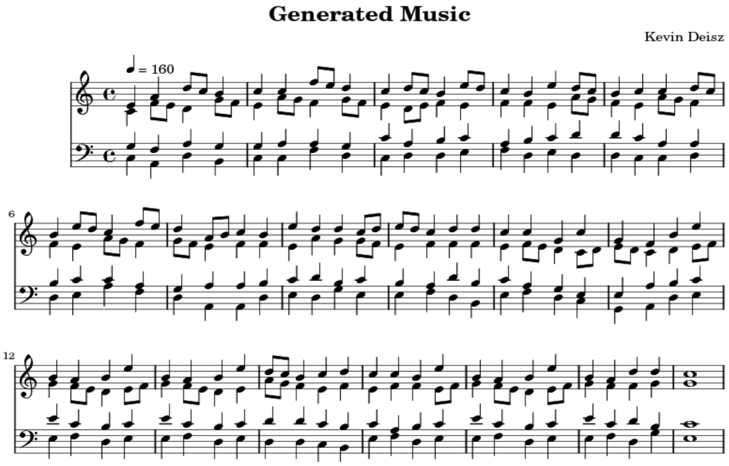
Computer Systems Lab Research 2008-2009
Automated Musical Part Writing
Kevin Deisz

This project will generate its own music based off of simple part writing rules, i.e. parallel fifths, parallel octaves, spacing errors and voice crossing errors. The code employs basic recursive constraint satisfaction problem code found in many similar problems such as n-queens or solving sudoku.



```
current_chord = [] def constraint_func( A, a, B, b ): #
--- voice crossing, distancing, parallel 5ths and 8ths
--- # global current_chord if ( (ab and A>B) ): if
abs(A-B)==1 and ( abs(a-b)>7 ): return False if A!=0
and B!=0 and ( abs(current_chord[A]-
current_chord[B])==15 and abs(a-b)==15 ): return
False if A!=0 and B!=0 and ( abs(current_chord[A]-
current_chord[B])==7 and abs(a-b)==7 ): return False
if ( abs(current_chord[A]-current_chord[B])==4 and
abs(a-b)==4 ): return False return True return False
```

The above method is the only constraint function given to the CSP code, after the domains are set and the assignments map created. The domains are created off of only the structured major progressions, and only can be a root, third or fifth of the given chord. The eighth notes are generated passing tones.