main parameters

Length is determined by no. of 'lines'

Specify entry delays and associated chord sizes and their frequency of occurrence

Define degrees of change via seven system processes, per section

There are 7 processes, P1 to P7

Specify nature of time representation (fractions, decimals, metric specification), and accumulation*

* Accumulation determines how rhythmic values are presented; when using conventional notation, use metric specification

Subsidiary Screen 1: Options

- There are four choices:
  - Recalculation: recalculate the branching table determine parametrical structure
  - Horizontalization: make decisions about order and duration of tones and tone groups (this is optional, and neither required for, nor identical with, presentation of PR1 output for electroacoustic and Midi instruments)
  - Calculate entry delays in terms of a geometric series within limits defined in seconds
  - Set tab for branching to define parametrical structure (affects only branching table) [mere logistics]

Recalculation

- The ‘branching table’ defines the degree of change for each parameter for each section computed
- This table can be redefined by hand or recalculated automatically, one parameter at a time
- ‘Recalculation’ means resetting computational processes for chosen parameters in the sections indicated (P1 to P7)
- Through recalculation, only one parameter can be changed at a time; however, change of entry delays also changes pitches and, thereby, registers: in short, the effect is *systemic*

Horizontalization

- **Horizontalization is optional**, and has nothing to do with presentation of data for use in Midi or electroacoustics
- Horizontalization linearizes chordal structures in a way specified by the composer
- In the score table, the horizontalized tones are indicated underneath the respective chord, marked by a preceding '-'
- Horizontalization can be made dependent upon instrument, entry delay, and/or chord size
- In addition, you need to specify how exactly chords are to be 'taken apart' into linear strings of single 'notes'
- Horizontalization applies to all instruments selected for the purpose
- **It is best to experiment with different settings, to see what input creates what output, thereby composing variants of a base score whose root cause you fully understand**

How to Specify Horizontalization

Specify dependence of horizontalization on instrument, entry delay, and/or chord size

Specify how exactly horizontalization is to be carried out, in terms of (1) time range, (2) grouping, (3) periodicity, (4) duration, (5) note sequence, and (6) dynamics

(in manual, ‘note by note,’ composition intuitive)

Horizontalization Example

- Horizontalization depends on:
    - instrument 1-4, entry delay 1-5 (sum up 1-5 successive entry delays as the range to be used in horizontalization, where different durations result from differently defined ranges), chord size=3-6
    - time range (no. of entry delays)=7, grouping=single (notes), periodicity=irregular, duration=0.4 sec. to maximum/random, sequence=upward, register respected, dynamics= next chord, diminuendo.
- Here, horizontalization *depends on* all three parameters named (instr, entry delay, chord size), and is *specified in terms of* number of entry delays forming a 'time range,' groupings to be formed, regularity of grouping, minimum to maximum duration, direction, and adjustment of dynamics

Calculate Entry Delays

- To calculate entry delays automatically, you specify:
  - the number of delays
  - the upper and lower limit, in seconds
- This allows for arbitrary series of entry delays to be computed between limits freely chosen by the composer
- Use of the D=>F and F=>D buttons brings about the adjustment of the two columns

Subsidiary Screen 2: Output

- This screen offers three options:
  - output to the screen: for inspecting output on the screen
  - output to a file: for saving the data to a file
  - output to the printer: for printing out the data

Subsidiary Screen 3: Play

- This screen offers 4 choices
  - Midi
    - **serves as a "sketch pad" only, not as an autonomous representation of computed scores**
  - Csound (electroacoustic score)
  - Kyma (electroacoustic score)
  - Settings (electroacoustic only)
    - Tuning (concert pitch)
    - Panning (location)
    - Dynamic values (ppp to fff)

Design of Musical Form

Musical Form

- In PR1, musical form is computed through a feedback loop between the artist's mind and the computer program
- The program contains the composition's "grammar," while the artist's mind contains the living musical knowledge
- Musical form is based on reflection, the process of designing parametrical structures that change over time in ways predetermined by PR1 *system processes*
- Musical form is 'envisioned' by the composer; his decisions are determined by envisioning an overall form
- As in software engineering, the notion is "garbage in, garbage out"
- In this way, PR1 challenges and develops musical thinking

Composing “Top Down”

- The major difference between your previous compositional experience and use of PR1 is that you are designing compositions ‘top down,’whether a single section or an entire composition
- PR1 challenges you to develop compositional ideas that are more abstract, pertaining to longer stretches of time and to how to shape musical form
- You are forced to think about how one section or movement is going to differ from another, and how the sequence of sections is going to constitute a convincing musical form
- You carry out planning in terms of degrees of change for different parameters, and think about how different parameters relate to each other (parametrical counterpoint)

Compositional Processes

- There are seven processes computing material according to different degrees of change, P1 to P7
- P1 entails maximal change, while P7 entails redundancies and repetitions (of instruments, entry delays, chords, registers, and dynamic levels); P4 is a compromise between change and stability
- Let’s say you define a section as follows:
  - instrument =1, entry delay (“rhythm”) = 6, pitch (chord, “harmony”) = 3, register = 5, dynamics = 2
  - this means instruments (tone colors) change constantly, rhythm is not quite ‘metric’ but stable, pitch changes moderately quickly, register is relatively stable (creating ‘melos’), and dynamics is highly changeable
  - in this way, you define the ‘character’ or ‘gesture’ of sections, and introduce the possibility of contrast and transition

Systemic Approach

- Since there are 7 system processes (P1 to P7), you could design a piece of seven sections or movements in which no section would be like another
- Every parameter would use all seven system processes in some order (1-7 or 7-1)
- You would thus compose based on “tendencies” towards or away from maximal or minimal change
- You could then use ‘parametrical counterpoint” to set one parameter up against another, e.g.: instrument 1=>7, “rhythm” (entry delay) 7=>1, etc.
- In this way, you would be defining the ‘deep structure’ of your composition based on tendencies of parametrical change
- WHAT AN ADVENTURE!

Defining Contrast

- Let’s say you are defining two sections, #1 and #2
- The first uses the formula: instrument =1, entry delay (“rhythm”) = 6, pitch (chord, “harmony”) = 3, register = 7, dynamics = 2
- There are many ways for you to make #2 a “constrasting” section
- For instance, you could move away from the ‘melic’ character of #1 (register=7) and its ‘metric’ feel (entry delay=6) to a wide distribution of tones in registers (register=1) and diffuse rhythm (entry delay=1)
- Depending on your chordal structure (chord sizes), you will end up with a very different esthetic result even if you leave the other processes (instr, pitch, dynamics) in place

Knowing Your Instruments

- Limits up to which scores ‘make sense’ differ in the vocal, instrumental, Midi and electroacoustic domains
- Writing for instruments, you may want to honor conventional limits as to what can be performed
- If you stretch those limits, you have to make sure your score can be performed without strain
- These acoustic limits are reduced in using Midi, and even more strongly disappear in the electroacoustic domain
- In the electroacoustic domain, (thank God) there are other limits having to do with which scores match which digital orchestra, something we don’t need to think about here.

Duration as a 'Free' Parameter

- Shaping texture, harmony, melody, and character of your music **by way of decisions regarding duration** is the hall mark of PR1 composition
- **Durations are independent of entry delays, and can be shorter or longer than, or equal to, entry delays**
- In instrumental and vocal composition, durations are chosen manually, dependent upon context; in Midi and electroacoustic music, they are computed automatically and without knowledge of context (but judged contextually by the interpreting composer)
- One and the same score can sound very differently depending on use of durations
- Combined with other parameters, duration is a powerful tool to shape your music (and not only its texture)

Compositional Procedures

Options for Beginners

- To begin, it is best to start simple
- There are two main options for 'composing a piece', say, of four movements:
  - compute the 4 sections independently, one section at a time, using <u>different</u> parameter repertories and different selection processes for each section, in a way reflecting your notion of contrast and transition between sections
  - compute the 4 sections simultaneously, using <u>one and the same</u> parameter repertory, selecting different selection processes for each of the sections, in a way reflecting your notion of contrast and transition between sections
- In each case, the parameter repertory defines the score's *deep structure*, while the selection processes define how the deep structure is changing over time
- Further differentiation occurs through use of durations (horizontalization) specifying linear time flow (surface structure); --some programs allow for playing scores backwards, a great way to extend systematicity

Designing Single Sections

- Begin by designing single sections to get a ‘feel’ for how the PR1 output relates to your input
- Make gradual changes to the input in order to ‘move away’ from the intial design in a way you understand
- When you have a sense of how your input determines your output, you are ready to define two or more sections
- Throughout the form you are designing, focus on the relationship between system processes, keeping the parameter repertory stable for better insight into what auditory changes occur

Designing Entire Movements

- To design entire movements composed of sections, you need some overriding parametrical idea (why otherwise use algorithmic composition?)
- You have to imagine, e.g., what system process P1 will output in terms of tone color (‘instrument’), harmony (chordal structure), time delay, and dynamics, and tempo
- You can also experiment with system process P4 which computes a compromise between maximal change and stability
- Use the seven system processes to compose both transitions between sections and contrasts
- As Gertrude Stein said, “it is composition, and only composition, that makes everything different”

Designing a Piece Top Down

- The need to develop systemic parametrical ideas is most pronounced when using PR1 to compose an entire piece ‘top down’
- Carefully think about how the notion of ‘degree of change’ can help you structure an entire piece of up to 14 sections or movements
- For instance, you might want to unify a movement in terms of certain parameters like ‘harmony’ (intervallic structure)
- In this case, you would use several independent design screens, since each screen adheres to a specific harmony and chord size selection
- **Only your musical ideas can tell you how to use PR1 properly and creatively!**

Interpreting PR1 Tables

- PR1 output is referred to as a 'score table,' and can be made use of in many different ways, depending on the composer's intentions and esthetics
- PR1 output is numerical, and is meant to be <u>interpreted</u> by the composer (interpretive composition)
- There are two ways of interpreting score tables:
  - manually, "note by note," for voices and instruments, as in traditional composition, with 'inner hearing' engaged
  - auditorily, by 'orchestrating' output by way of electroacoustic instruments designed in Csound, Kyma, and Midi
- In manual interpretation, the composer proceeds from his/her insight into the function of a particular section within a larger compositional design

Output Examples

Score Table #1, Fractional

PROJECT 1 - - Score Table
Name: form1d-4i
Comment: example for 4 instruments/voices
with accumulation without horizontalization
Branching Table
Instrument: 7 Entry Delay: 2 Pitch: 3 Register: 4 Dynamics: 5

Section 1

INSTR		RHYTHM				HARMONY	SEQ	REGISTER	DYNAMICS
		TEMPO	BEGIN	END	FERM				
1	* 1	40	* 3/8			* G# C# A#	321	S 333	* f
2	1		0/0			D#	1	2	f
3	1		0/0			C	1	5	f
4	1		3/5			* F A# G	123	444	f
5	1		3/4			F# D# G#	123	555	* ff
6	1		5/8			* A D B E	1423	B 3245	ff
7	1		5/8		3	C#	1	3	ff
8	* 4		1/1			* D A C G# D#	12543	24352	ff
9	4		4/5			F#	1	4	* pp
10	4		1/3			* D#	1	4	pp
11	4		1/3			G# F	12	S 23	pp
12	4		0/0			A	1	5	pp
13	4		1/3			D	1	4	pp
14	4		2/5			B	1	5	* p
15	* 2		* 5/8			* C# G# B F#	1432	3234	p
16	2		5/8			* F# C# E	321	254	p
17	2		2/3			C G A#	231	532	p
18	2		4/5			* C G F F# C# B	654213	543252	* mf

Using the Score Table, #1

- The main columns are ‘instrument,’ ‘rhythm,’ ‘harmony,’ ‘sequence,’ ‘register,’ and ‘dynamics’, with degrees of change P4, P2, P3, P4, P5*
- Of these, ‘rhythm’ refers to entry delay, indicated in fractions; for example, ‘1/1’ here stands for ‘quarter note [or half note] at tempo 40,’ and ‘5/8’ stands for ‘5 thirty-second notes’ [with 1/1 equalling a quarter note], or 5/8 of a 1/1 unit taking up 0.625 of a 1/1 unit
- The composer decides about time signature, and ‘bar lines’ or ‘measures into which to fit the entry delay series (column)
- We use ‘table lookup’** to string together entry delays to form meters and bars, resulting in ‘rhythm’ in the conventional sense
- Durations are free. That is, the composer decides whether a duration for the first entry delay (5/8) is going to be shorter, longer, or equal to 5/8
- In this way, the composer defines texture, harmony, density, even gesture

*‘Sequence’ is an optional parameter for sequencing single tones; ** table lookup is automated in metrical presentation, see slides 59-60

Using the Score Table, #2

- Rhythm decisions are made based on context
- In the example, context is characterized as follows:
  - dynamic fields (P5)
  - tone color (P7) -- homogeneous tone color fields
  - register [tone height] alternating between ‘stable’ and ‘quick changing (P4); this means ‘mostly melic,’ forming short motives at various levels of tone height
  - ‘rhythm’ (entry delay) is the most quick-changing parameter: diffuse, complex meter (P2) which can, however, be simplified by using a low differentiation metric representation (slides 59-60)
  - ‘pitch’ (harmony), here based on intervals 5-9-7-10 (semitones), is relatively quick-changing (P3), emphasizing ‘fourth’ (5 semitones) and ‘fifth’ (7 semitones)
- This suggests we use tone color and dynamic fields to create unity and stability
- There are 5 registers; we interpret ‘register’ according to the instruments used; for each instrument, ‘5’ could mean something different; it could even differentiate an instrument group into single instruments (see slide 26)
- This output may be too complex for a beginner, who would start with a single instrument, and relatively low degrees of change (P5-7); however, using metrical representation (slides 59-60) simplifies the transcription task
- **In all cases, the score table challenges our musical imagination, AND THAT IS EXACTLY WHAT IT IS MEANT TO DO!**

Score Table #1, Metrical (a)

PROJECT 1 - - Score Table
Name: form1d-4i
Comment: example for 4 instruments/voices
with accumulation, without horizontalization
Branching Table
Instrument: 7 Entry Delay: 2 Pitch: 3 Register: 4 Dynamics: 5

Section 1

INSTR	RHYTHM	TEMPO	BEGIN	END	FERM	HARMONY	SEQ	REGISTER	DYNAMICS
1	* 1	40	* 1:81	1:84		* G# C# A#	321	S 333	* f
2	1		1:84	1:84		D#	1	2	f
3	1		1:84	1:84		C	1	5	f
4	1		1:84	2:11		* F A# G	123	444	f
5	1		2:11	2:87		F# D# G#	123	555	* ff
6	1		2:87	3:84		* A D B E	1423	B 3245	ff
7	1		3:84	4:11	3	C#	1	3	ff
8	* 4		4:11	5:11		* D A C G# D#	12543	24352	ff
9	4		5:11	5:87		F#	1	4	* pp
10	4		5:87	6:82		* D#	1	4	pp
11	4		6:82	6:85		G# F	12	S 23	pp
12	4		6:85	6:85		A	1	5	pp
13	4		6:85	6:87		D	1	4	pp
14	4		6:87	7:82		B	1	5	* p
15	* 2		* 7:82	7:87		* C# G# B F#	1432	3234	p
16	2		7:87	8:84		* F# C# E	321	254	p
17	2		8:84	9:82		C G A#	231	532	p
18	2		9:82	9:88		* C G F F# C# B	654213	543252	* mf

Under 'Begin,' entry delays are expressed in metrical terms, specifying subdivisions of a unit measure, for 9 'beats' (no triplets); for example, '1:84' indicates that the tone begins on the fourth 32nd note of a unit of 8 making up a quarternote

Score Table #1, Metrical (b)

PROJECT 1 - - Score Table
Name: form1d-4i
Comment: example for 4 instruments/voices
with accumulation without horizontalization
Branching Table
Instrument: 7 Entry Delay: 2 Pitch: 3 Register: 4 Dynamics: 5

Section 1

INSTR		RHYTHM				HARMONY	SEQ	REGISTER	DYNAMICS
		TEMPO	BEGIN	END	FERM				
1	* 1	40	* 1:81	1:84		* G# C# A#	321	S 333	* f
2	1		1:84	1:84		D#	1	2	f
3	1		1:84	1:84		C	1	5	f
4	1		1:84	2:11		* F A# G	123	444	f
5	1		2:11	2:87		F# D# G#	123	555	* ff
6	1		2:87	3:32		* A D B E	1423	B 3245	ff
7	1		3:32	4:11	3	C#	1	3	ff
8	* 4		4:11	5:11		* D A C G# D#	12543	24352	ff
9	4		5:11	5:87		F#	1	4	* pp
10	4		5:87	6:82		* D#	1	4	pp
11	4		6:82	6:85		G# F	12	S 23	pp
12	4		6:85	6:85		A	1	5	pp
13	4		6:85	6:87		D	1	4	pp
14	4		6:87	7:82		B	1	5	* p
15	* 2		* 7:82	7:87		* C# G# B F#	1432	3234	p
16	2		7:87	8:84		* F# C# E	321	254	p
17	2		8:84	9:82		C G A#	231	532	p
18	2		9:82	9:88		* C G F F# C# B	654213	543252	* mf

Under 'Begin,' entry delays are expressed in metrical terms, now including triplets; for example, '3:32' indicates that the note begins on the second eighth note under a triplet differentiating the third quarternote

Score Table #1, Decimal

PROJECT 1 - - Score Table

Name: form1d-4i
Comment: example for 4 instruments/voices
with accumulation without horizontalization
Branching Table
Instrument: 7 Entry Delay: 2 Pitch: 3 Register: 4 Dynamics: 5

Section 1

INSTR		RHYTHM				HARMONY	SEQ	REGISTER	DYNAMICS
		TEMPO	BEGIN	END	FERM				
1	* 1	40	* 0	.37		* G# C# A#	321	S 333	* f
2	1		.37	.37		D#	1	2	f
3	1		.37	.37		C	1	5	f
4	1		.37	.97		* F A# G	123	444	f
5	1		.97	1.72		F# D# G#	123	555	* ff
6	1		1.72	2.35		* A D B E	1423	B 3245	ff
7	1		2.35	2.97	3	C#	1	3	ff
8	* 4		2.97	3.97		* D A C G# D#	12543	24352	ff
9	4		3.97	4.77		F#	1	4	* pp
10	4		4.77	5.10		* D#	1	4	pp
11	4		5.10	5.44		G# F	12	S 23	pp
12	4		5.44	5.44		A	1	5	pp
13	4		5.44	5.77		D	1	4	pp
14	4		5.77	6.17		B	1	5	* p
15	* 2		* 6.17	6.80		* C# G# B F#	1432	3234	p
16	2		6.80	7.42		* F# C# E	321	254	p
17	2		7.42	8.09		C G A#	231	532	p
18	2		8.09	8.89		* C G F F# C# B	654213	543252	* mf

Under 'Begin' and 'End, entry delays are shown in decimal form; at tempo MM=60 this would correspond to seconds

Midi Example, ASCII

```
form1d-4i; 1
 5
 1   206
 1             0             562           44            4
 1             0             562           37            4
 1             0             562           46            4
 1             562           0             27            4
 1             562           0             60            4
 1             562           900           53            4
 1             562           900           58            4
 1             562           900           55            4
 1             1462          1125          66            5
 1             1462          1125          63            5
 1             1462          1125          68            5
 1             2587          937           45            5
 1             2587          937           26            5
 1             2587          937           59            5
 1             2587          937           64            5
 1             3525          937           37            5
 4             4462          1500          26            5
 4             4462          1500          57            5
 4             4462          1500          36            5
 4             4462          1500          68            5
 4             4462          1500          27            5
 4             5962          1200          54            1
 4             7162          500           51            1
 4             7662          500           32            1
 4             7662          500           41            1
 4             8162          0             69            1
 4             8162          500           50            1
 4             8662          600           71            2
 2             9262          937           37            2
```

Midi output occurs in two forms: ASCII and binary; the latter can be used to control a Midi instrument

Csound Example

```
;form1d-4iCsound
i1    0      0.93   2000   246.94   -0.28
i1    0      0.93   2000   184.99   -6.12
i1    0      0.93   2000   220      -0.10
i2    0.93   0.5    2000   349.22   6.12
i4    1.43   0.56   2000   32.70    -0.20
i4    1.43   0.56   2000   77.78    0.71
i4    2      1.12   2000   220      1
i4    2      1.12   2000   293.66   -1
i4    3.12   0.18   2000   61.73    0.75
i4    3.12   0.18   2000   41.20    0.97
i3    3.31   0.18   32000  69.29    -0.59
i3    3.31   0.18   32000  92.49    0.61
i4    3.5    0.93   32000  369.99   -0.10
i4    3.5    0.93   32000  277.18   0.30
i4    3.5    0.93   32000  493.88   -0.44
i4    3.5    0.93   32000  329.62   0.32
i2    4.43   0.18   32000  36.70    0.89
i3    4.62   1.2    32000  130.81   -0.57
i3    4.62   1.2    32000  174.61   0.32
i3    5.82   1.5    32000  97.99    0
i1    7.32   0.37   32000  293.66   0.30
i2    7.7    0.37   4000   164.81   0.28
i1    8.07   0.5    4000   73.41    -0.89
i1    8.07   0.5    4000   48.99    0.16
i4    8.57   0.5    4000   82.40    0.77
i4    8.57   0.5    4000   207.65   0.48
i3    9.07   1.2    4000   277.18   0.14
i4    10.27  1.5    4000   51.91    0.69
i4    10.27  1.5    4000   34.64    0.75
i4    10.27  1.5    4000   58.27    0.38
i3    11.77  0.18   4000   146.83   -0.55
i3    11.77  0.18   4000   195.99   0.20
i1    11.96  0      8000   329.62   0.55
```

p1=instr, p2=start, p3=dur; p4=amp, p5=freq, p6=location, fixed 6 p-field format for an instrument using 'soundin'

Kyma Example

;form1d-4ikyma

i1	0	0.93	246.94	2000	-0.28
i1	0	0.93	184.99	2000	-6.12
i1	0	0.93	220	2000	-0.10
i2	0.93	0.5	349.22	2000	6.12
i4	1.43	0.56	32.70	2000	-0.20
i4	1.43	0.56	77.78	2000	0.71
i4	2	1.12	220	2000	1
i4	2	1.12	293.66	2000	-1
i4	3.12	0.18	61.73	2000	0.75
i4	3.12	0.18	41.20	2000	0.97
i3	3.31	0.18	69.29	32000	-0.59
i3	3.31	0.18	92.49	32000	0.61
i4	3.5	0.93	369.99	32000	-0.10
i4	3.5	0.93	277.18	32000	0.30
i4	3.5	0.93	493.88	32000	-0.44
i4	3.5	0.93	329.62	32000	0.32
i2	4.43	0.18	36.70	32000	0.89
i3	4.62	1.2	130.81	32000	-0.57
i3	4.62	1.2	174.61	32000	0.32
i3	5.82	1.5	97.99	32000	0
i1	7.32	0.37	293.66	32000	0.30
i2	7.7	0.37	164.81	4000	0.28
i1	8.07	0.5	73.41	4000	-0.89
i1	8.07	0.5	48.99	4000	0.16
i4	8.57	0.5	82.40	4000	0.77
i4	8.57	0.5	207.65	4000	0.48
i3	9.07	1.2	277.18	4000	0.14
i4	10.27	1.5	51.91	4000	0.69
i4	10.27	1.5	34.64	4000	0.75
i4	10.27	1.5	58.27	4000	0.38

p1=instr, p2=start, p3=dur; p4=freq, p5=amp, p6=location
4-voice instrument based on the ‘sample’ icon

References

[On reserve in library]

Koenig, G.M. (1999). Project One revisited: On the analysis and interpretation of PR1 tables. In J. Tabor (Ed.), Otto Laske: Navigating new musical horizons. Westport, CT: Greenwood Press, ch. 3

Laske, O. (2002). Score manipulation as a tool for compositional and sonic design. In T. Licata (Ed.), Electroacoustic Music. Westport, CT: Greenwood, ch. 6

Laske, O. (1999). Furies and Voices: Compositional theoretical observations. In J. Tabor (see above), ch. 9

Laske, O. (1990). The composer as the artist's alter ego. Leonardo 23(1):53-66.

Laske, O. (1989). Composition theory in Koenig's Project 1 and Project 2. In C. Roads (Ed.) The Music Maschine. Cambridge, MA: The MIT Press, 119-130.

Annelise Laske Studio

Otto E. Laske PhD M.Mus. M.Ed

Specialist in Algorithmic Composition

for Instruments, Voices, and Electroacoustic Media

51 Mystic Street
West Medford, MA 02155, U.S.A.
(781) 391-2361, olaske@cdremsite.com
www.emf.org/subscribers/laske/index.html
www.cdemusic.org/artists/laske.html
www. cdemusic.org/store/cde_search

What gets computed gets reflected upon

- Consultation on Algorithmic Composition
- Design of Compositional Multi-Media Environments