Intelligent Music Software

Robert Keller
Harvey Mudd College
keller@cs.hmc.edu

Stauffer Talk
30 June 2011

Interaction

Please interrupt the talk with questions.

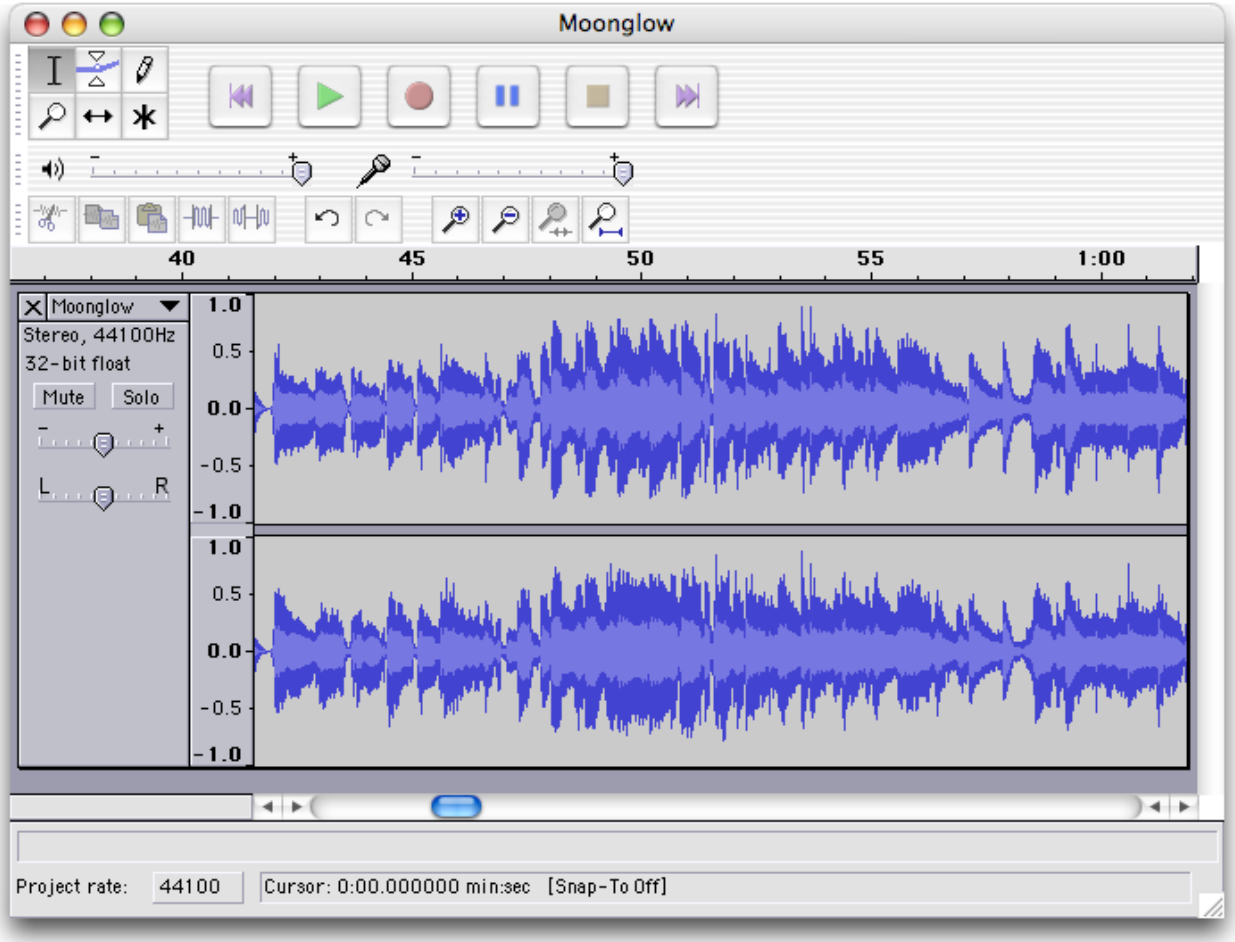
Outline

- Describing the space
  - Music software in general
  - Intelligent music software
  - Prior art
- Our project
  - Impro-Visor
  - RBM-provisor
  - Current work

Music Software Varieties

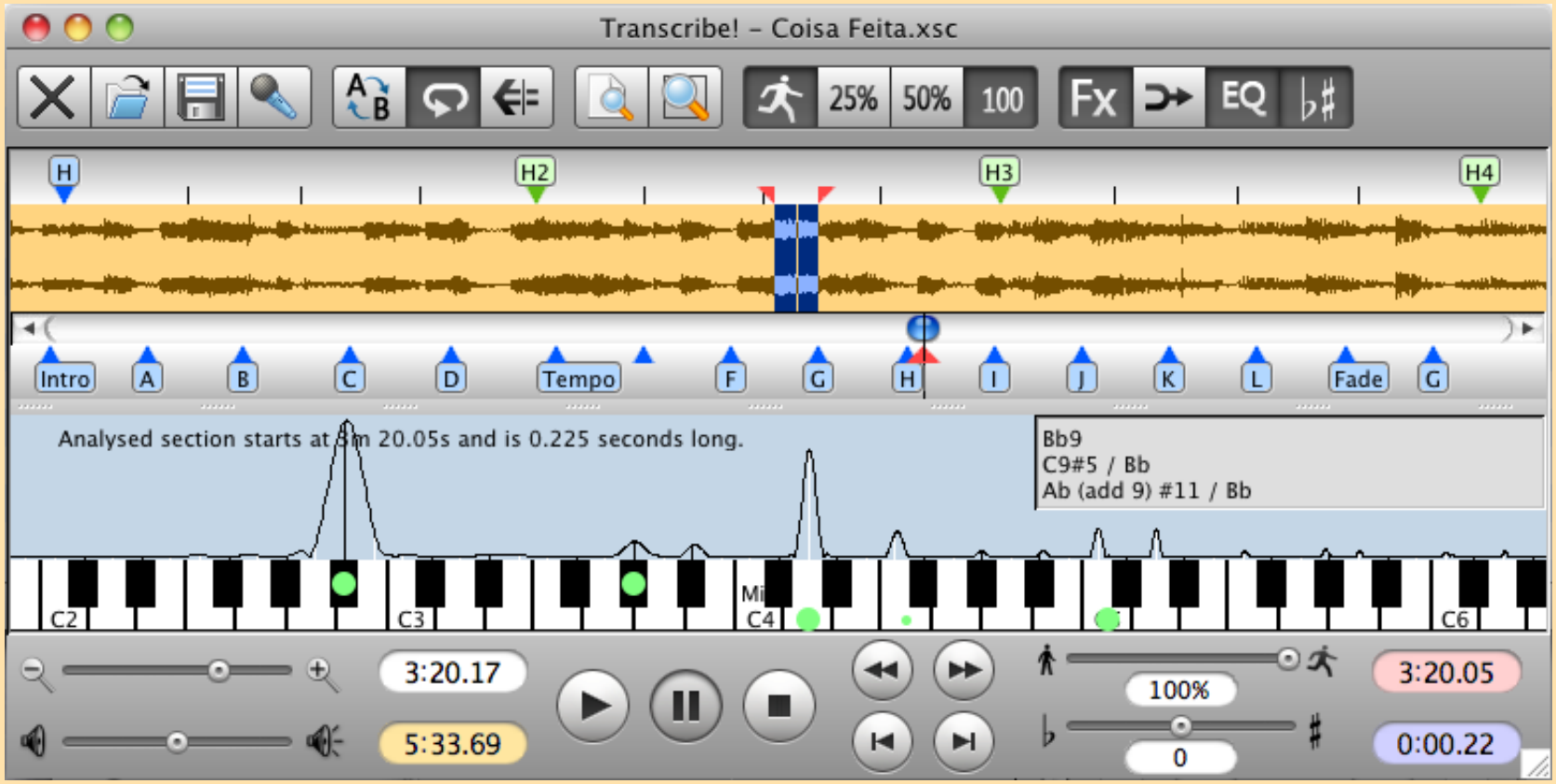
- Music organizer, searcher
- Music recommender
- Music player (mp3, wav, MIDI, …)
- Music recorder
- Music transcriber (audio to score)
- Music synthesizer (imitate instruments)
- Music generator (create music)
- Music notation editor (“scorewriter”)
- Digital audio workstation (DAW)
- Music composition assistant
- Music score follower (educational)

Example: **Audacity**
sound recorder and track editor
(Dominic Mazzoni, HMC ‘99, while at CMU)



Example: **Transcribe!**

transcription (slow-down) software analyzes audio spectra



Intelligent Music Software

Definition of “Intelligent”

Merriam-Webster on-line

1.

a: having or indicating a high or satisfactory degree of **intelligence** and mental capacity

b: revealing or reflecting good judgment or sound thought : skillful

2.

a: possessing **intelligence**

b: guided or directed by intellect : rational

3. ?

a: guided or controlled by a computer;
especially : using a built-in microprocessor for automatic operation, for processing of data, or for achieving greater versatility

b: able to produce printed material from digital signals
as in *an intelligent copier*

Definition of “Intelligence”

Merriam-Webster on-line

1.a: the ability to learn or understand or to deal with new or trying situations: reason; also: the skilled use of reason

1,b: the ability to apply knowledge to manipulate one's environment or to think abstractly as measured by objective criteria (as tests)

1.c : mental acuteness : shrewdness

2.a : an intelligent entity; especially : angel

2.b : intelligent minds or mind, as in *cosmic intelligence*

3: the act of understanding : comprehension

4. a : information, news

4.b : information concerning an enemy or possible enemy or an area; also : an agency engaged in obtaining such information

5: the ability to perform computer functions

wikipedia

- *Intelligence* derives from the Latin verb *intelligere* which derives from interlegere meaning

  to "pick out" or discern.

- In other words,

  the ability to make decisions.

Intelligence

- We will assert that

Intelligent Music Software

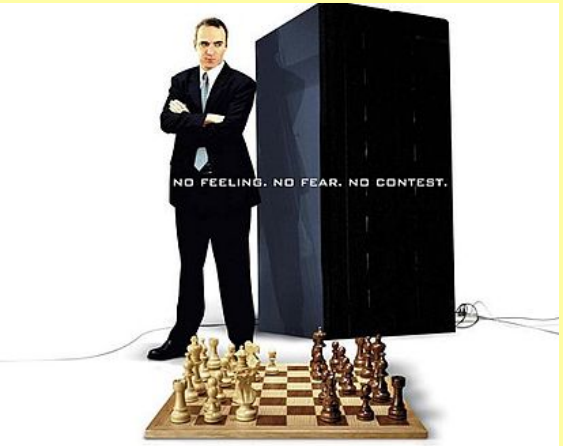
can make decisions that aid its user.

- Plus, it's the name of our project.

Learning

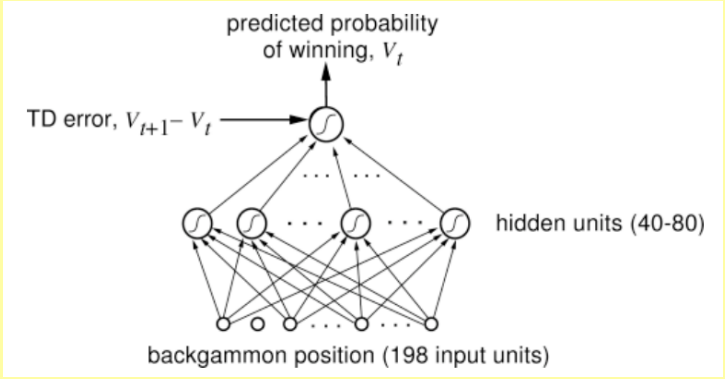
- Ideally, intelligent software can also "learn", so as to ***improve*** its ability to make decisions.

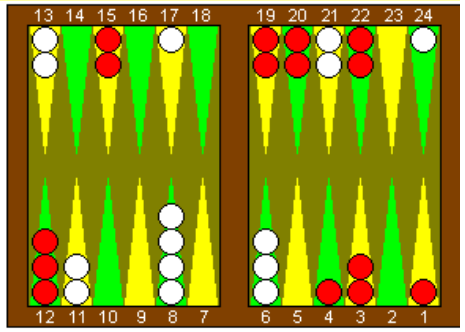
Do these famous AI programs learn?



Deep Blue, 1997
chess computer

Watson (center), 2011
Jeopardy computer





TD-Gammon, 1994

A Few Examples of Prior Art
in Intelligent Music Software

- EMI (Experiments in Music Intelligence)
- Band-in-a-Box
- GenJam
- Artificial Virtuoso & The Continuator
- SmartMusic

EMI (Experiments in Musical Intelligence)
David Cope, UC Santa Cruz, 1981+

- Composes classical music, such as Bach chorales, string quartets, piano sonatas.

- http://artsites.ucsc.edu/faculty/cope/

Band-in-a-Box

PG Music Incorporated, 1990+

- Generates accompaniments from chord changes and style specification.
- Constructs jazz solos, apparently from a database.
- Can extract a style specification from a MIDI performance.
- Proprietary

GenJam (Genetic Jammer)

Al Biles, Rochester Inst. of Tech., 1994+

- Improvises jazz solos.
- Trades interactively with human soloist.
  - http://www.youtube.com/watch?v=xWHU8uE043g
  - http://www.ist.rit.edu/~jab/GenJam.html
- Proprietary

Artificial Virtuoso & The Continuator

François Pachet, Sony Labs, Paris

- Improvise with no musical knowledge, using a wiimote as input controller
- Generate jazz melodies of a preprocessed audio backing track.
- http://www.youtube.com/watch?v=pXXd11jmPTs
- Learns to play in the user's style.

SmartMusic
MakeMusic, Inc.

- Provides feedback for student practice session.

- http://www.youtube.com/watch?v=xhYXO6TPKw4
- http://www.youtube.com/watch?v=VMcXj-1kmeQ

- Invented by Prof. Roger Dannenberg at CMU.

- Proprietary

Emerging Academic Area:
Computational Creativity

- Computers create, or help humans better create: visual art, music, stories, jokes, …
- 10 years of workshops
- First International Conference in Lisbon, 2010
- Second International Conference in Mexico City, 2011

Conventional Wisdom

for learning to improvise

- Choose a solo from some jazz master.
- Transcribe it from audio and memorize it.
- Repeat, until you know how to improvise.

problems with
Conventional Wisdom
for learning to improvise

- Difficult enough to be a show-stopper.
- The learner does not own the result.
- You might end up sounding like a clone (although this is not so likely).

Alternative Way
for learning to improvise

- Pick a tune.
- Construct your own solo over the chord progression of the tune. (Note: You own it.)
- Try to play your solo. Improvise as needed to make it sound good.
- Repeat, with different tunes.

The alternative way led to concept

Impro-Visor

- Punny title for “Improvisation Advisor”.
- A software “workbook” that would help in the alternative method, or even in the conventional method.
- By making suggestions and correcting likely mistakes.

Impro-Visor

Keller, et al., HMC, 2005+

- **Original objective**: A notation tool to help jazz musicians learn to *improvise* by providing suggestions to the student in *composing* his/her own solos.

- Several *secondary* objectives, including:
  - Provide backing tracks
    (similar to Band-in-a-Box)
  - Improvise on its own, as for demonstration or companionship
    (but not yet interactively as does GenJam)
- Free, open-source

Project Participants: HMC

- Prof. Belinda Thom
- Stephen Jones ‘07
- Aaron Wolin ‘07
- David Morrison ‘08
- Martin Hunt ‘08
- Sayuri Soejima ‘10
- Stephen Lee ‘10
- Greg Bickerman ‘10
- Emma Carlson ‘11
- Paul Hobbs ‘12
- Xanda Schofield ‘13
- August Toman-Yih ‘13

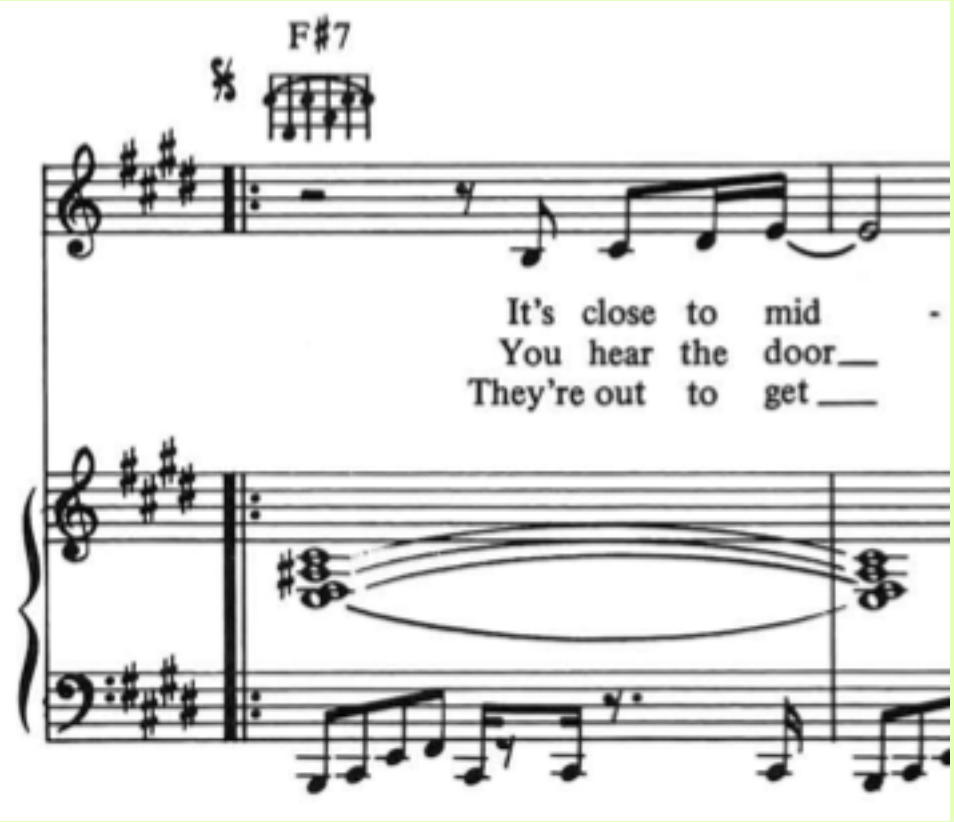
Project Participants: From Elsewhere

- Steven Gomez, Darmouth College
- Jim Herold, Cal Poly Pomona
- Brandy McMenamy, Carleton College
- John Goodman, UK
- Jon Gillick, Wesleyan University
- Kevin Tang, Cornell University
- Chad Waters, Winthrop University
- Peter Swire, Brandeis University
- Sam Bosley, Stanford University
- Lasconic (Nicolas Froment), France
- Julia Botev, Rice University
- Ryan Wieghard, Pomona College
- Zack Merritt, University of Central Florida
- Amos Byon, Troy H.S., Fullerton, CA

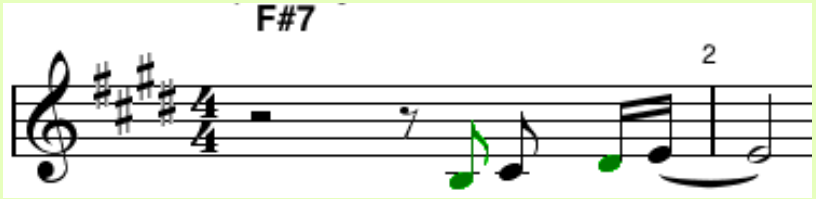
How Impro-Visor Works

- All configuration information is in the form of user-editable text files:
  - **Vocabulary**, defines
    - Scales, Chords, Cells, Idioms, Licks, Quotes
  - **Styles**
  - **Grammars**
  - **Leadsheet**, specifies
    - Chord progression
    - Melody, solo

Leadsheet vs. Sheet Music



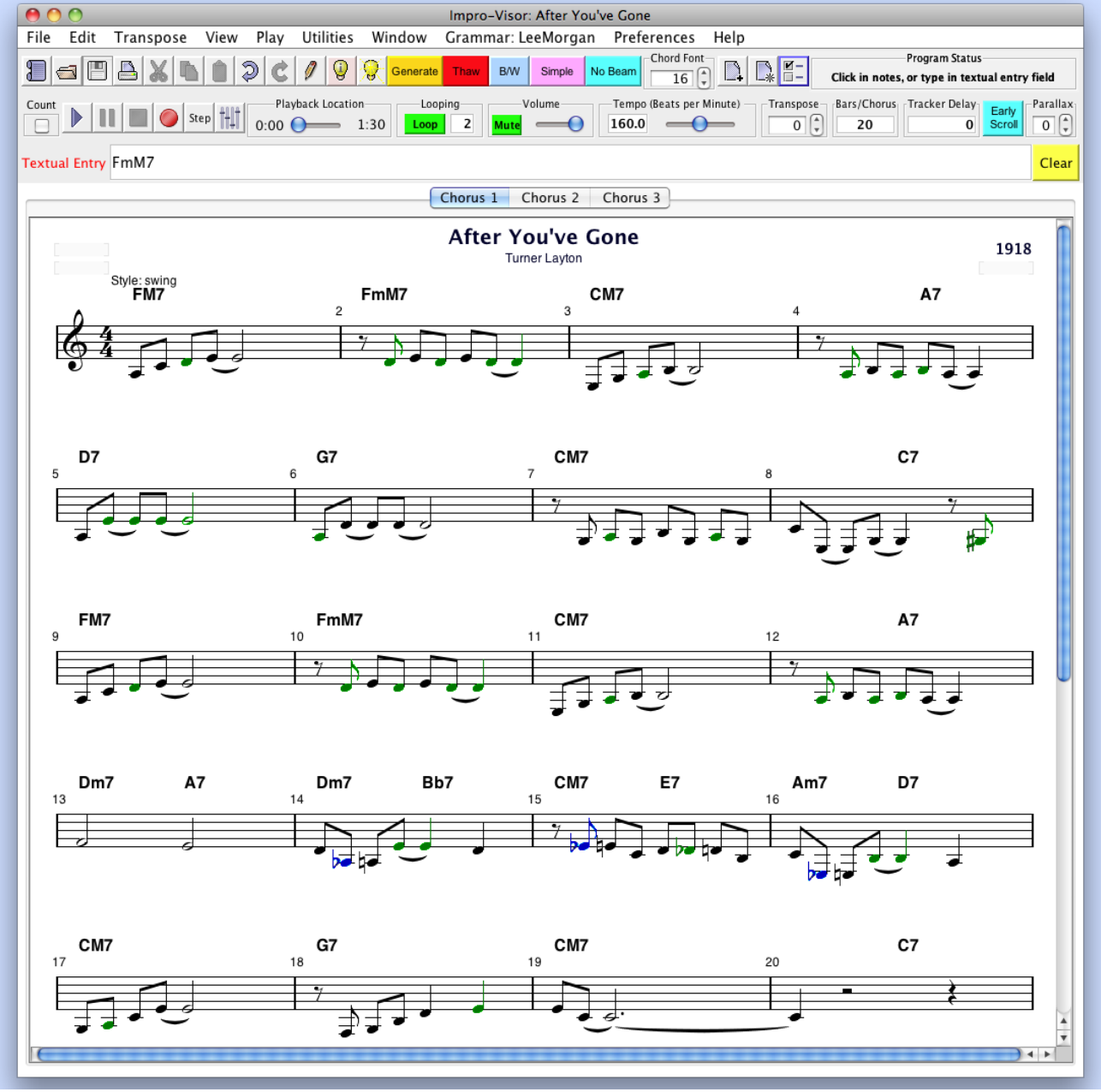
1 bar of **sheet music**



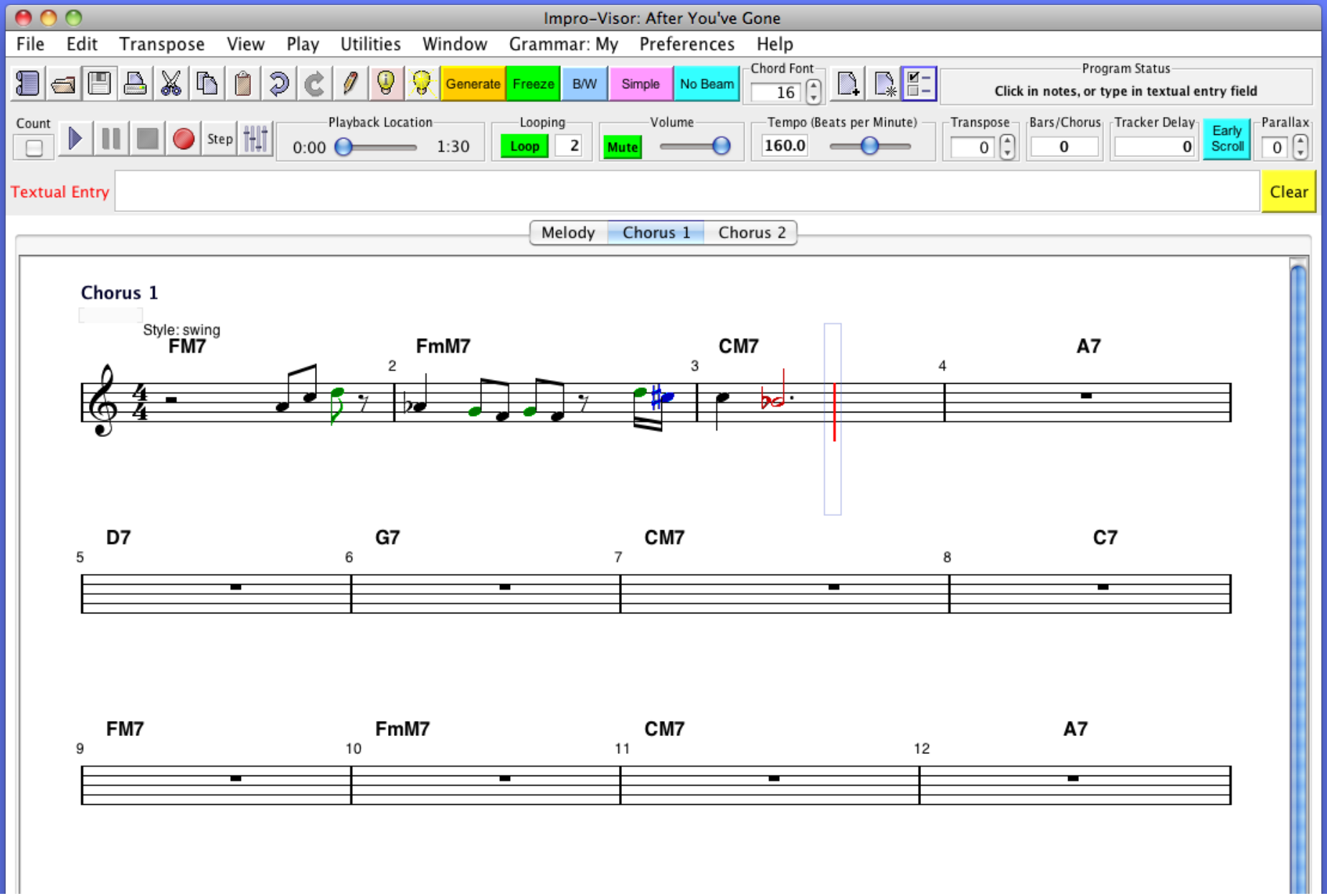
1 bar of a **leadsheet**

The accompaniment is left to the performer.

Impro-Visor's Leadsheet View

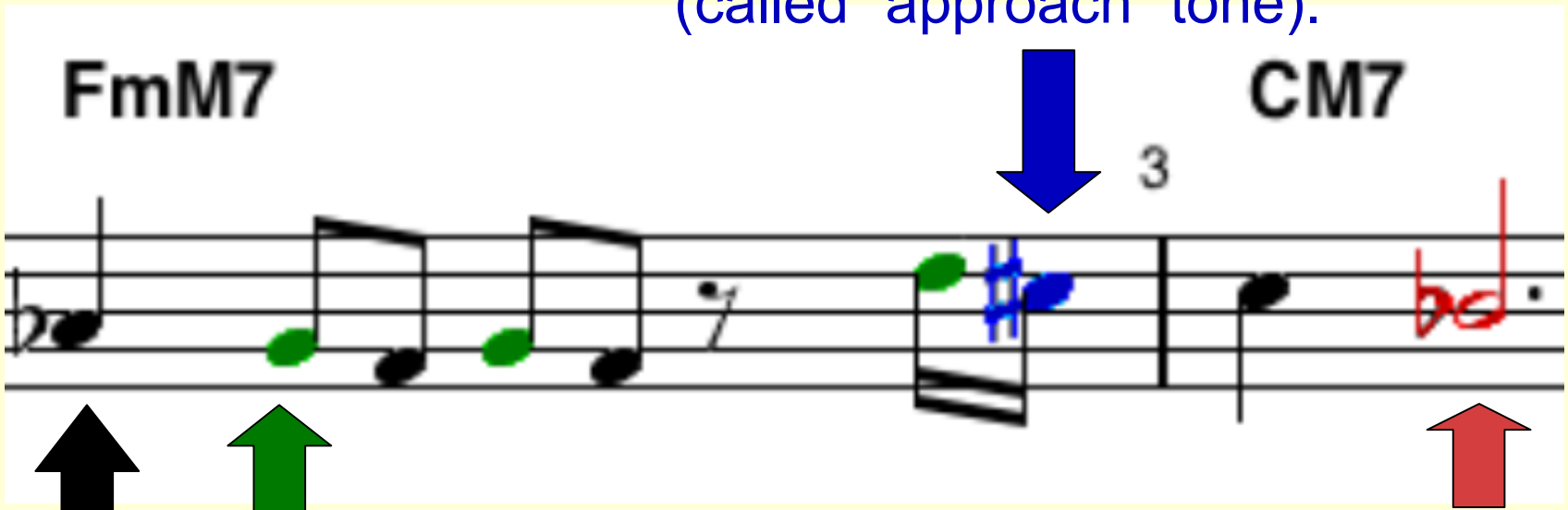


The Improviser's (Person's) Task



Four Note-Color Significance

Blue: Half-step away from chord or color (called “approach” tone).



Red: None of the others (“outside”).

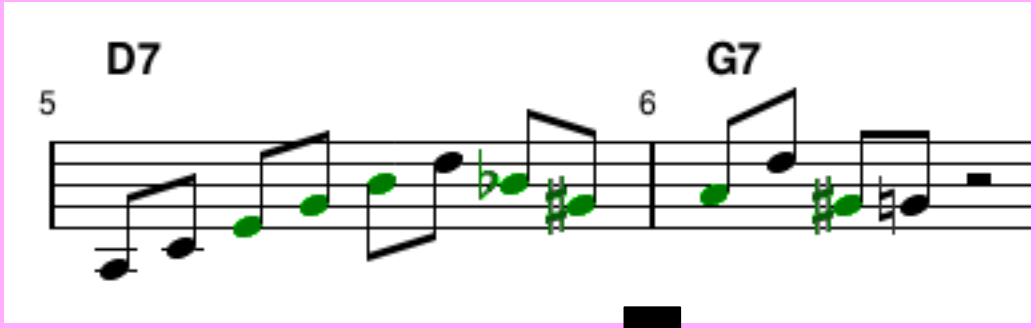
Green: tone not in the chord, but sonorous with it (called “color” tone).

Black: tone in the chord

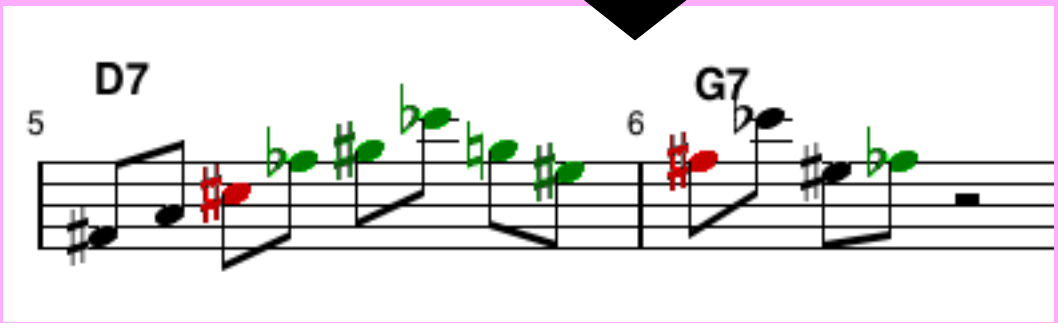
Intelligent Note-Entry Advice

- Four color indicators as just noted.
- Harmonic entry mode: clicked notes gravitate to chord and color tones.
- Harmonic transposition of a group of notes.

Ordinary (Uniform) Transposition

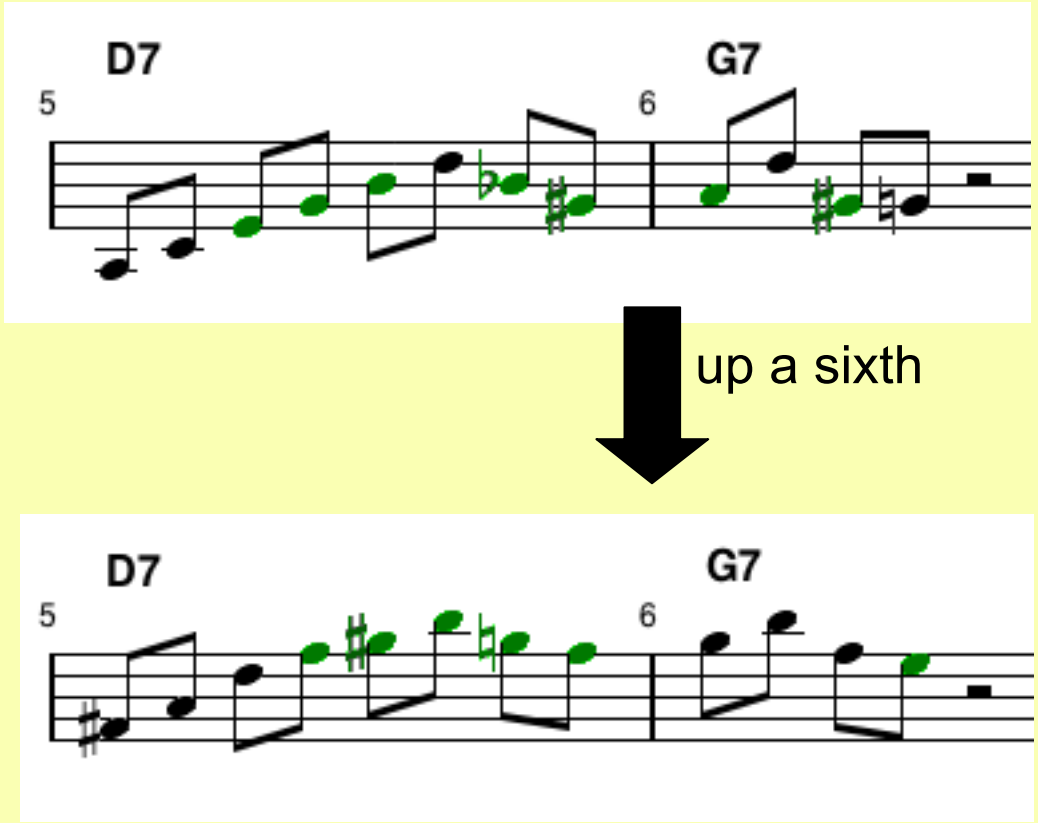


up a sixth



Some discordant notes

Harmonic Transposition

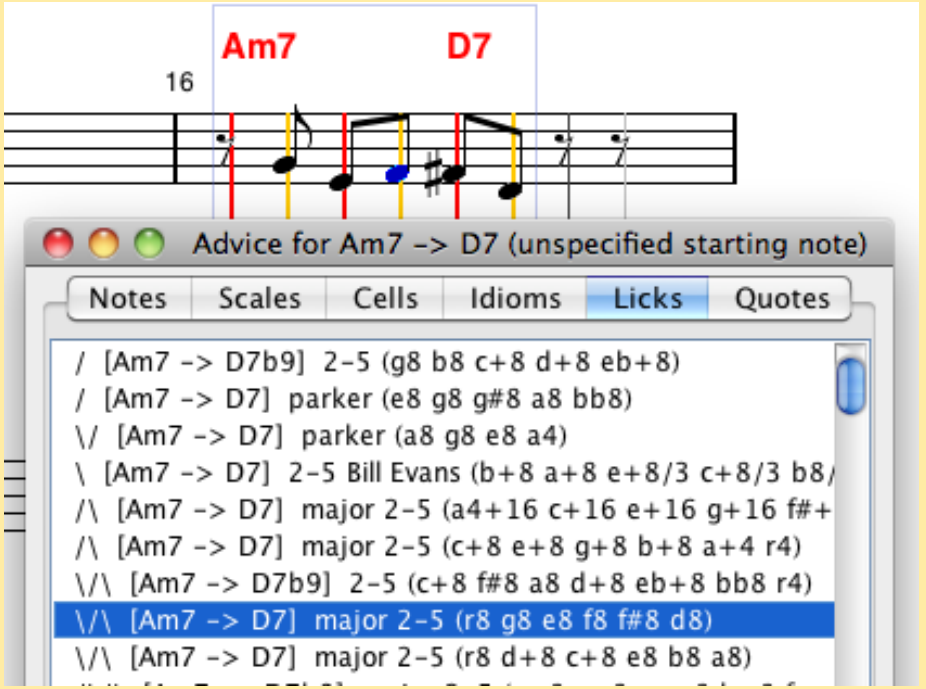


No discordant notes

Generating Licks

- **Lick** = a short melodic phrase
  - sometimes idiomatic
  - sometimes original

- Prior to introducing lick generation, Impro-Visor used a **database** to store lick suggestions.

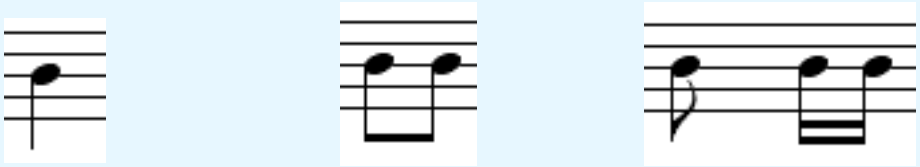


Lick Generation Uses a **Probabilistic Grammar**

- Grammars are a generative **specification**, typically for languages:
  - natural language
  - programming language
  - graphical language
  - musical language
- Typical use in software is **analytic**.
- But Impro-Visor uses a grammar generatively.

Grammar Illustration

- Let B denote one beat of music
- We could fill a beat with a variety of rhythms:



- A grammar represents all of these possibilities:

  B → X4

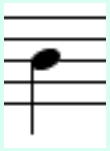
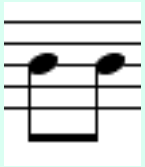
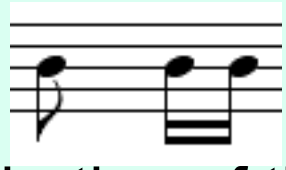
  B → X8 X8

  B → X8 X16 X16

  Here X4, X8, X16 are understood terminal symbols, while B is a non-terminal to be expanded.

Probabilistic Grammar Illustration

- Assign a probability to the various choices
- Probabilities will then dictate a prevalent style

- A grammar represents a distribution of these possibilities:

B → X4	p = 0.3	common
B → X8 X8	p = 0.6	frequent
B → X8 X16 X16	p = 0.1	rare

Grammars Can Exhibit Hierarchy and Recurrence

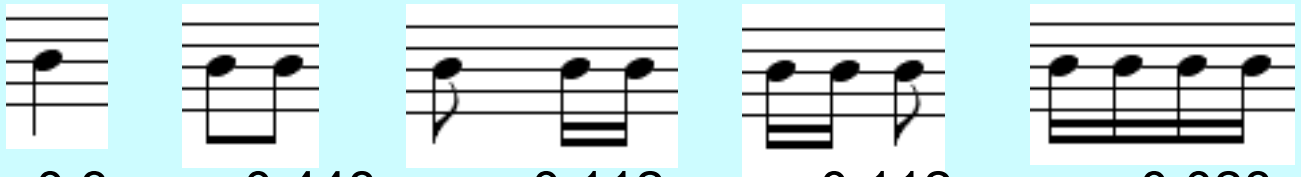
- Instead of

B → X4	p = 0.3	common
B → X8 X8	p = 0.6	frequent
B → X8 X16 X16	p = 0.1	rare

- Use

B → X4	p = 0.3	common
B → C C	p = 0.7	frequent
C → X8	p = 0.8	very frequent
C → X16 X16	p = 0.2	rare

- Generates



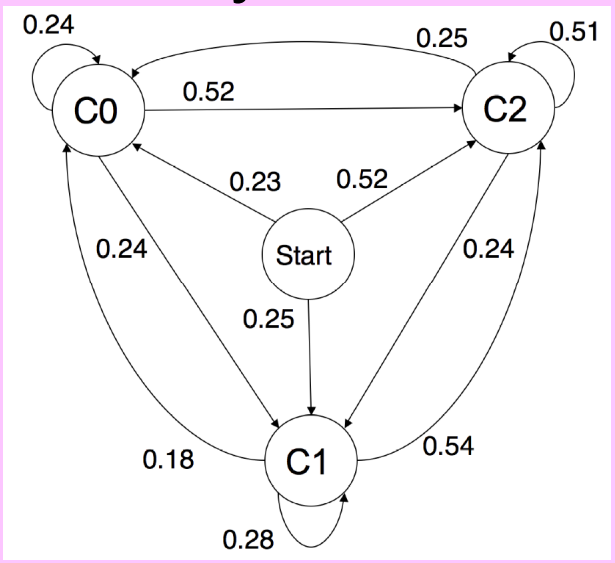
p = 0.3 p = 0.448 p = 0.112 p = 0.112 p = 0.028

Recurrence Allows a Grammar to Fill Arbitrary Number of Beats

- R → B R One beat, then more
- R → empty No expansion

Markov Chains as Grammars

- Recurrent productions allow us to embed an arbitrary Markov chain in the grammar.
- The reason for wanting this will be explained shortly.



Markov chain

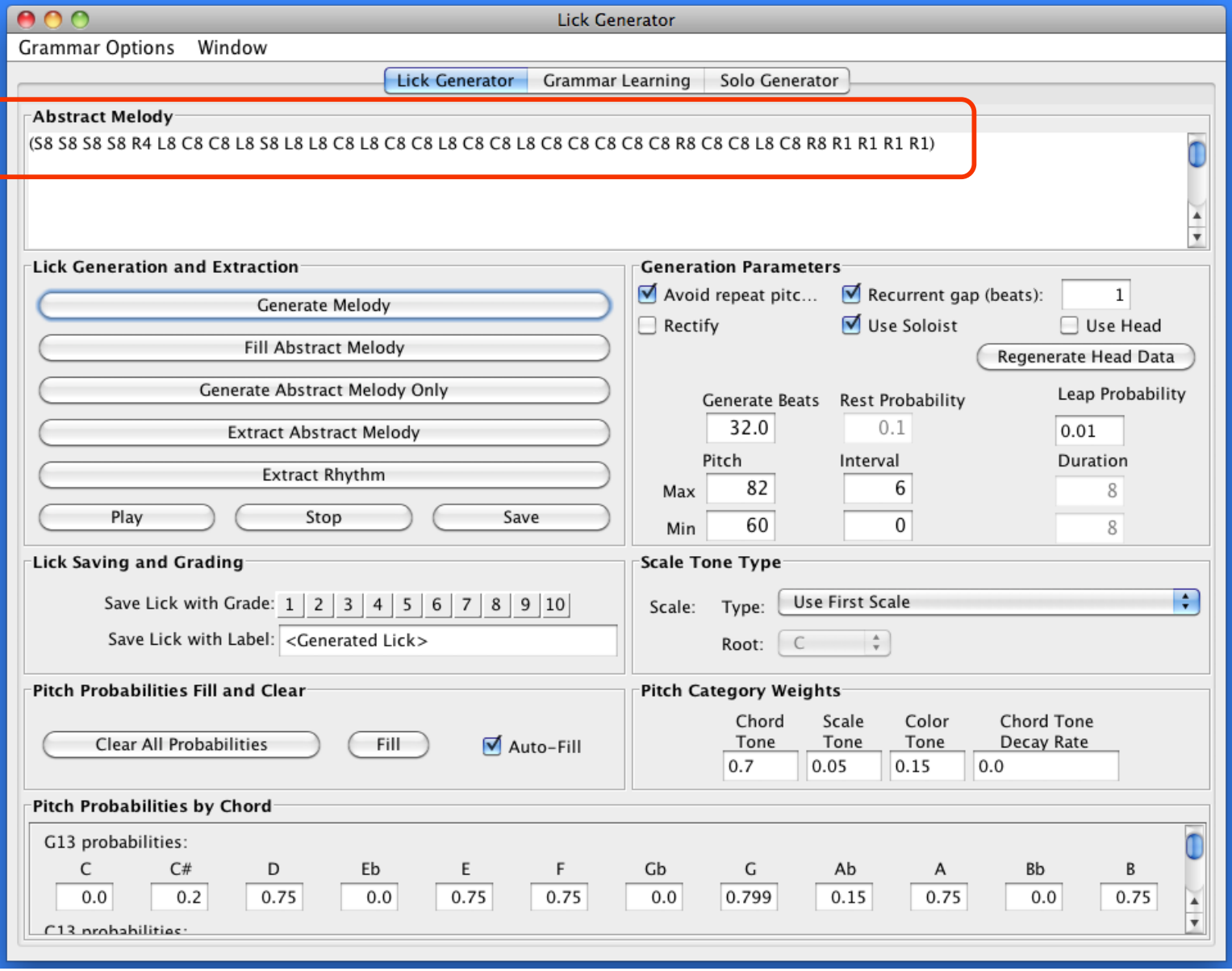
Production Rule	Probability
Start(Z)→ C0(Z)	0.23
Start(Z) → C1(Z)	0.25
Start(Z) → C2(Z)	0.52
C0(0) → ()	1
C1(0) → ()	1
C2(0) → ()	1
C0(Z) → Q0 C0(Z-1)	0.24
C0(Z) → Q0 C1(Z-1)	0.24
C0(Z) → Q0 C2(Z-1)	0.52
C1(Z) → Q1 C0(Z-1)	0.18
C1(Z) → Q1 C1(Z-1)	0.28
C1(Z) → Q1 C2(Z-1)	0.54
C2(Z) → Q2 C0(Z-1)	0.25
C2(Z) → Q2 C1(Z-1)	0.24
C2(Z) → Q2 C2(Z-1)	0.51
Q0→((Δ 0 0 R2 R4 R8 C16/3) (Δ1 1 A16/3 L16/3)	1
Q1→((Δ 0 0 C8) (Δ -9 -9 C8) (Δ 2 3 C8 G4+8 R4))	1
Q2→((Δ 0 0 C4/3) (Δ 1 2 L4/3 A4/3) (Δ-7 -1 C4/3 G4 C8/3))	1

Grammar

Use of Note Color Categories in the Grammar

- In Impro-Visor grammars, **terminal** symbols correspond to the **note categories**, plus note durations.
- We call the string of terminals an **abstract melody**.
- The actual notes are filled in based on the chord of the moment and probabilities.
- This allows a single grammar to be used for an **arbitrary** chord progression.

Abstract Melody Visualized in Impro-Visor's Lick Generator Controls



The Complete Grammar “My Fours” with Terminals in Bold

(startsymbol P)
(base (P 0) () 1.0)
(rule (M4) (**A4**) 0.01)
(rule (M4) (**L4**) 0.2)
(rule (M4) (**S4**) 0.1)
(rule (M8) (**A8**) 0.01)
(rule (M8) (**C8**) 0.4)
(rule (M8) (**L8**) 0.2)
(rule (M8) (**S8**) 0.1)
(rule (N2) (**C2**) 1.0)
(rule (N4) (M4) 0.75)
(rule (N4) (**R4**) 0.25)
(rule (N8) (M8) 0.9)
(rule (N8) (**R8**) 0.1)
(rule (Seg1) (**C4**) 1.0)
(rule (Seg2) (**N2**) 0.06)
(rule (Seg2) (N8 **H4.**) 0.3)
(rule (Seg2) (V2) 0.3)
(rule (Seg2) (V4 V4) 0.6)
(rule (Seg2) (V8 N4 V8) 0.12)
(rule (Seg2) (V8 V8 V8 V8) 0.6)
(rule (Seg4) (H4. N8 Seg2) 0.1)
(rule (Seg4) (H4/3 H4/3 H4/3 Seg2) 0.02)
(rule (Seg4) (Seg2 H4/3 H4/3 H4/3) 0.02)
(rule (Seg4) (Seg2 V4 V4) 0.52)
(rule (Seg4) (V8 N4 N4 N4 V8) 0.01)
(rule (V2) (S16 S16 S16 S16 M4) 0.05)
(rule (V2) (S16/5 S16/5 S16/5 S16/5 S16/5 M4) 0.0050)
(rule (V2) (S8 S8 S8 S8) 0.3)
(rule (V2) (S8/5 S8/5 S8/5 S8/5 S8/5) 5.0E-4)
(rule (V4) (H8/3 H8/3 A8/3) 0.01)
(rule (V4) (H8/3 H8/3 H8/3) 0.05)
(rule (V4) (H8/3 S8/3 H8/3) 0.02)
(rule (V4) (N4) 0.22)
(rule (V4) (V8 V8) 0.72)
(rule (V8) (H16 A16) 0.01)
(rule (V8) (N8) 0.99)
(rule (P Y) (Seg4 Seg4 Seg4 Seg4 R1 R1 R1 R1 (P (- Y 3840))) 1)

Grammar Construction

- Grammar construction by hand is fun, but tedious.

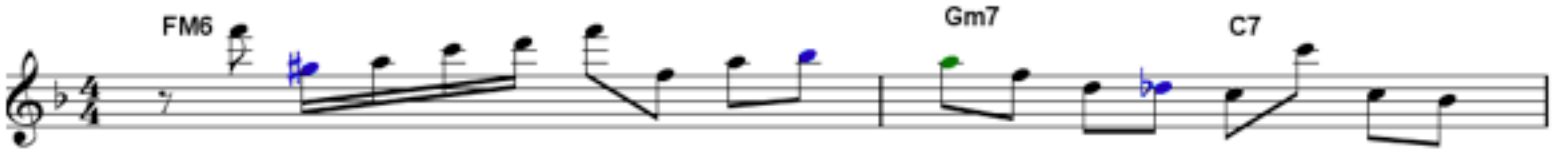
- A better approach might be to have the software **learn the grammar** from examples.

Grammar Learning Feature

- Impro-Visor can **learn a grammar** by examining one or more transcribed solos.
- For greater coherence special construct called a “slope” is introduced, from which **melodic contours** can be constructed.
- Slopes can appear in the rules and contain terminals.

Slopes Encode Contours

(a) Original melody:



(b) Melody contour:

(c) Abstract melody using slopes (Δ's):

(R8 C8 (Δ -9 -9 A16) (Δ1 3 C16 C16 C16 C8) (Δ-12 -12 C8) (Δ1 4 C8 A8)

(Δ -4 -1 L8 C8 C8 A8 C8) (Δ12 12 C8) (Δ-12 -2 C8 C8))

From Transcription to Grammar

1. The transcription is “windowed” into small **chunks**, say 1 or 2 bars long.

2. Each window contents becomes an **abstract melody**.

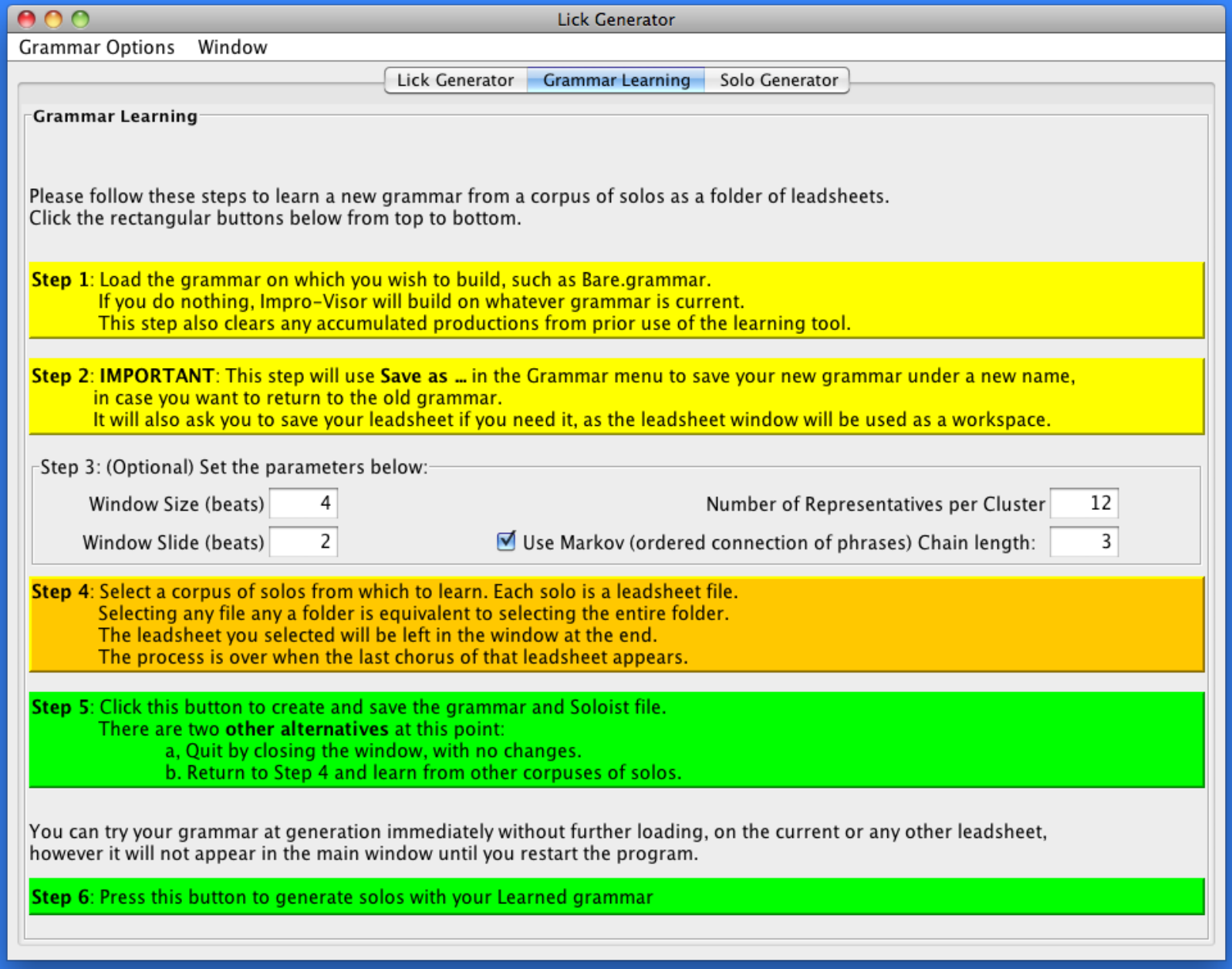
3. The set of abstract melodies are **clustered** by similarity. The **clusters become the nodes** of a Markov chain.

4. The **transition probabilities** for the chain are obtained by re-examining the transcription.

5. The chain is converted to a **grammar**, with selected **representatives of clusters encoded as slopes**.

The entire process takes a **few seconds**, depending on the size of transcriptions.

Impro-Visor's Grammar Learning Interface



A Blind-Evaluation Experiment

- Grammars were inferred from solos of **3 different** famous trumpet players with different styles.

- Subjects were asked to listen to the original solos, plus solos generated from the grammar on a different tune, to **see if they could match the styles**.

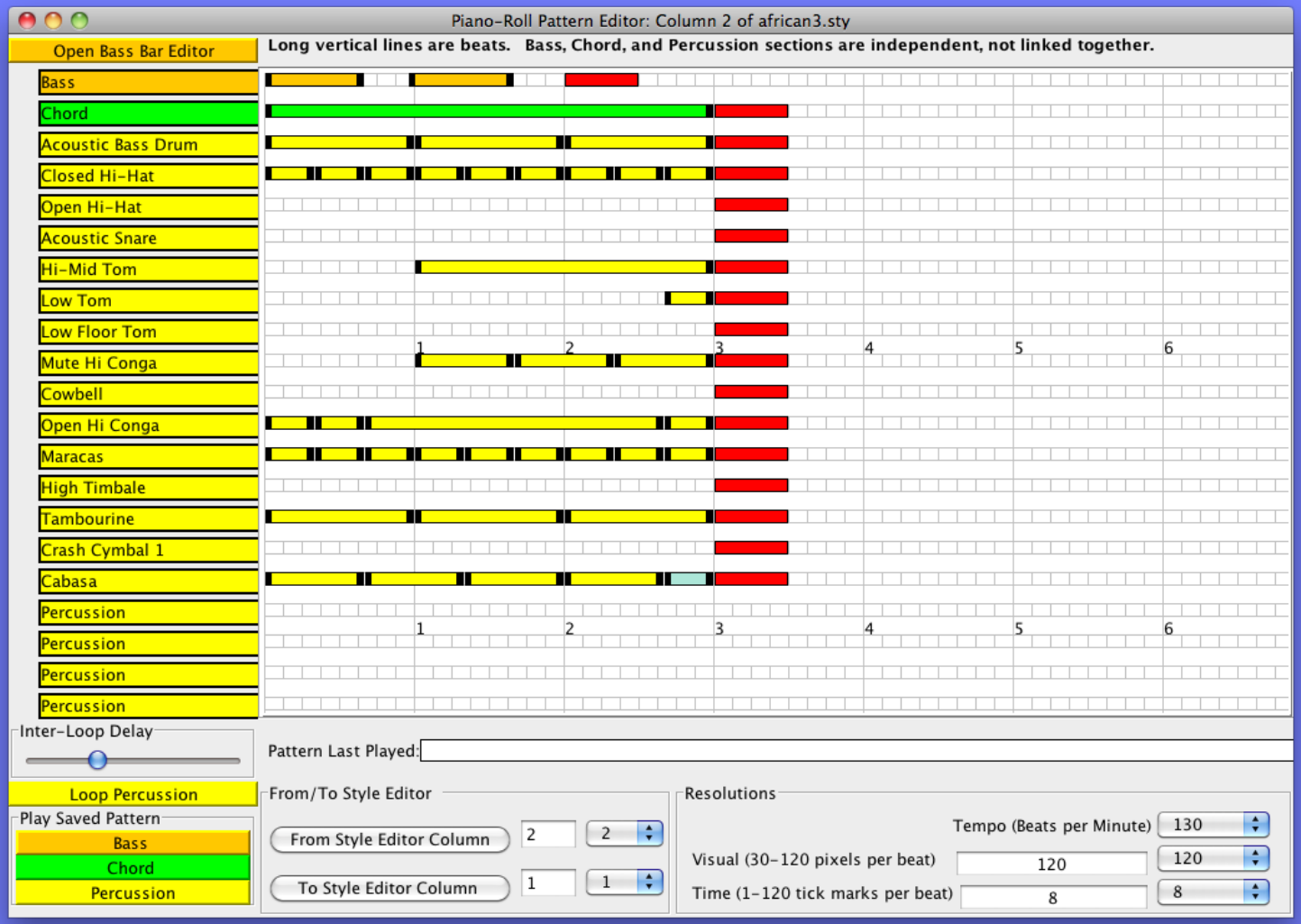
- Correct matches were obtained at 95%, 90%, and 85% levels for the soloists, and 85% of subjects correctly matched all three.

Other Learning in Impro-Visor

- Impro-Visor can **learn** a style specification (in its own language), given two inputs:
  - A MIDI file of a performance in that style.
  - A leadsheet file indicating the corresponding chords.
- As with grammar learning, clustering is used.

- A research problem is to **eliminate the second requirement**. The chords would need to be identified to construct the bass patterns.

Style Pattern Represented in Impro-Visor's Piano-Roll Editor



A Different Approach to Learning: **RBM-provisor**

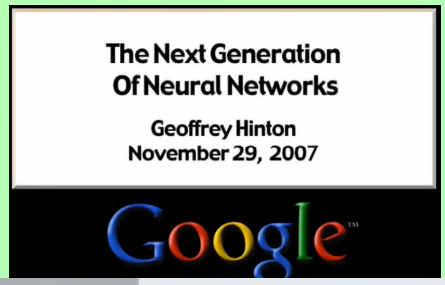
- We applied **Restricted Boltzmann Machines** (RBMs) in the form of Deep Belief Networks to the problem of improvising music.
- RBMs are neural networks based on **probabilities** of switching, determined by learned synaptic weights.
- An RBM tries to learn a set of **concepts** based on a set of input samples.
- They stabilize to a probability distribution reflecting those concepts, and can generate music probabilistically.

Deep Belief Networks

Geoffrey Hinton, U. of Toronto

- Hinton demonstrated how a stack of RBM's can learn higher order concepts sufficient to perform tasks such as digit recognition.

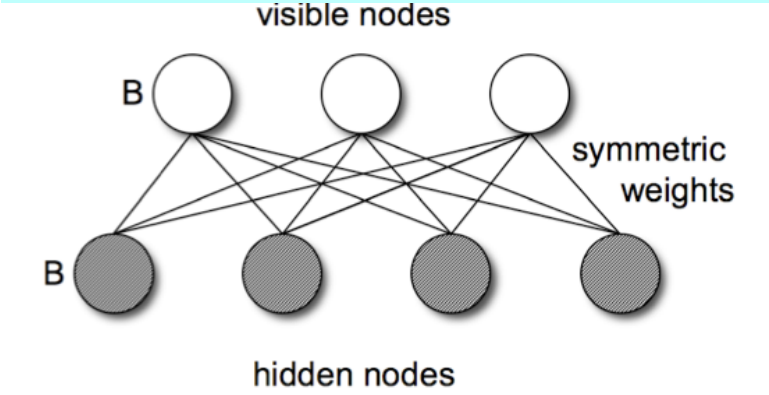
http://www.youtube.com/watch?v=AyzOUbkUf3M



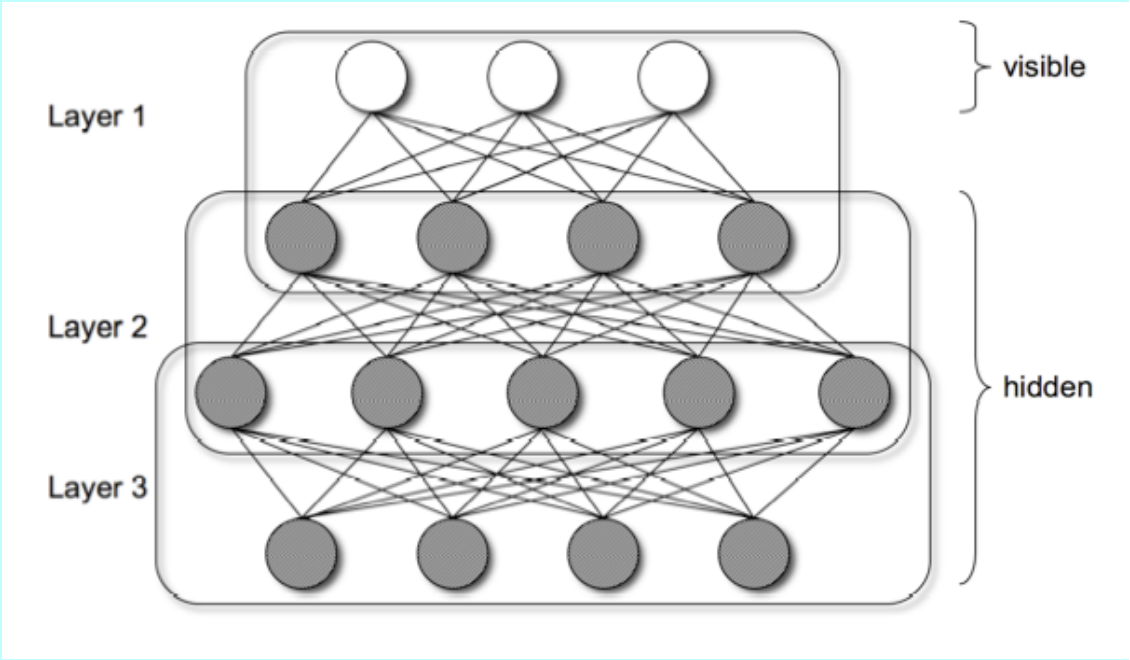
- We applied a similar idea to learning concepts that produce melodies from chord progressions.

- The idea was to build in **as little musical knowledge as possible**.

Restricted Boltzmann Machines
&
Deep Belief Networks

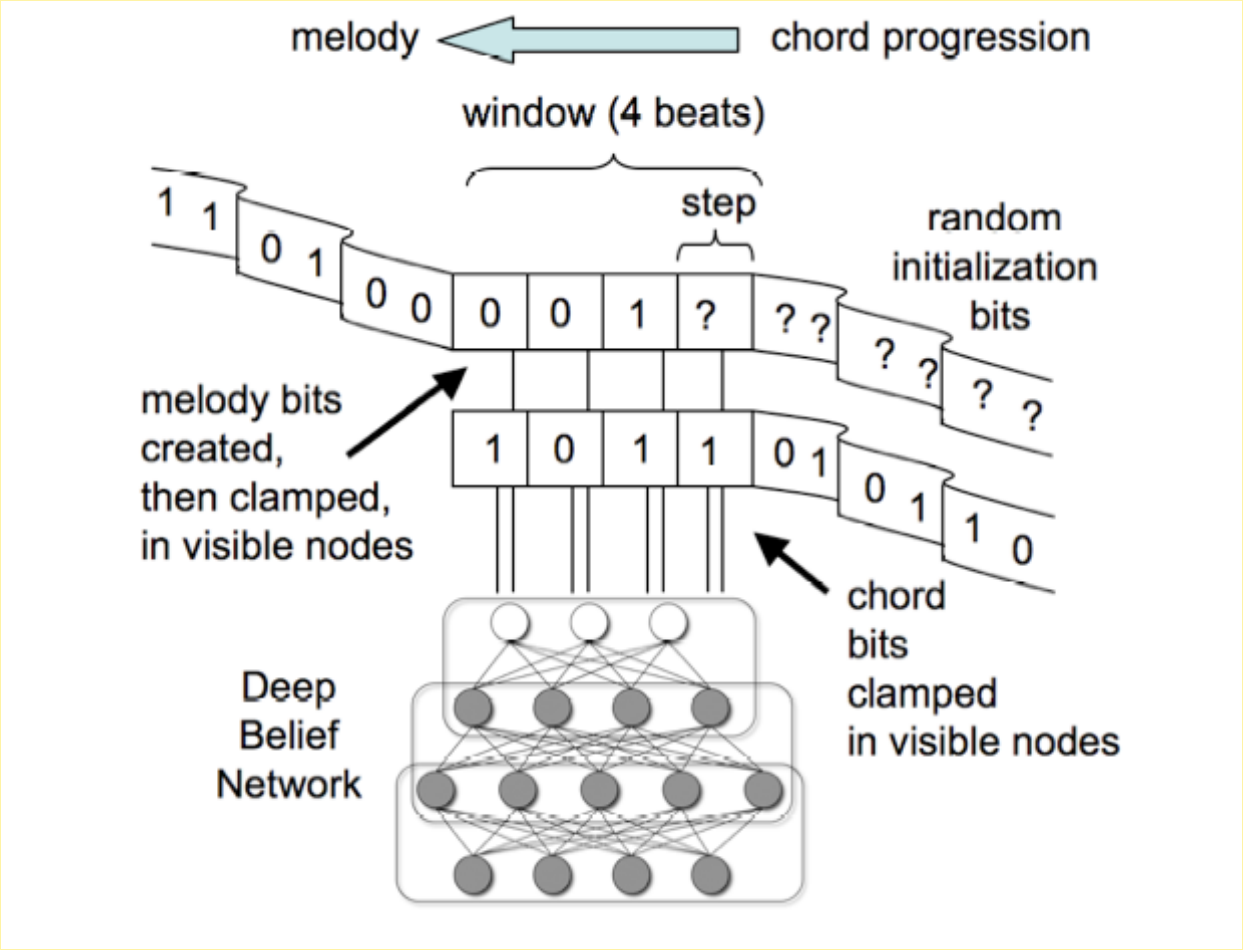


RBM



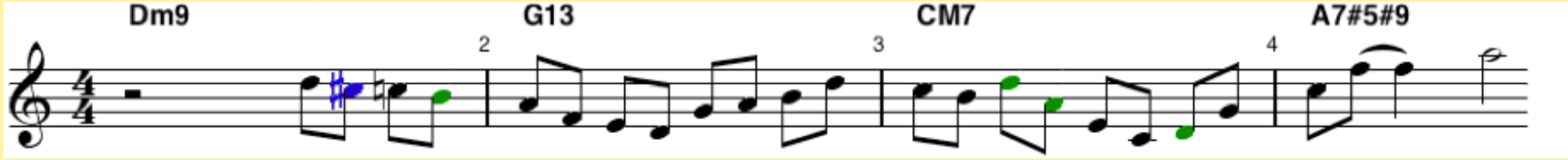
DBN (3-layer)

Improvising Jazz with a Deep Belief Network

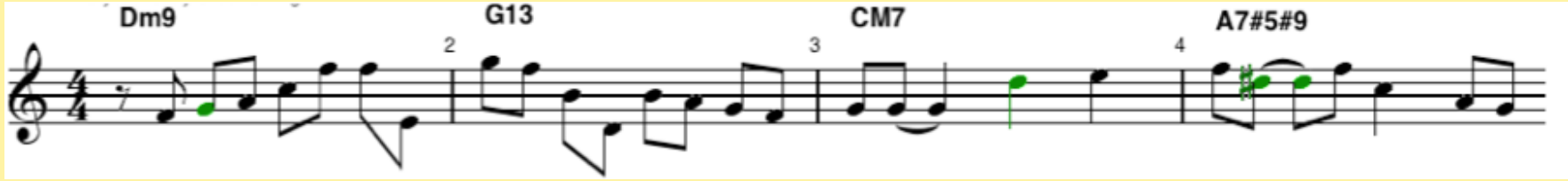


RBM-provisor Examples

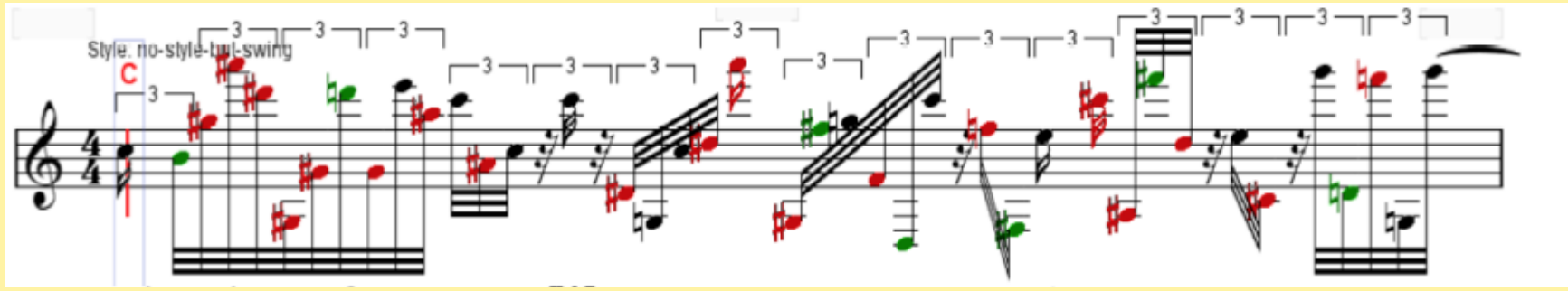
Example from Training Set



Output from Trained Network



Output from Untrained Network (Random)



Current R&D

- A modular approach to representing and manipulating harmonic sequence ("chord bricks") and key centers.
- Help musicians understand tune construction.
- Help players recognize the importance of key centers in improvisation.

Some References

- http://www.impro-visor.com

- Keller, Jones, Morrison, Thom, and Wolin, **A Computational Framework Enhancing Jazz Creativity**, *Third Workshop on Computational Creativity*, 2006 (ECAI '06).

- Gillick, Tang, and Keller, **Machine Learning of Jazz Grammars**, *Computer Music Journal*, 34:3, pp. 56–66, Fall 2010, MIT.

- Bickerman, Bosley, Swire, and Keller, **Learning to Create Jazz Melodies Using Deep Belief Nets**, *Proc. First International Conference on Computational Creativity*, 228-237, January, 2010.