Roberto Noel
Enric Junque de Fortuny
Machine Learning
10/05/2019

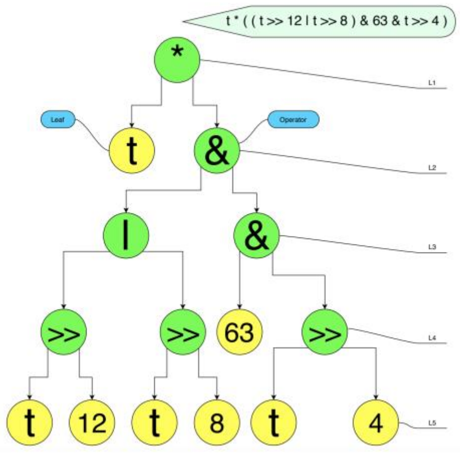
PROJECT PROPOSAL

On September 6, 2011, a counter complex user, “Viznut”, released a video showing how short, on-line C programs can generate complex sounds. Since then, the subject has gone viral, with YouTube channel “Computerphile” uploading a video about it, and hundreds of others attempting to compose their own songs. One quite iconic composition is the 42 melody, which was separately discovered by several people: “t*(42&t>>10)”. This style of music generation is now dubbed Bytebeat. Recently, many deterministic adaptations of the style have appeared (take this “Rick Roll” for example). Though these are interesting, the logic behind them is quite trivial. The least understood versions of these expressions, have a constrained set of operators, excluding conditionals, trig functions, and lists.

My objective with this project is to leverage the power of machine learning to generate new, optimized versions of these expressions while maintaining the constraints of my inputs. To do this, I will design a genetic algorithm which will pick its “winning” children through Music Information Retrieval (MIR). There are many MIR techniques which include onset detection, melody extraction, etc. I will use a combination of these techniques to create a musicality metric which will ultimately determine the evolution of the expressions.

Before creating this metric however, I must come up with a way to mutate parent expressions into the children which will be evaluated. To do this, it is useful to use a data structure, which will allow for decomposition and re-composition of the parent. For this, I chose a binary-tree called an expression tree (see fig.1), in which the internal nodes are operators and the leaves are variables. The in-order traversal of the tree should return the original expression.

The tree representation of the expression acts as a sort of “musical DNA”, and opens up a world of possibilities for mutation. I will have to experiment with different strategies so as to prevent over or under mutation and make sure that every kind of mutation is possible (eg. change of operator, change of variable, or change of parentheses). It may also be possible for expressions to “mate” if they share similar MIR traits.


t*((t>>12|t>>8)&63&t>>4)
*
L1
Leaf
Operator
t
&
L2
|
&
L3
>>
>>
63
>>
L4
t
12
t
8
t
4
L5