A proposal for automatic harmony analysis with Minimalist syntax

Sean P. Anderson, University of Michigan
ICCC ’22 Doctoral Consortium

today

- motivation
- symbolic model
- problem: fitting to Bach chorales
- questions

exploring overlap in computational systems for music and language

GTTM (Lerdahl & Jackendoff, 1983): music does not have a syntax

exploring overlap in computational systems for music and language

GTTM (Lerdahl & Jackendoff, 1983): music does not have a syntax

…but generative linguistics has progressed since then:

Minimalist Program, Chomsky (2014)

-> “Identity Thesis”: Katz & Pesetsky (2011), Mukherji (2014)

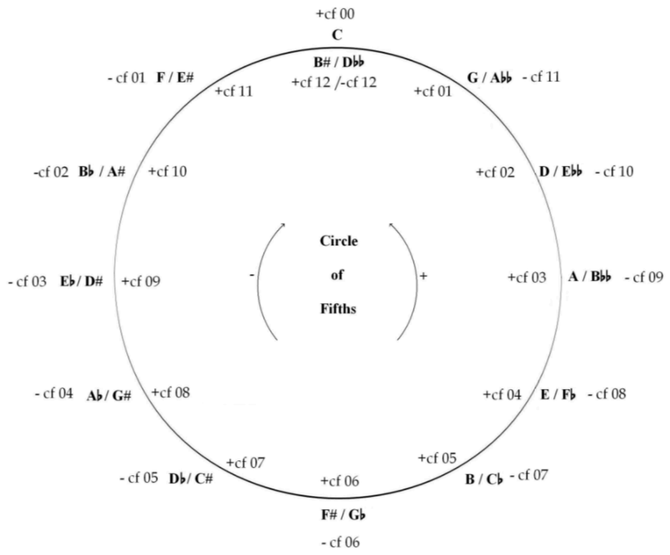
towards empirical investigation of “identity thesis” theories of harmony

- do they have merit?
- better interpretable models of harmony

modeling harmony with Minimalist syntax

given: bag of chords,
each with features

Example 1.2-28. Major/minor triadic relationships represented as "Circle of Fifths" features



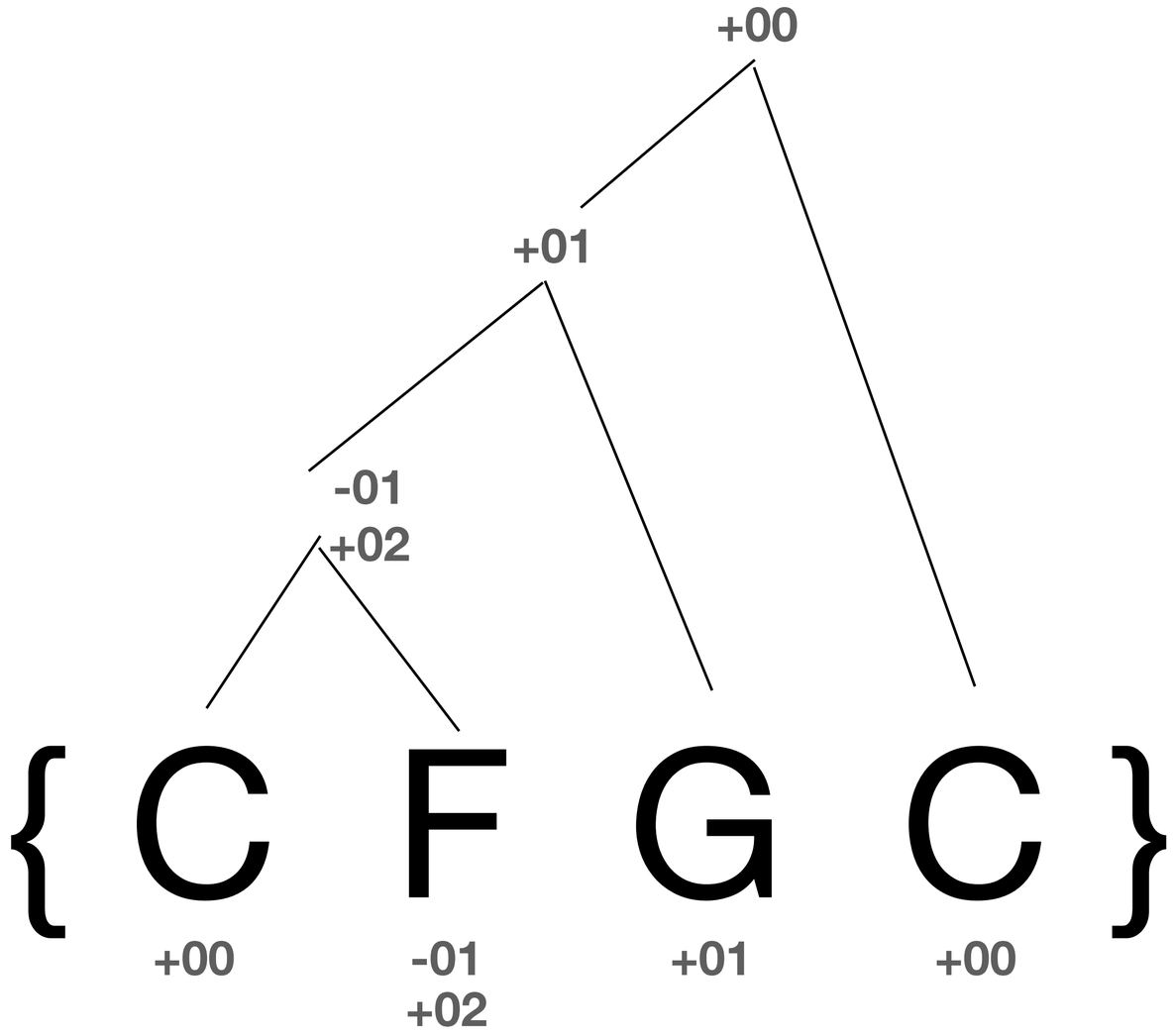
Mukherji (2014)

{ C	F	G	C }
+00	-01 +02	+01	+00

modeling harmony with Minimalist syntax

given: bag of chords,
each with features

If features “Agree”, then
two subtrees are <u>Merged</u>

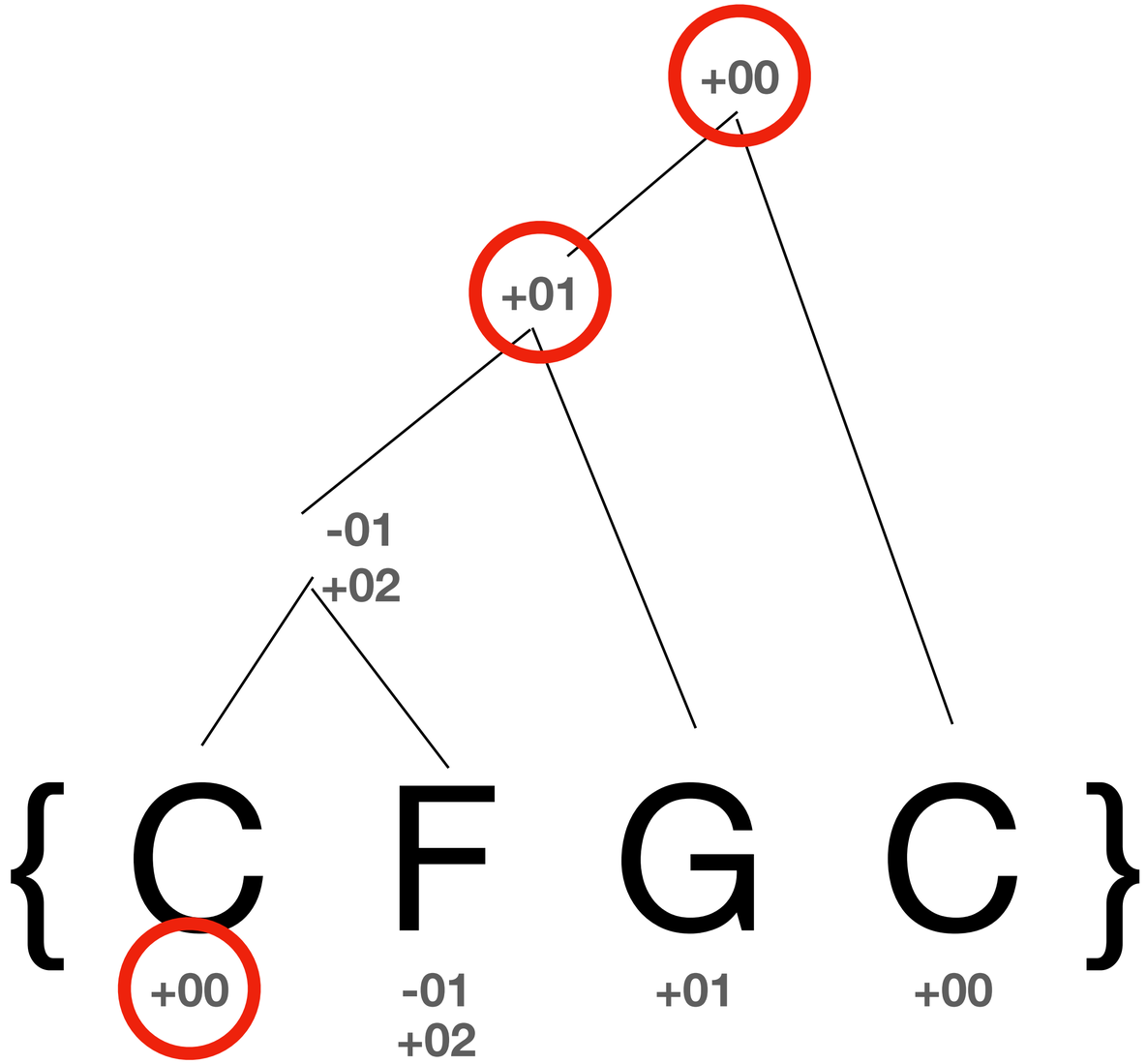


modeling harmony with Minimalist syntax

given: bag of chords,
each with features

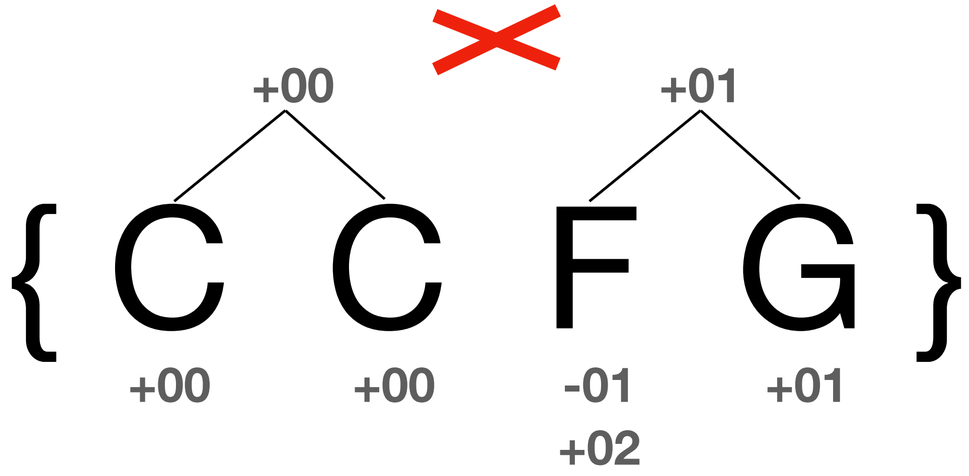
If features “Agree”, then
two subtrees are Merged

check features to
determine success



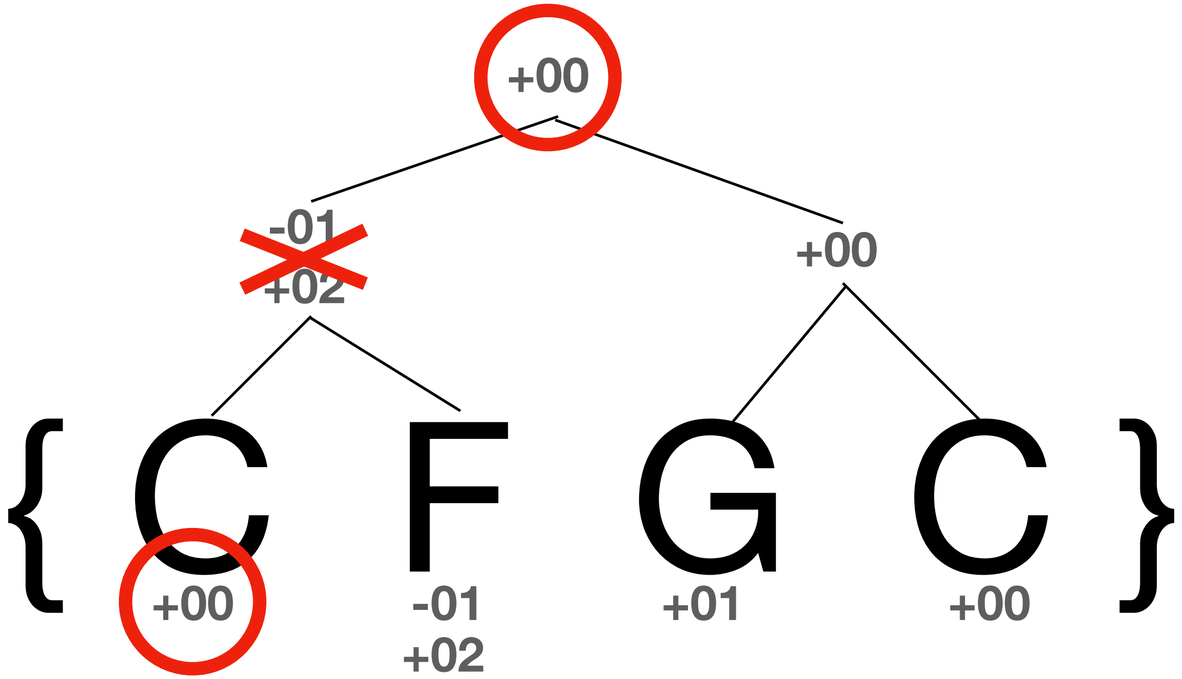
modeling harmony with Minimalist syntax

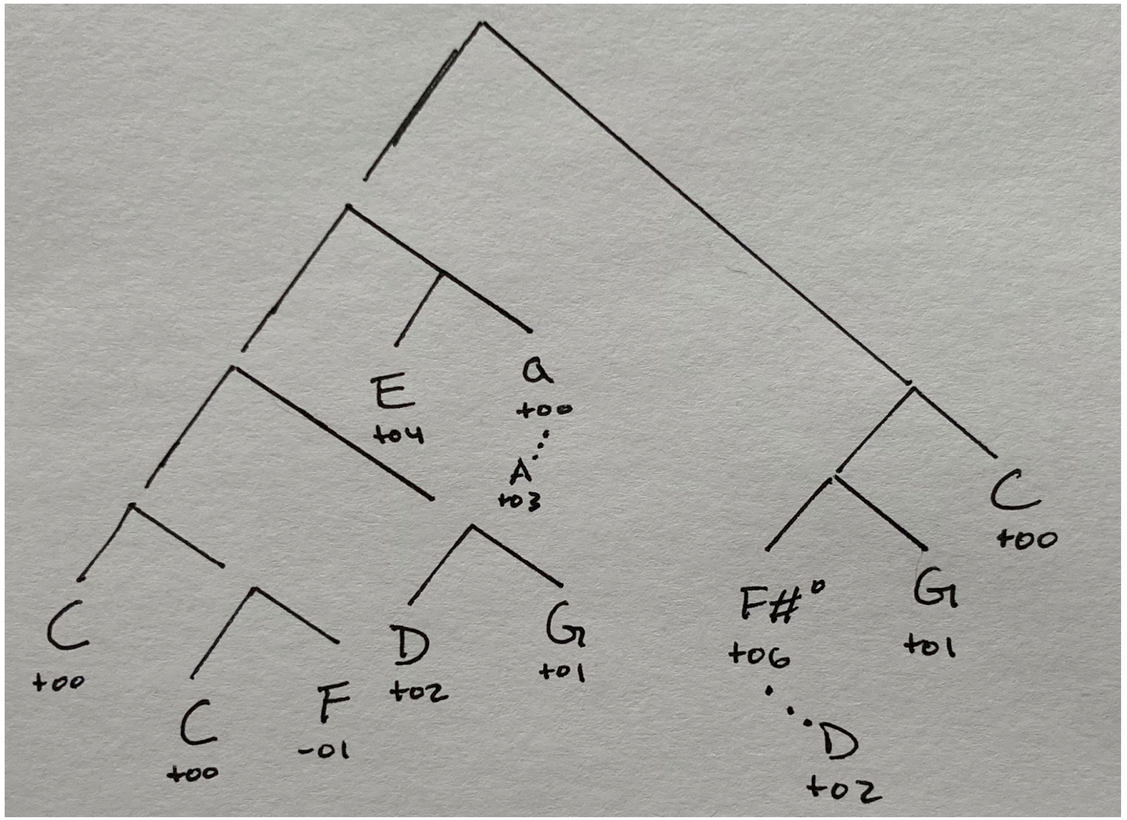
given: bag of chords,
each with features

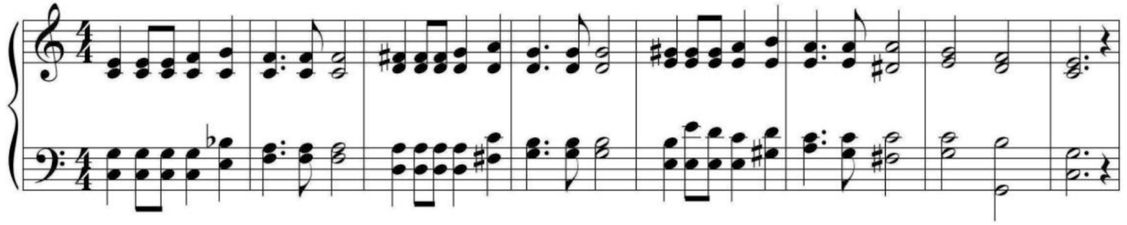


If features “Agree”, then
two subtrees are Merged

these derivations
would be thrown out







Bortniansky's *Tebe Poem*

hand-designed feature checking

SO2. Agree(SO1, SO2) = True if and only if:

1) Labels(SO1)[*c5*] - Labels(SO2)[*c5*] = 1 or 0, OR

2) Labels(SO1)[*c3*] - Labels(SO2)[*c5*] = 1 or 0, OR

3) Labels(SO2)[*type*] = "minor", SO2 is a *Stufe*, and Labels(SO1)[*c5*] - Labels(SO2)[*c3*] = 1 or 0.

Agree(SO1, SO2) = False otherwise.

Anderson (2020)
Mukherji (2014)

goal: **learned feature checking**

{ C G D G C F a Bº C... }

{ a E a d E C G a d E... }

...N ~= 300

$\phi_{i,j}$: Labels(SO1)[feature] - Labels(SO2)[feature] ~ k

feature ∈ {c5, c3, …}

~ ∈ {=, ≠}

k ∈ {-1, 0, 1}

$$(\phi_{1,1} \wedge \phi_{1,2} \wedge \ldots) \vee (\phi_{2,1} \wedge \phi_{2,2} \wedge \ldots) \vee \ldots$$

program synthesis: handful of I/O examples -> best fitting symbolic function

One challenge: verification

given feature-checking function: $(\phi_{1,1} \wedge \phi_{1,2} \wedge \ldots) \vee (\phi_{2,1} \wedge \phi_{2,2} \wedge \ldots) \vee \ldots$

and input chords:

{ D B° G G a C C C F... }

How to verify model will produce desired ordering? (many failing derivations)

[C G D G C F a B° C...]