



SERIALISM, FRACTALS, GRAMMARS AND GRIDS

Algorithmic Composition



Serialism

- Arnold Schoenberg and *Serialism*
- Chromatic scale – 12 notes/octave with equal ratios between (half)steps
- Pitch – an element of the chromatic scale
- Pitch class – pitch mod 12, e.g. “C-sharp” without regard to octave
- Tone row – permutation of the 12 pitch classes
- Music based on tone rows can be *atonal*

Melodic/Tone Row Transformation

- Original: $p[i]$
- Transposition: $T(p[i],c) = (p[i] + c) \bmod 12$
- Inversion: $I(p[i]) = (-p[i]) \bmod 12$
- Retrograde: $R(p[i]) = p[12 - i]$
- Also: $(p[i]*5) \bmod 12 = I(p[i]*7)$

Why serialism?

- In general, listeners cannot hear retrograde and/or inversion relationships
- Intervals are preserved
- Tone “row-ness” is preserved
- “Denial of tonality” produces new textures

Fractals and Nature

- Melodic contours are often fractal-like
- Composers often use fractal curves to generate music data
- Examples:
 - Austin, Canadian Coastline
 - Cage, Atlas Eclipticalis

Grammars

melody ::= intro middle ending
 middle ::= phrase | middle phrase
 phrase ::= sequence-a | sequence-b

```
function melody()
  return seq(intro(), middle(), ending())
function middle() return #?(random() < 0.5,
  phrase(), seq(middle(), phrase()))
function phrase() return #?(random() < 0.3,
  sequence-a(), sequence-b())
```

Pitch and Rhythm Grids

- Quantize random numbers to scales, grids


```

define function pitch-filter(p, f)
  return #?(member(p % 12, f),
            p, pitch-filter(p + 1, f))

define function grid-scale(grid-function)
  return score-gen(score-len: 50,
                  ioi: 0.15,
                  pitch: funcall(grid-function, 60 + random(12)),
                  name: quote(pluck-kwp))

define function c-major(p)
  return pitch-filter(p, {0 2 4 5 7 9 11})

```



Quantizing to Rhythmic Grid

```

function on-beat(time, beat-len)
  begin with beats = round(time / float(beat-len))
  return beats * beat-len
end

define function grid-rhythm(grid-function)
  return score-gen(score-len: 100,
                  time: on-beat(real-random(0, 15), 0.15),
                  pitch: funcall(grid-function, 60 + random(12)),
                  dur: 1.0,
                  name: quote(pluck-kwp))

exec score-play(grid-rhythm(quote(c-major)))

```

