

Music Automaton

Abstract

This project tries to answer the question, if a piece of music can be considered as a result of a so called grammar.

Furthermore the thesis is, that a piece of music, generated on a given grammar, always sounds kind of 'plausible'.

In general it's said that a piece 'speaks his own language', which in this case would be the language, induced by the grammar.

Following here the hierarchy of Noam Chomsky, the language would be estimated as a 'type-0 language'. (The restraint in the productions $|w_1| \leq |w_2|$ does not exist, and there are no limits for the productions)

Given the grammar, there are constructed entities, called words.

Remember: Each word constructed by the same grammar is part of the same language.

So the automaton will produce a certain amount of words, which will simply be concatenated. That means that the piece of music is considered as a sequence of words.

(On the linguistic level, this would be the phrase).

To keep the things clear, the output is a piano score, which means the automaton is limited to the 3 (more or less) scalable parameters: Time, Pitch, Velocity.

The formal grammar

The formal grammar is defined as: $G=(V, \Sigma, P, S)$.

i)

Variables (V): There are no Variables except S. $V=\{S\}$

ii)

The Alphabet (Σ) is defined as $\Sigma=\{T \times P \times V\}$

It's the cartesian product of the sets T,P,V.

T contains the time-possibilities.

P contains the pitch-possibilities.

V contains the velocity-possibilities.

Where $|\Sigma^*|$ is infinite, only due the T-set. P is limited here. It's maximum range is 0 - 127 (kind of \mathbb{N} . - it would be infinite, if any micro-intervall would be allowed (kind of \mathbb{R}) . V is limited.

Its range is 0 - 6, which stands for *ppp, pp, p, mp - mf, f, ff, fff* . , would be infinite if any dynamik transition with infinite small steps like *crescendo* would be allowed.

The *terminal symbols* are defined as *notecoordinates*,

where a notecoordinate is a vector of 3 entries $\begin{pmatrix} t \\ p \\ v \end{pmatrix}, t \in T, p \in P, v \in V.$

t is the time - component,

p is the pitch - component
v is the velocity (dynamics) - component.

iii)

The Productions (P)

It occurs the problem, that these 3 components can not be treated exactly the same way.

Respectively they are dependent of each other.

So it's for example possible to 'repeat' a note, which means to duplicate the time-value. The current word would then contain 2 notes, with same duration, pitch, velocity, just displaced in the time-grid.

As a parallel operation you could 'repeat' a note in pitch, what would be a note, same duration, velocity, but displaced in the tonal-grid (tonal-grid means any predefined scale). Thus it forms something, which is in musical terms an interval.

But on the contrary it's not possible to 'repeat' a note in velocity, because each note maps only 1 dynamic.

The set of production rules in a Chomsky-like grammar is a set of allowed replacements like

$a \rightarrow ab$, $ab \rightarrow c$... etc.

In case of the music automaton the production rules are defined as functions

$f : \Sigma \rightarrow \Sigma$.

This is allowed accepting the claim that any function whose X (domain) and Y (codomain) are the same set, can be explained by a finite sequence of replacements.

TODO ? : Mathematical proof.

Example:

Let be $P = \{(1) a \rightarrow ab, (2) b \rightarrow abc\}$.

Applying (1) then (2) produces $a \rightarrow ab$ (1) $\rightarrow aabc$.

Can be done by a function:

```
f(x) {  
  return duplicate x + next 2 subsequent elements.  
}
```

This could be even more general, by giving a parameter to f.

Lets say: param is the number of repeats and the number of appended elements.

```
f(x, param) {  
  return param times x + next param subsequent elements.  
}
```

gives with $X = a$ and $param = 5$: aaaaabcdef.

Functions

To proceed the equivation $X \rightarrow Y$, (X, Y sets of notecoordinates), there may be applied a *clone*- function to double the value of the lefthand side.

Musicautomaton has the following elementary functions:

They are *shifting* one of the 3 parameters of notecoordinate.

Pt1 changes the starting time of notecoordinate(s).

Pt2 changes the end time of notecoordinate(s).

Pp changes the pitch of notecordinate(s).

Pv changes the dynamics of notecordinate(s).

There are 2 'Time'-functions, as the time-component of a notecordinate is a tuple (start, end).
The functions are working on (predefined) grids.

The T-grid is a sequence of possible t-positions. The
For example placing to half notes in a grid [0 to 16] ,
grid width = 16th, is :
note1.t_start = 0 , note1.t_end = 8;
note2.t_start = 8 , note2.t_end = 16;

The P-grid is a subdivision of the midi-range (0 - 127) by any scale,
defined as a sequence of intervalls.

Example: range = (36, 72) scale = [2,2,1,2,2,2,2] wold give a melodic c-minor scale from C3 to C5.

The V-grid was decribed above.

All operations on this grid are done in modulo. Wich means that the automaton is working in
 $\mathbb{Z} \text{ sizeOfGrid}$.

Based on the elementary functions, its possible to construct kind of macros, wich means:
Elementary functions are applied in any combination.

Examples:

Mt_repeat repeats notecordinate(s) in time
would call: *clone*, then on the clone *Pt2* and *Pt1* with the same Parameter.
Mp_add 'repeats' a pitch of notecordinate(s). (makes intervall / chord)
would call: *clone*, then on the clone *Pp*.

As described above, there is no repeat-function for the dynamics.

iv)

The starting element:

$S = \{ \text{notecordinate} \in \{T, P, V\}, \neq \emptyset, \text{finite} \}$

is a set of notecordinates containing at least one element.

Setting the Parameters for the automaton, is the same as defining a grammar, for the music piece.

Defining Σ by defining T-,P,V- grids.

Defining P by writing any combination of *Productionfunctions*

Defining S by any method. Only constraint: $S \in \Sigma$

(V is only {S}).

Further work:

Is the automaton universal ? Is he able to produce any piece of music (representable on piano) ? To do so: Is it possible to provide a kind of script for each possible piece of music ?

Fredrik Zeller, Tü. January 2022