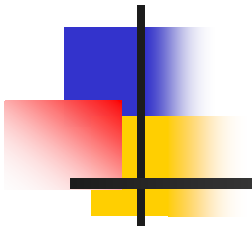
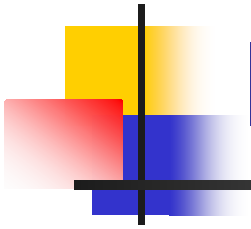


Intelligent Music Software

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

IEEE Computer Society
23 February 2012



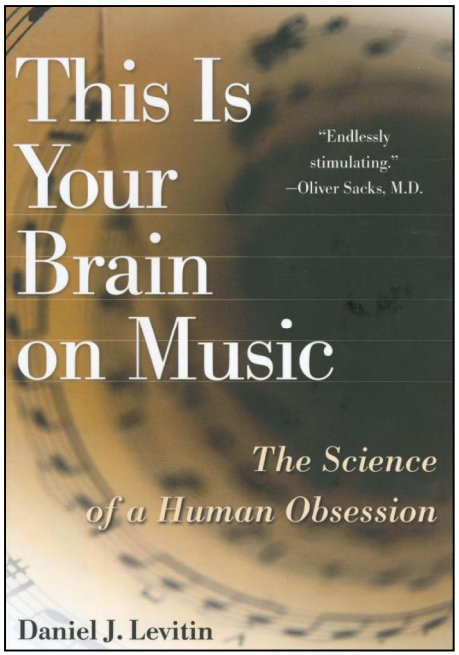
Interaction

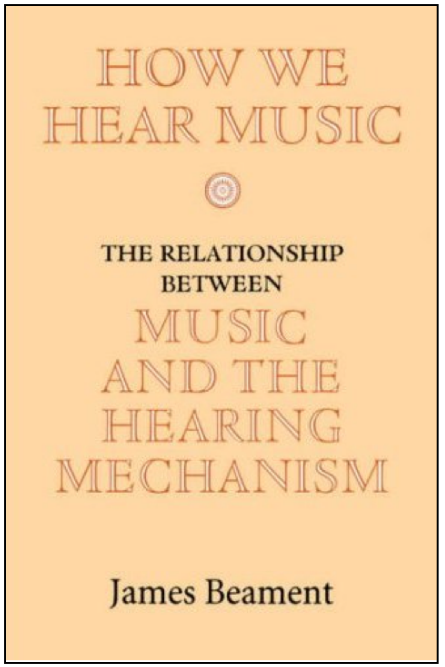
Questions that can be answered briefly are welcome during the talk.

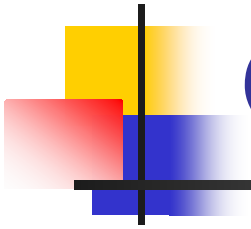
Music can be …

- entertaining
- pleasurable
- challenging
- therapeutic
- an object of study
  - mathematical
  - scientific
  - psychological

  …

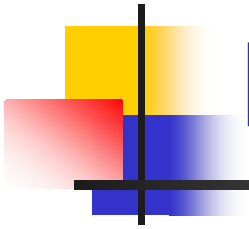






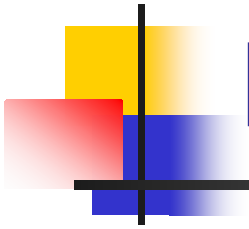
Outline

- Describing the space
  - Music software in general
  - Intelligent music software
  - Prior art
- Intelligent Music Software Project at Harvey Mudd College
  - Impro-Visor
  - RBM-provisor: Using Deep-Belief Networks
  - Recent work



Music Software Varieties

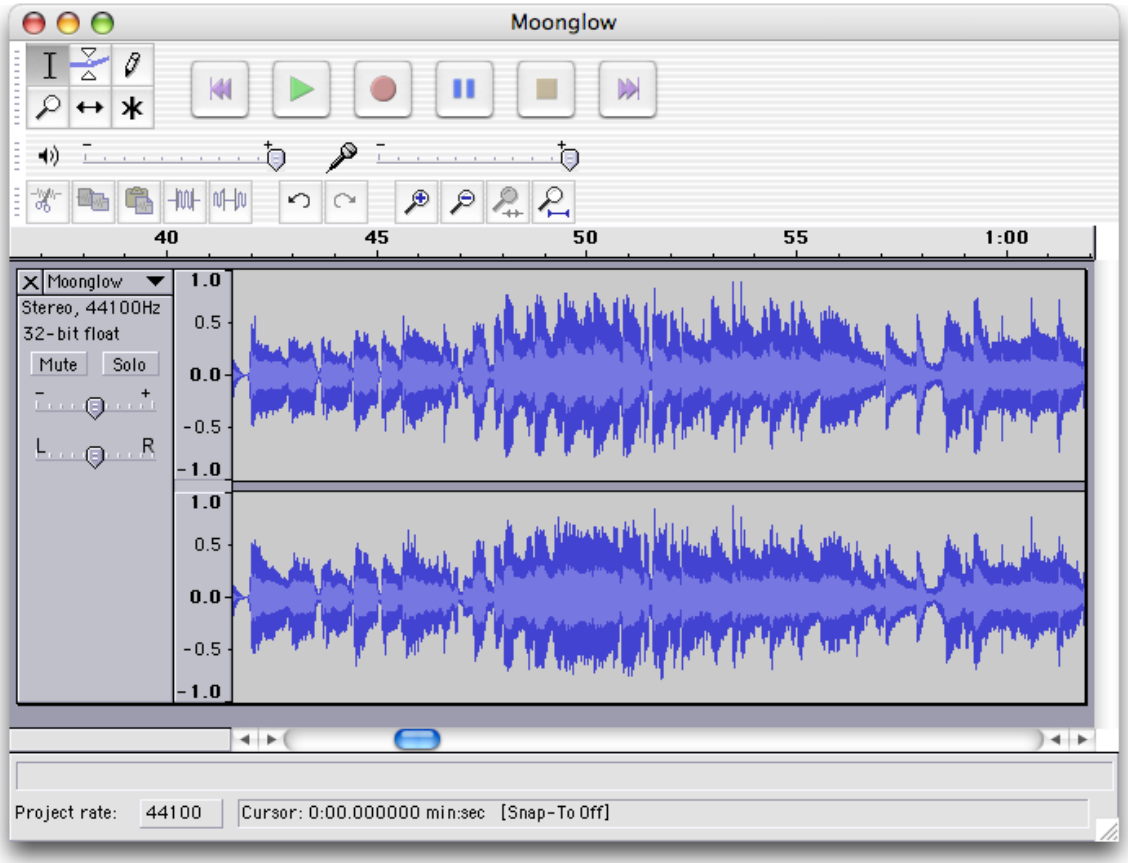
- Consumer products
  - Music player (mp3, aiff, wav, …)
  - Music organizer, searcher
  - Music recommender
  - Music recorder
  - Music synthesizer



Music Software Varieties

- Musician products
  - Music notation editor (“scorewriter”)
  - Digital audio workstation (DAW)
  - Music transcriber (audio to score)
  - Music generator (create music)
  - Music composition assistant
  - Music score follower (educational)

Example: **Audacity** (FOSS*)
Record and edit audio (sound)
(Dominic Mazzoni, HMC ‘99, CMU ‘01)



Dominic Mazzoni
and Xanda Schofield
in our lab, 2012

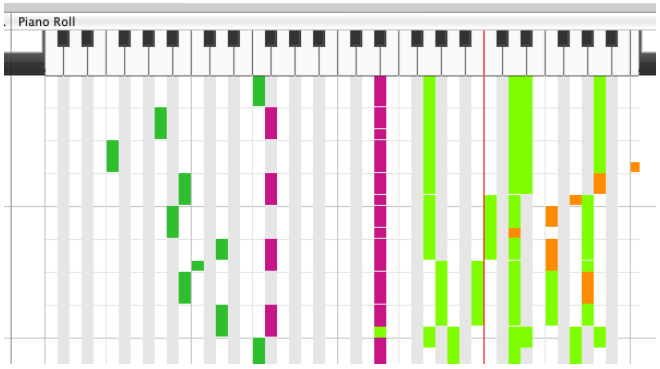
*FOSS = “Free, Open-Source, Software”

Two Major Music Universes

- **Audio** (mp3, wav, AIFF, au, …)

- **Symbolic**

MIDI = “Musical Instrument Digital Interface”



Rendering MIDI to Audio

- Electronically
  - Most digital pianos
  - Software players (ARIA, Kontakt, …) drive samples recorded from acoustic instruments
  - Synthesizers
- Mechanically
  - Modern versions of the player piano: Disklavier, Pianomation, PianoDisc

Instruments that Emit MIDI

Akai EWI-USB

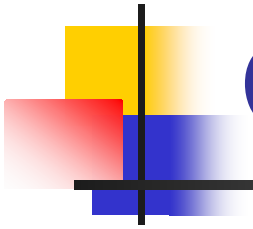
Electronic Wind Instrument

Yamaha Tenori-On

EVL MIDI Violin

Kitara MIDI Guitar

Apple iPad



Crossing the Universes

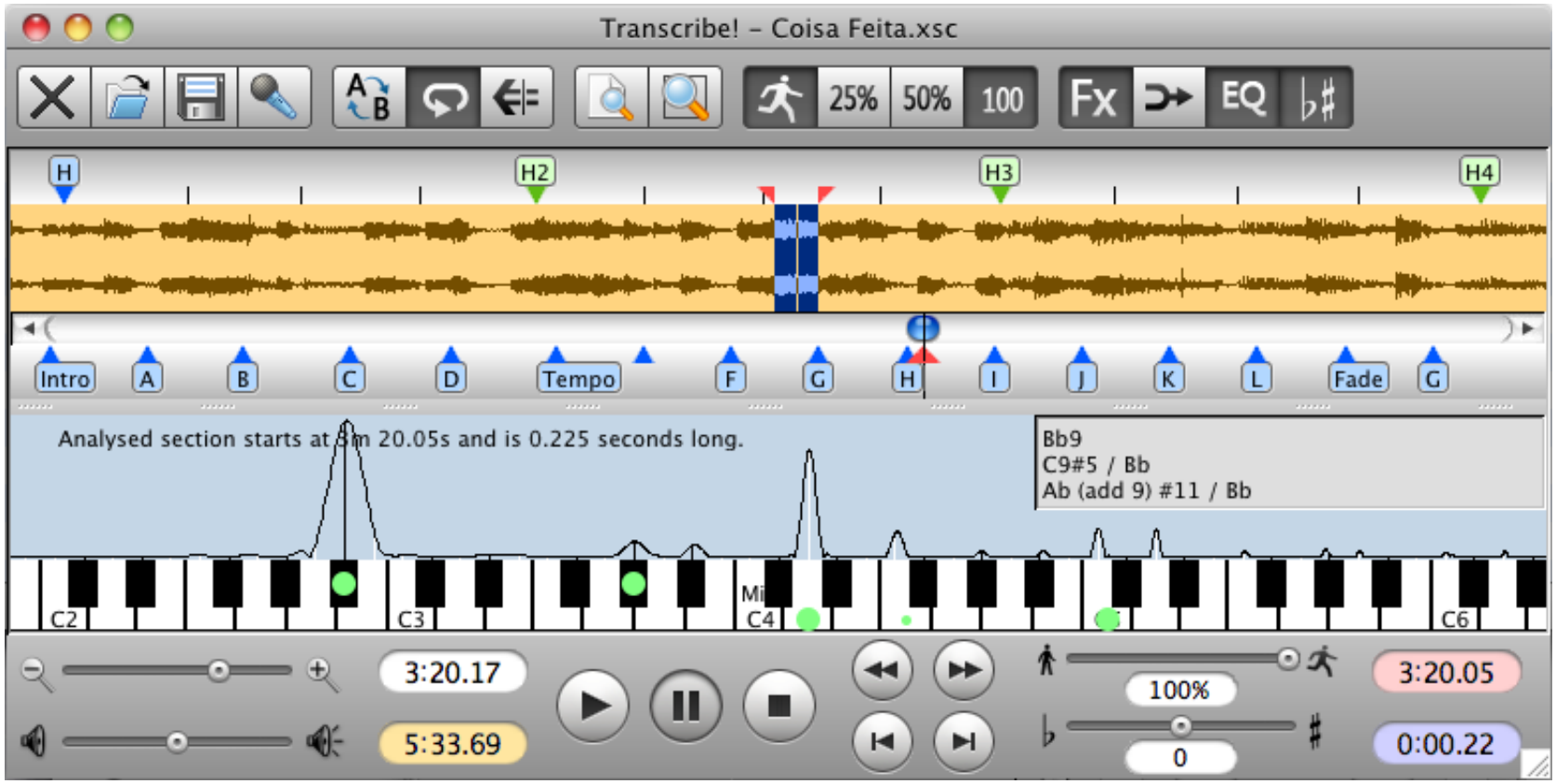
- MIDI → Audio
  - relatively easy
- Audio → MIDI
  - harder
  - doable for monophonic audio
  - difficult for polyphonic audio (e.g. chords, multiple instruments)

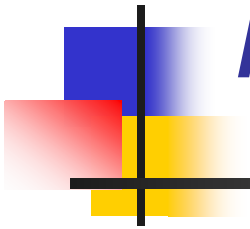
Example: **Transcribe!**

SeventhString Software .com

Transcription (slow-down) software

Analyzes audio spectra to suggest pitches for human transcription





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: **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

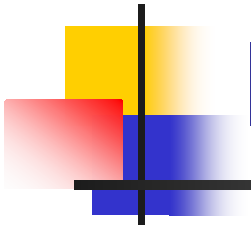
wikipedia

- *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.



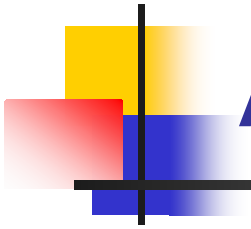
Intelligence

- I will adopt the convention that

Intelligent Music Software

makes decisions that aid its user.

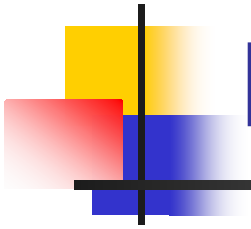
- (plus, it's the name of our project.)



Assertion

Any behaviors of current software that seem to be intelligent are results of the (meta-) intelligence of the software's designers.

This includes learning.

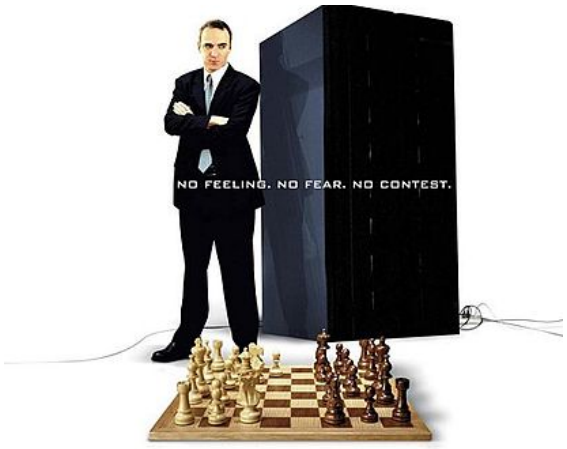


Learning

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

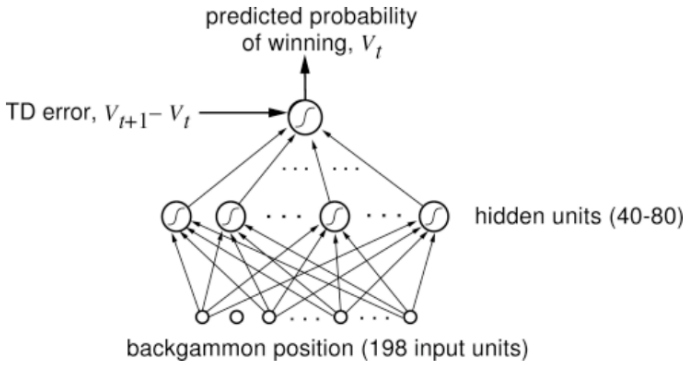
- Also ideally, humans can learn from the software, whether or not the software learns.

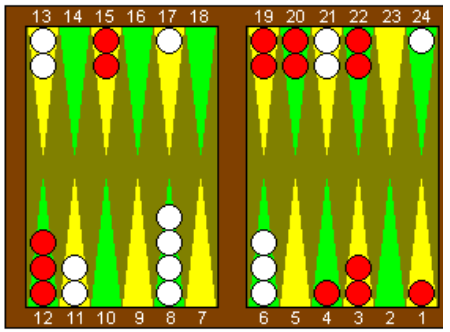
Which, if any, of these famous AI programs learn?



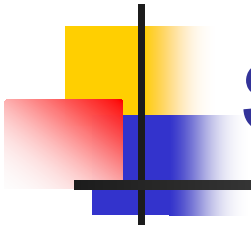
Deep Blue, 1997
chess computer

Watson (center), 2011
Jeopardy computer





TD-Gammon, 1994



Some Intelligent Music Candidates

- SmartMusic
- Digital Ear Real-Time
- Melodyne Editor
- IntelliScore Ensemble
- Band-in-a-Box
- EMI (Experiments in Music Intelligence)
- GenJam
- Artificial Virtuoso & The Continuator

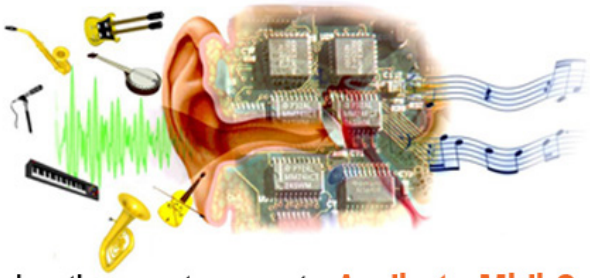
SmartMusic
MakeMusic, Inc.

- Provides feedback for student practice sessions ("used by over 75,000 students worldwide")
- http://www.youtube.com/watch?v=xhYXO6TPKw4
- Developed by Prof. Roger Dannenberg at CMU.
- Proprietary

Digital Ear Real-Time

digital-ear.com, Epinoisis Software

Converts monophonic audio to MIDI



Introducing the most accurate **Audio-to-Midi Converter**

“ Charlie Parker sounds pretty good as piccolo player! ”

-- DownBeat magazine

“Nothing could be simpler”

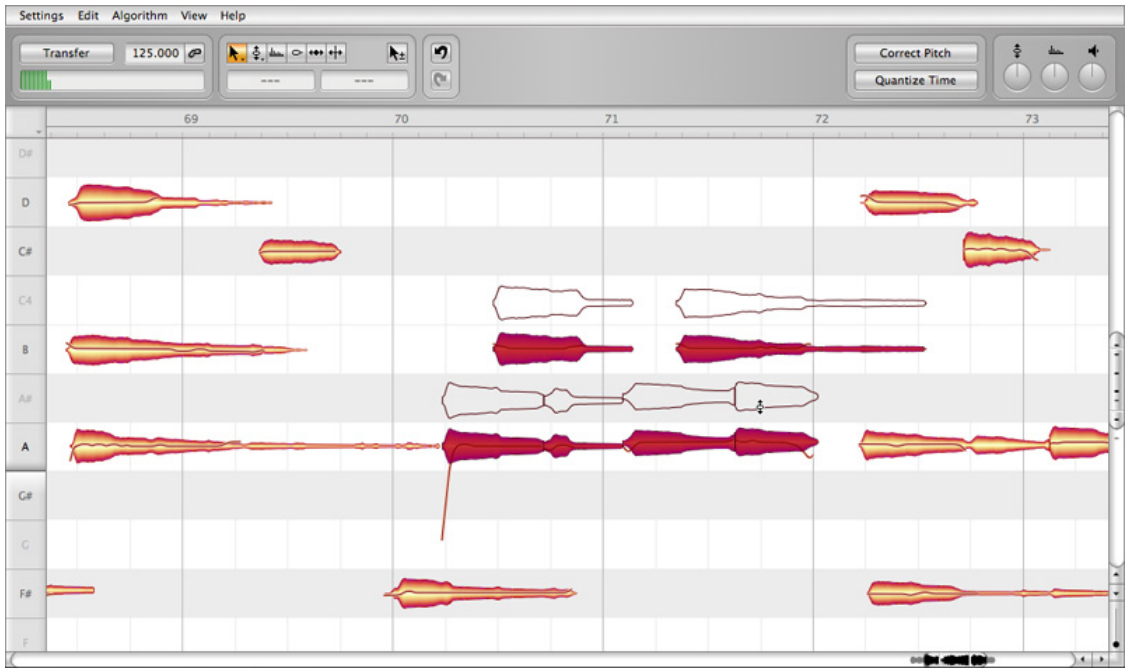
-- Ziff-Davis Network: Editor's Pick Award

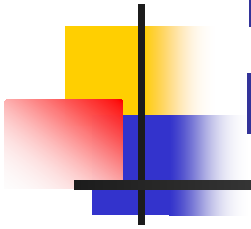
Melodyne Editor, 2000
celemony.com, Munich


Peter Neubäcker

- Polyphonic audio transcription
- Edit individual notes within chords
- Proprietary

Pitches





Intelliscore Ensemble

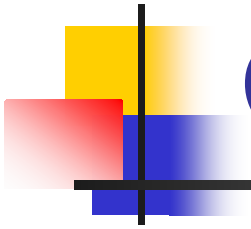
Innovative Music Systems, Inc., 2000+

- WAV to MIDI, off-line
- Polyphonic, but editing may be needed
- Patent number 6,140,568: “System and Method for Automatically Detecting a Set of Fundamental Frequencies Simultaneously Present in an Audio Signal."

Music Plus One (formerly Music++)

Prof. Chris Raphael, Indiana University, 1998+

- Virtual orchestra anticipates player's tempo, follows retakes, etc.
- http://www.music.informatics.indiana.edu/~craphael/music_plus_one/index.html



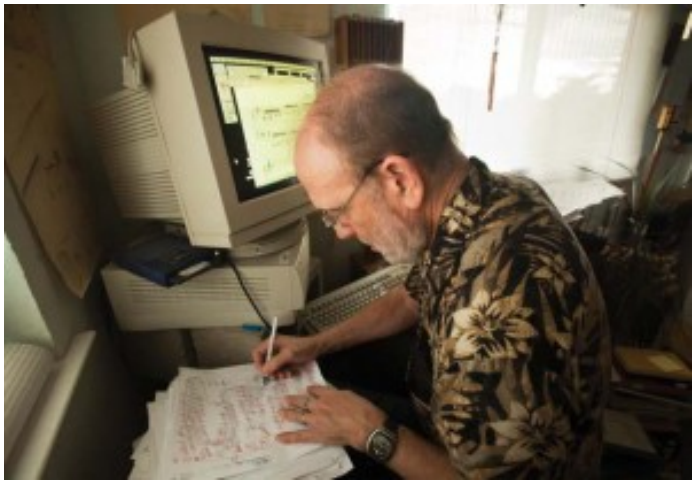
Creativity

- Ideally, intelligent music software can also “create”, i.e. use its ability to make decisions to produce new results that will intrigue the user.

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

- "Emmy", then "Emily Howell", composed classical music, such as Bach chorales, string quartets, piano sonatas.

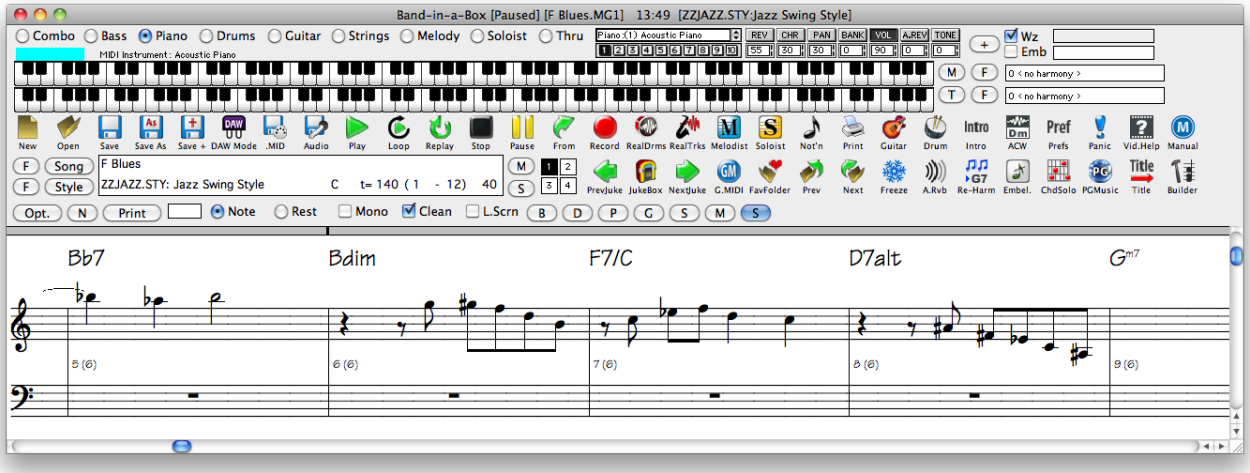
- http://www.miller-mccune.com/culture/triumph-of-the-cyborg-composer-8507/



Band-in-a-Box

PG Music Inc., 1988+, Peter Gannon

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



GenJam (Genetic Jammer)

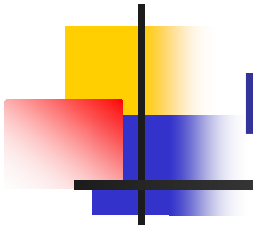
Al Biles, Rochester Inst. of Tech., 1994+

- Improvises jazz solos, based on genetic algorithm
- Trades interactively with human soloist.
  - http://www.youtube.com/watch?v=xWHU8uE043g
- Proprietary

Artificial Virtuoso & The Continuator

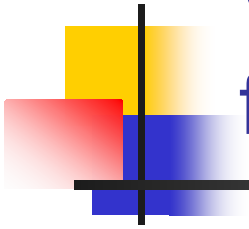
François Pachet, Sony Labs, Paris

- Improvise with no musical knowledge, using a Sony wiimote as input controller
- Generate jazz melodies of a preprocessed audio backing track.
- http://www.youtube.com/watch?v=pXXd11jmPTs (especially last few seconds)
- “Learns to play in the user’s style”.



HMC Intelligent Music Software Project

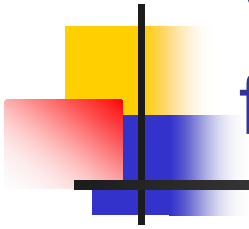
- Oriented toward helping musicians learn to improvise
- Focus is on jazz education, but not limited to jazz



Conventional Wisdom

for learning to improvise

- Choose a solo from some jazz master.
- Transcribe the solo from audio
  and memorize it.
- (Study chords, scales, etc.)
- Repeat, until you “know how to improvise”.



problems with Conventional Wisdom for learning to improvise

- Difficult enough to be a show-stopper for many newcomers.
- The learner does not own the result.
- The learner might end up sounding like a clone of some famous player (you wish!).

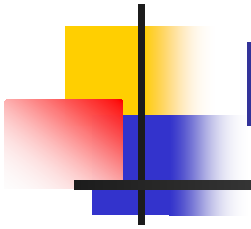
Our Alternative Way
for learning to improvise

- Pick a tune.
- Construct your own solo over the chord progression of the tune.
  (Note: You own it.)
- Try to play your solo.
  Modify as needed to make it sound good.
- (Study chord and scales.)
- Repeat, with different tunes, until you “know how to improvise”.

Impro-Visor (FOSS)
Keller, et al., HMC, 2005+

- Short for “Improvisation Advisor”.

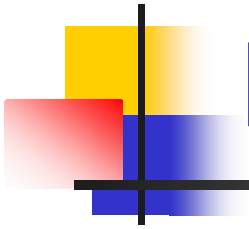
- A software “workbook” that can help in both the alternative method and the conventional method.



Impro-Visor Objectives

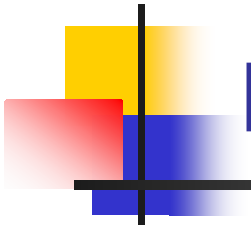
- **Original objective**: A notation tool to help jazz musicians learn to *improvise* by providing suggestions to the student in *composing* his/her own solos.

- **Secondary objectives** include:
  - Immediate feedback, visual and aural
  - Provide backing tracks
    (similar to Band-in-a-Box, but more tutorial)
  - Improvise on its own, for demonstration or companionship
    (not yet interactive as real-time GenJam)



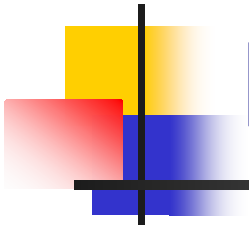
Project Participants: HMC

- 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

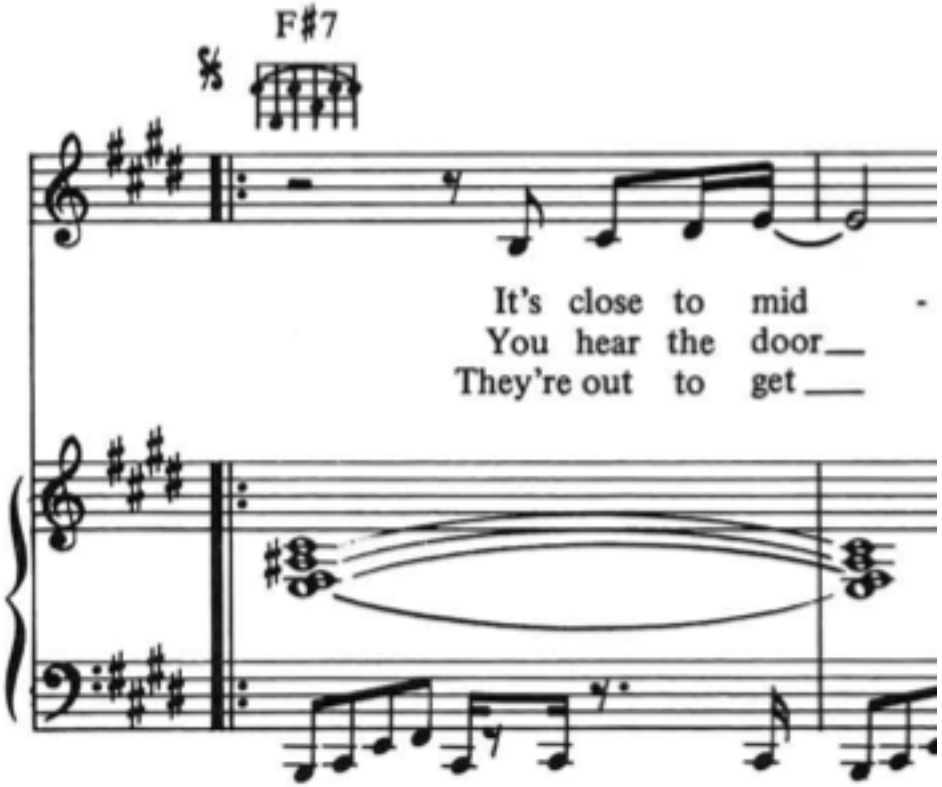


How Impro-Visor Works

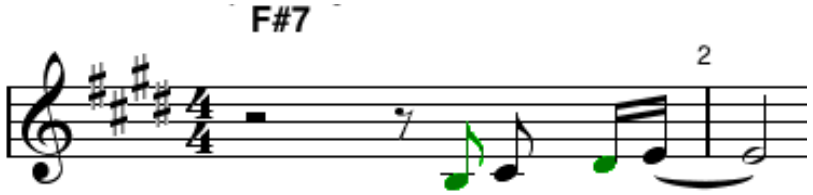
- Most musical information is in the form of user-editable text files:
  - **Vocabulary**, defines

    Scales, Chords, Cells, Idioms, Licks, Quotes
  - **Styles**
  - **Grammars**
  - **Leadsheet**, specifies
    - Chord progression
    - Melody, solo

Leadsheet vs. Sheet Music



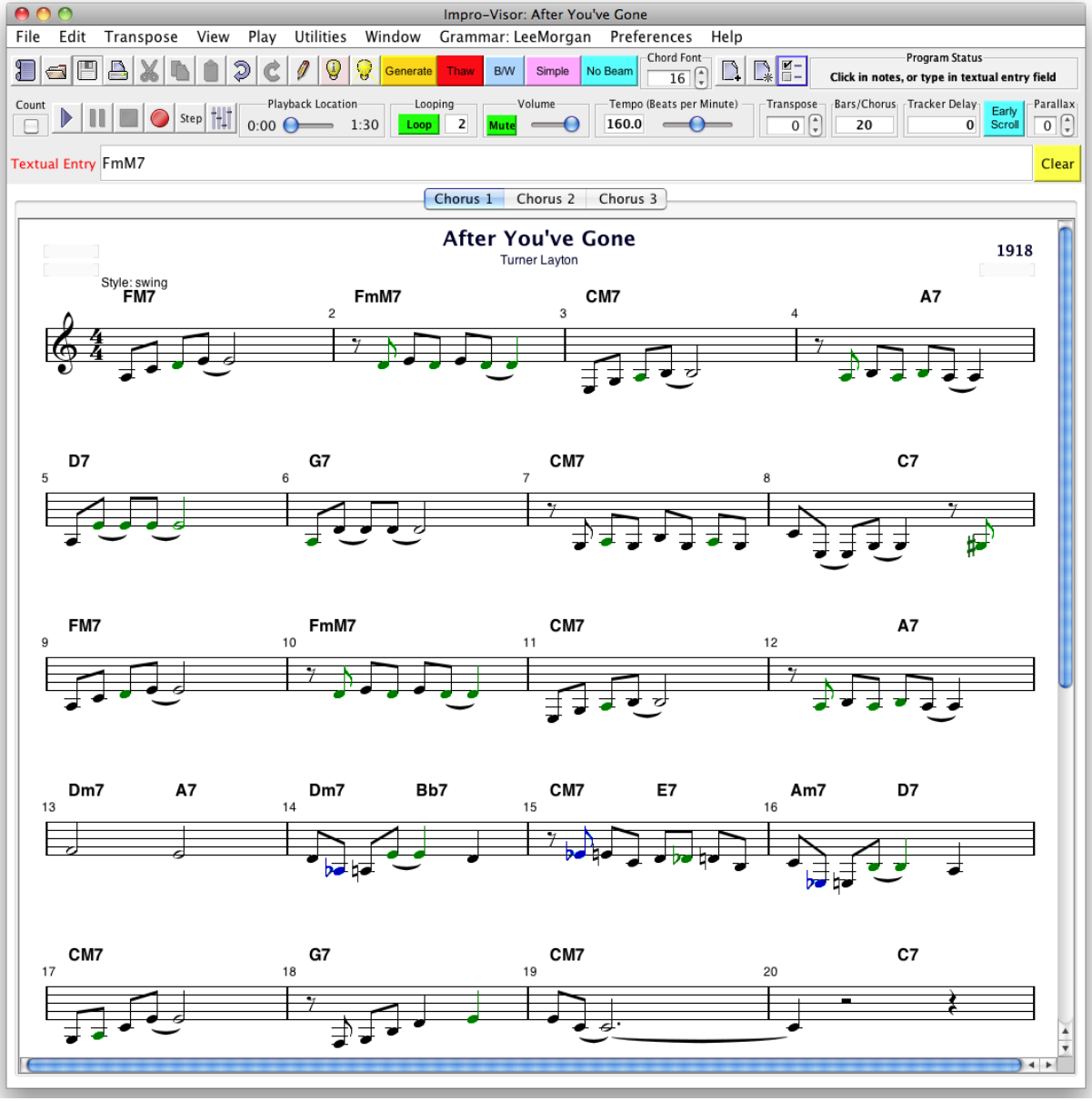
1 bar of **sheet music**



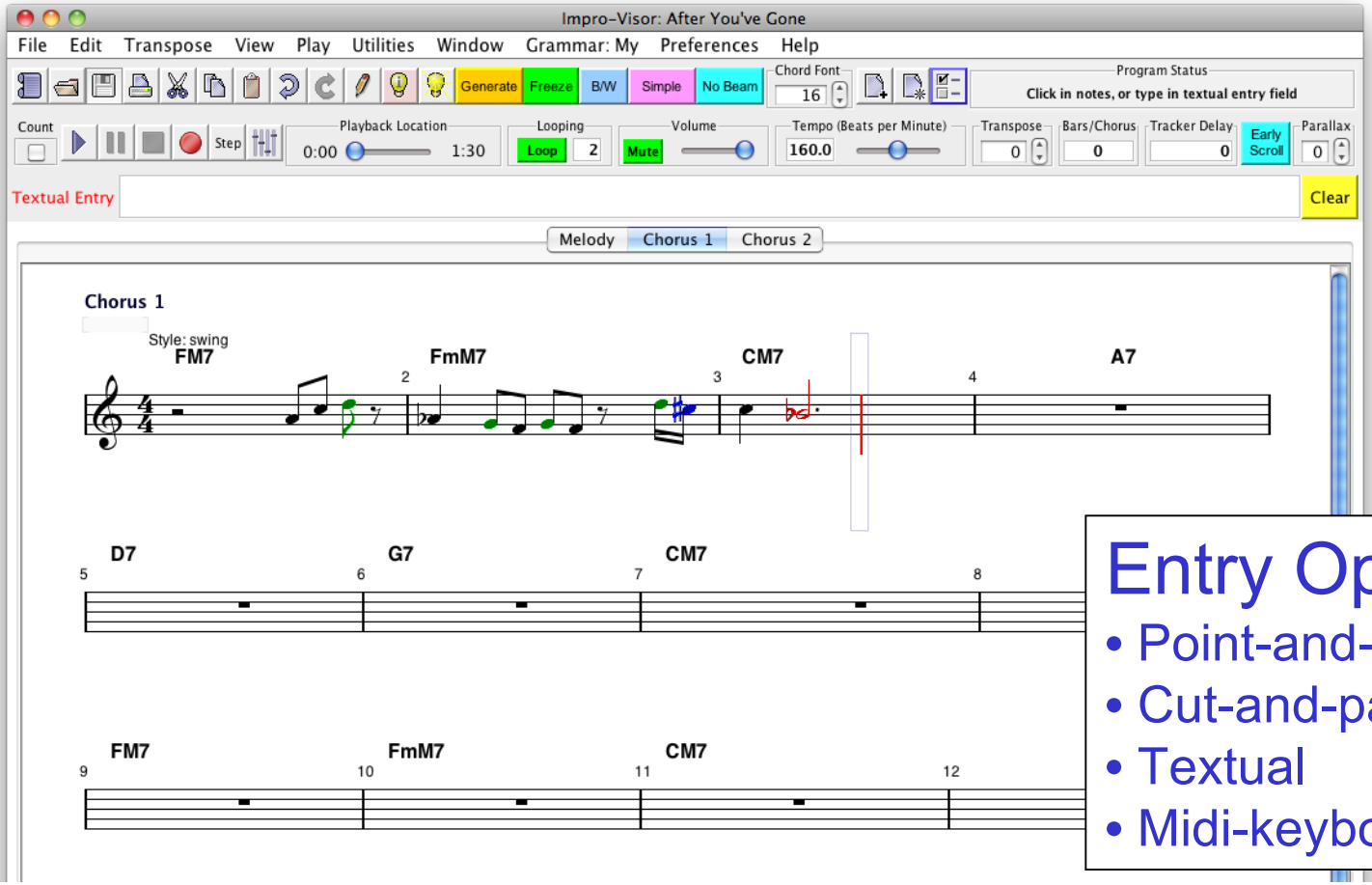
1 bar of a **leadsheet**

The accompaniment is left to the performer.

Impro-Visor's Leadsheet View



Constructing a Solo



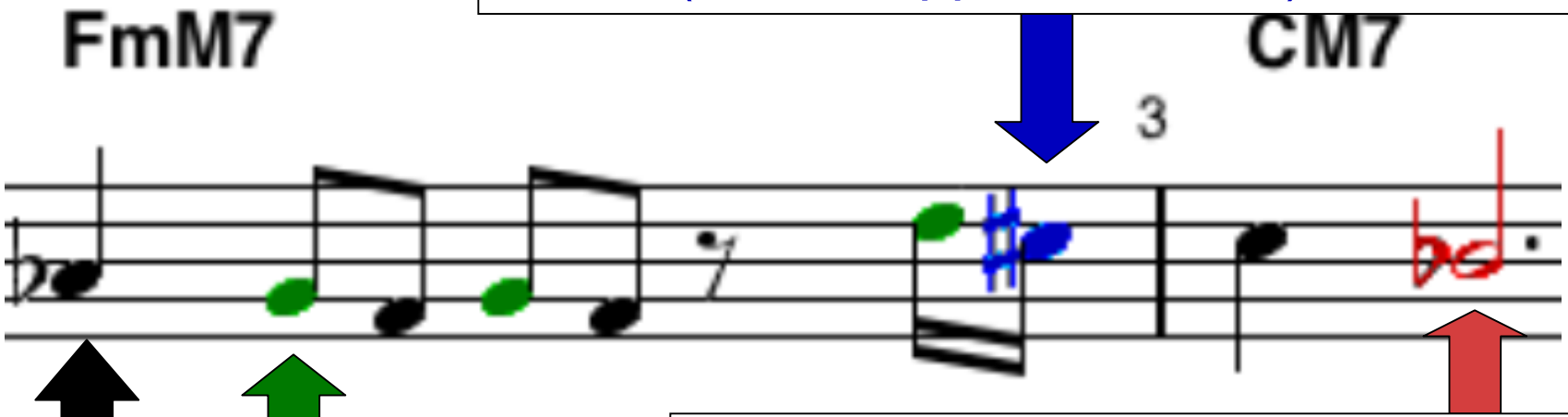
Entry Options:

- Point-and-click
- Cut-and-paste
- Textual
- Midi-keyboard

Four Note-Color Significance

(also used for other concepts later)

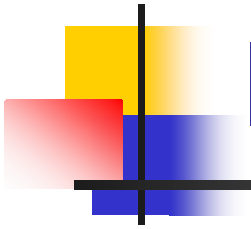
Blue: Half-step away from chord or color (called “approach” tone).



Red: None of the others (“outside”).

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

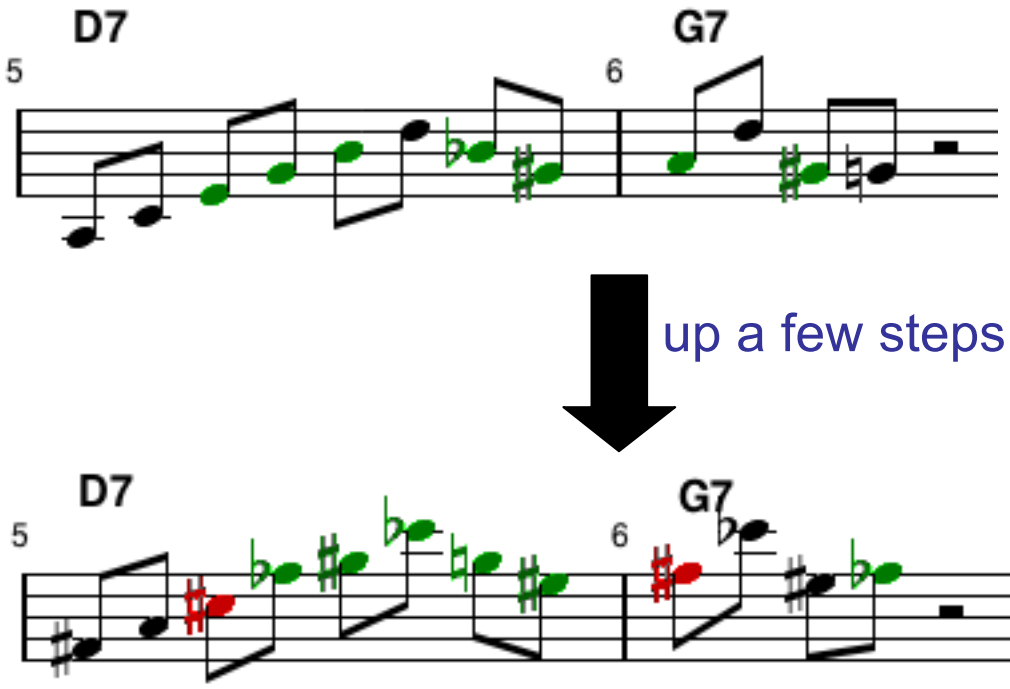
Black: tone in the chord



Intelligent Note-Entry Advice

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

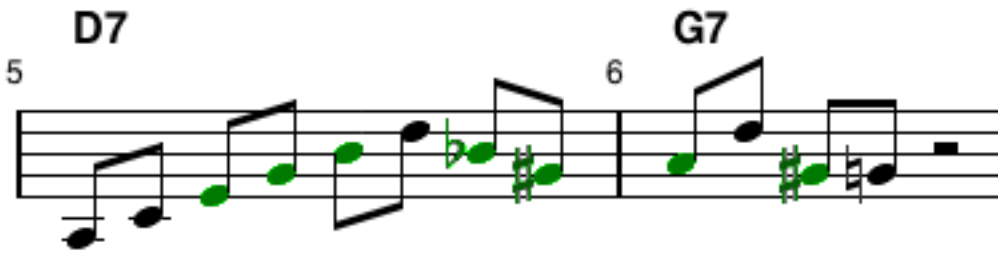
Ordinary (Uniform) Transposition



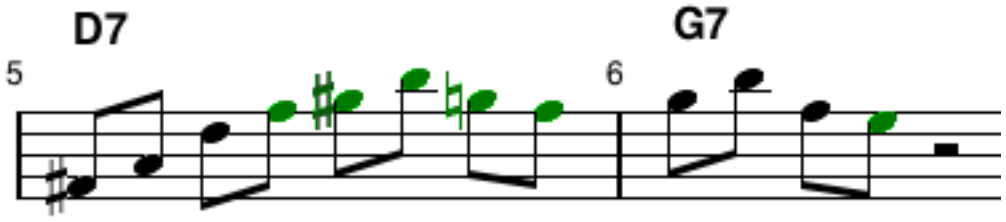
No discordant notes

Some discordant notes

Harmonic Transposition



up a sixth approximately

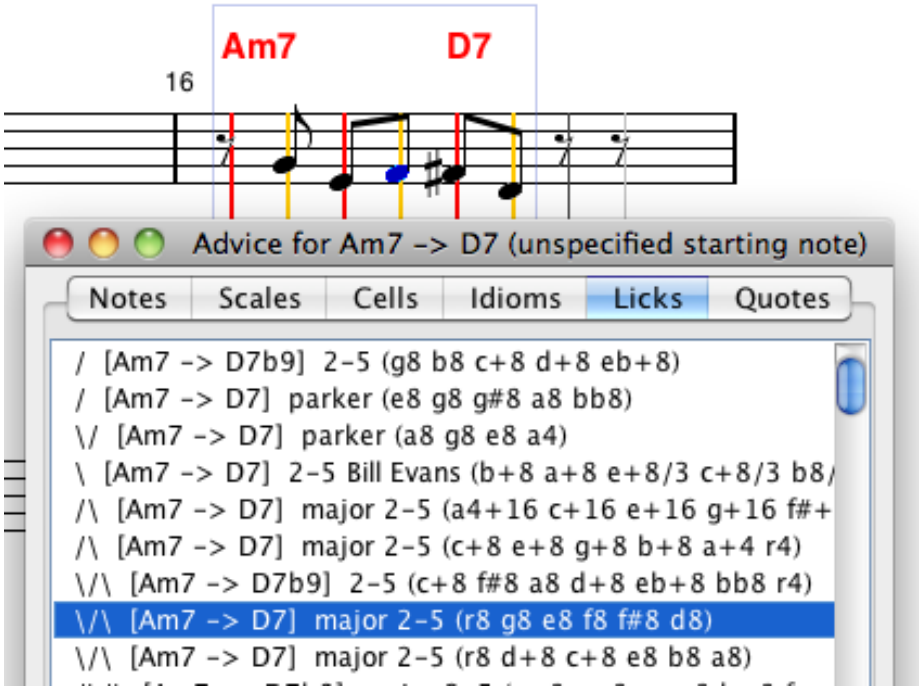


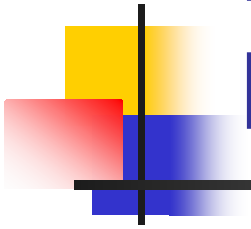
No discordant notes

Generating Licks for Examples

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

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



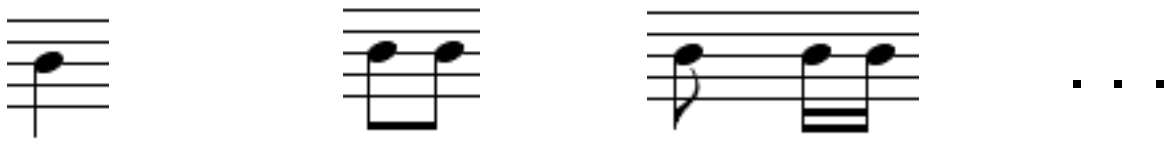


Lick Generation Uses a **Probabilistic Grammar**

- Grammars are a **generative specification**, typically for languages:
  - natural language
  - programming language
  - graphical language
  - musical language
    - melody
    - harmony
- Typical use of grammars in software is **analytic**.
- Impro-Visor, and other music software, use a grammar **generatively**.

Grammar Illustration

- We could fill a beat with a variety of rhythms:



. . .

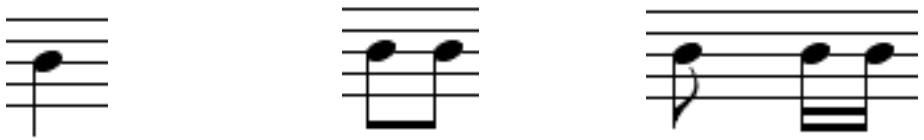
- Let B denote one beat of music
- A grammar represents all of these possibilities:

B → X4	4 means quarter note
B → X8 X8	8 means eighth note
B → X8 X16 X16	etc.

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

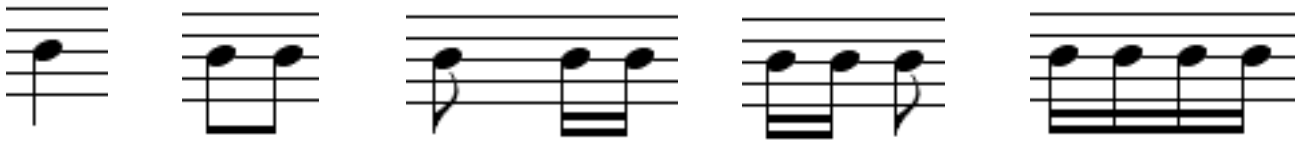
- Instead of

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

- Use

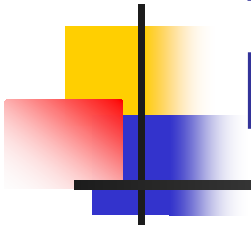
B → X4	p = 0.3	common
B → C C	p = 0.7	frequent
C → X8	p = 0.8	very frequent
C → X16 X16	p = 0.2	rare

- 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

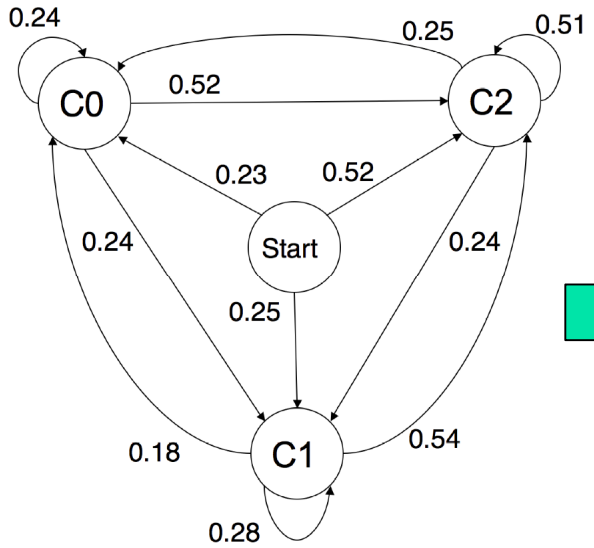


- R → B R One beat, then more
- R → empty No more

- So R can produce B, BB, BBB, BBBB, etc.

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.

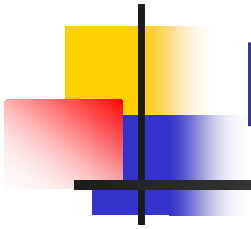


Markov chain

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

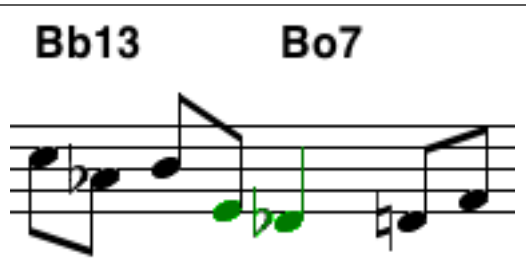
Abstract Melodies Based On Note Categories (“colors”)



- 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, rather than a specific one.

Abstract vs. Real Melodies

A real melody on the staff and as text



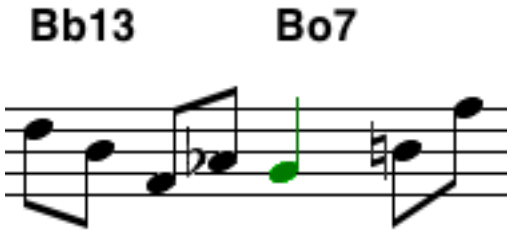
c+8 ab8 bb8 e8 db4 d8 f8

Abstract melody as text: C8 C8 C8 C8 L4 C8 C8

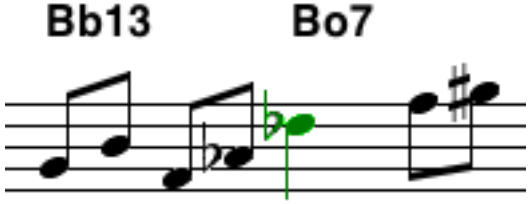
C = **C**hord tone
L = co**L**or tone
8 = 8th note
4 = quarter noter

Two other melodies with the same abstract melody

d+8 bb8 f8 ab8 g4 b8 f+8

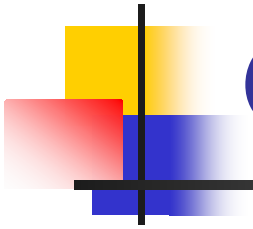


g8 bb8 f8 ab8 db+4 f+8 g#+8



A Complete Grammar "My Fours" with Terminals in Bold

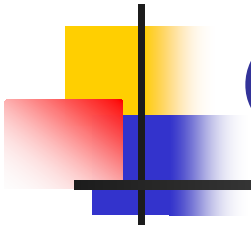
(startsymbol P)
(base (P 0) () 1.0)
(rule (M4) (**A4**) 0.01)
(rule (M4) (**L4**) 0.2)
(rule (M4) (**S4**) 0.1)
(rule (M8) (**A8**) 0.01)
(rule (M8) (**C8**) 0.4)
(rule (M8) (**L8**) 0.2)
(rule (M8) (**S8**) 0.1)
(rule (N2) (**C2**) 1.0)
(rule (N4) (M4) 0.75)
(rule (N4) (**R4**) 0.25)
(rule (N8) (M8) 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 and educational, but tedious.

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

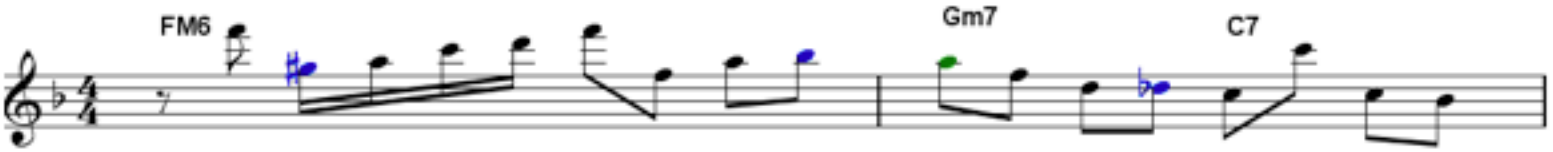


Grammar Learning Feature

- 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 in the place of terminals.

Slopes Encode Contours

(a) Original melody:



(b) Melody contour:

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

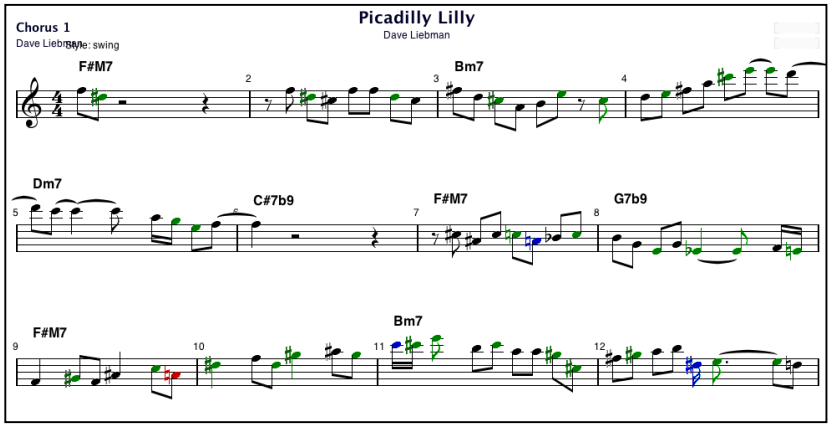
(R8 C8 (Δ -9 -9 A16) (Δ1 3 C16 C16 C16 C8) (Δ-12 -12 C8) (Δ1 4 C8 A8)

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

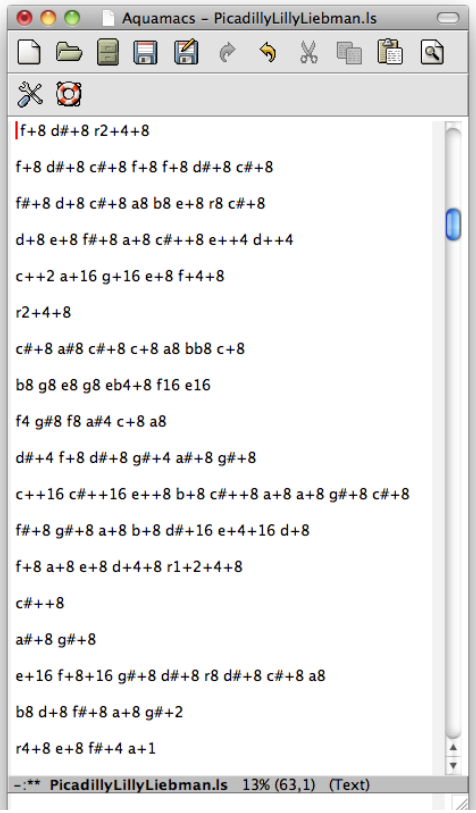
Grammar Learning Algorithm

enables grammar to be learned from transcriptions

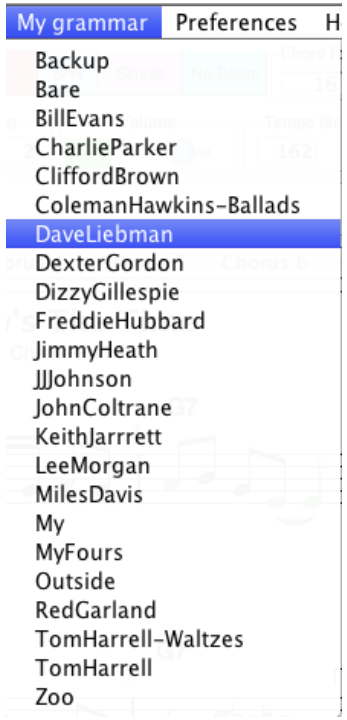
Transcription of Dave Liebman's Solo on Picadilly Lilly:



Transcription in Text



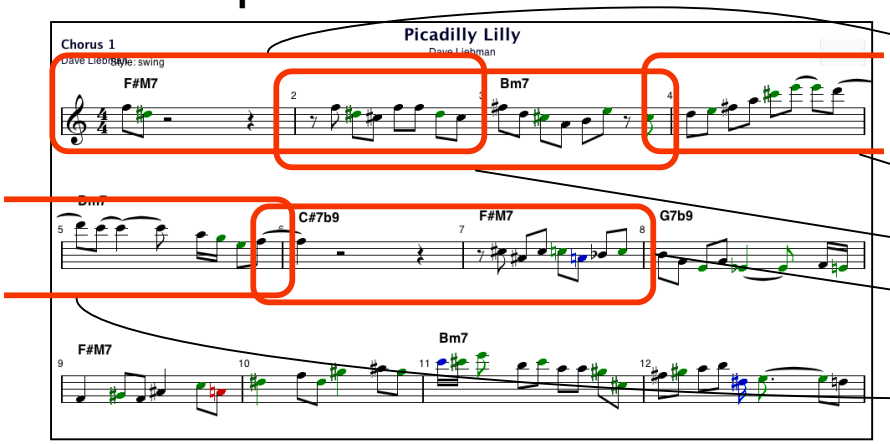
Learned Grammars



From Transcription to Grammar

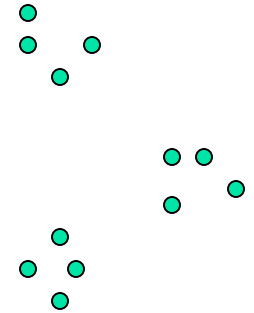
Transcription

2-bar melody **windows** in context of chords

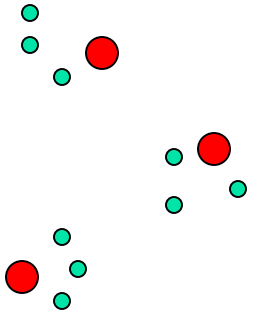


Abstract melodies

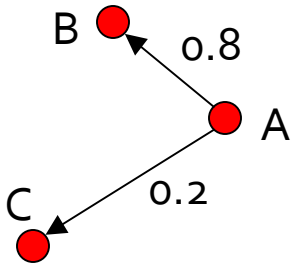
Abstract melodies **clustered** by similarity metric



Cluster **representatives** chosen



Representatives **Markov-Chained**



Grammar

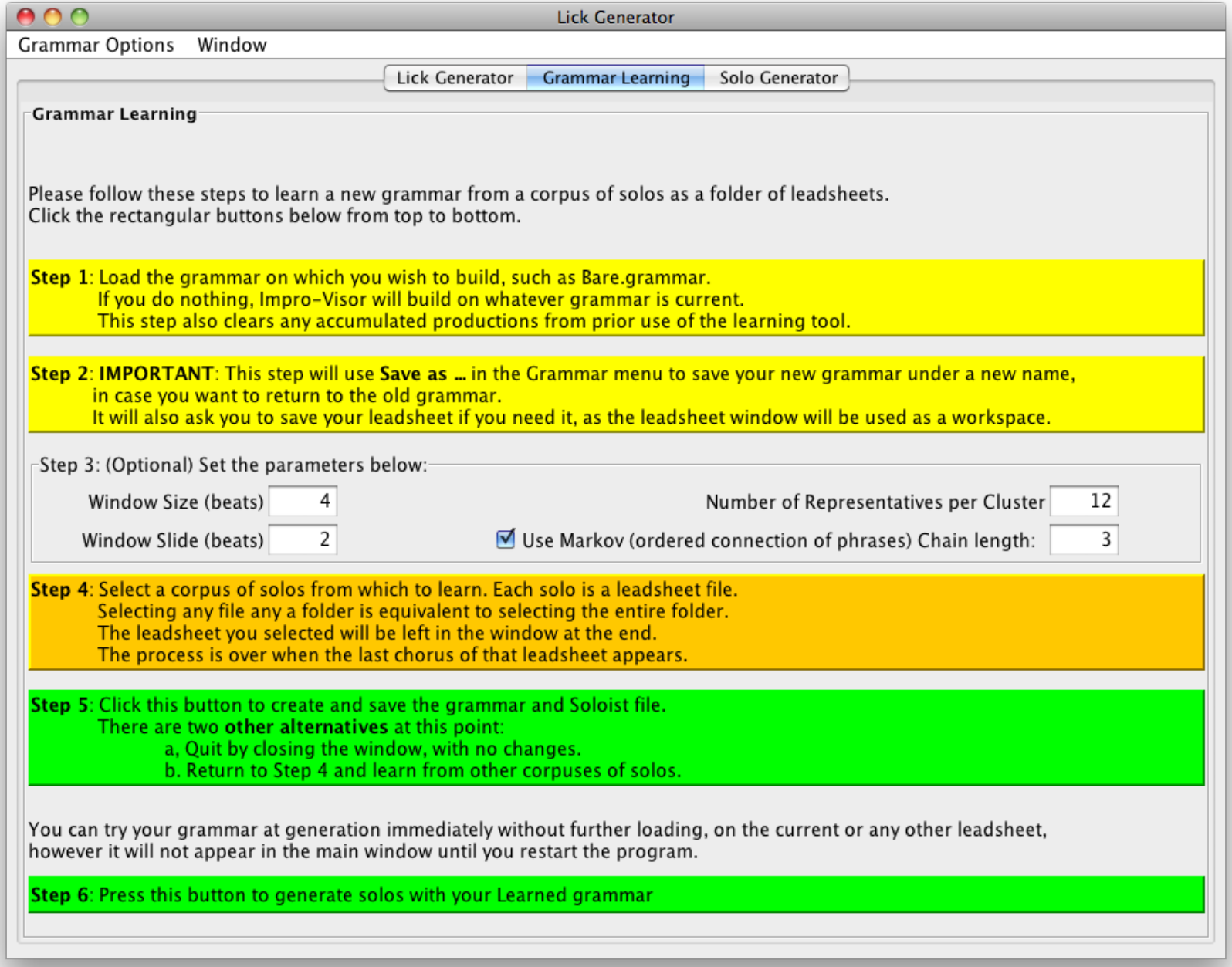
$A \rightarrow A_0\ B \quad (p = 0.8)$

$A \rightarrow A_0\ C \quad (p = 0.2)$

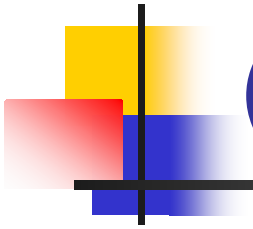
$B \rightarrow B_0 \quad (p = 1)$

$C \rightarrow C_0 \quad (p = 1)$

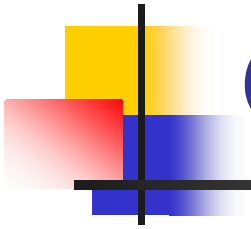
Impro-Visor's Grammar Learning Interface



A Blind-Evaluation Experiment (Gillick, Tang, and Keller)



- 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.



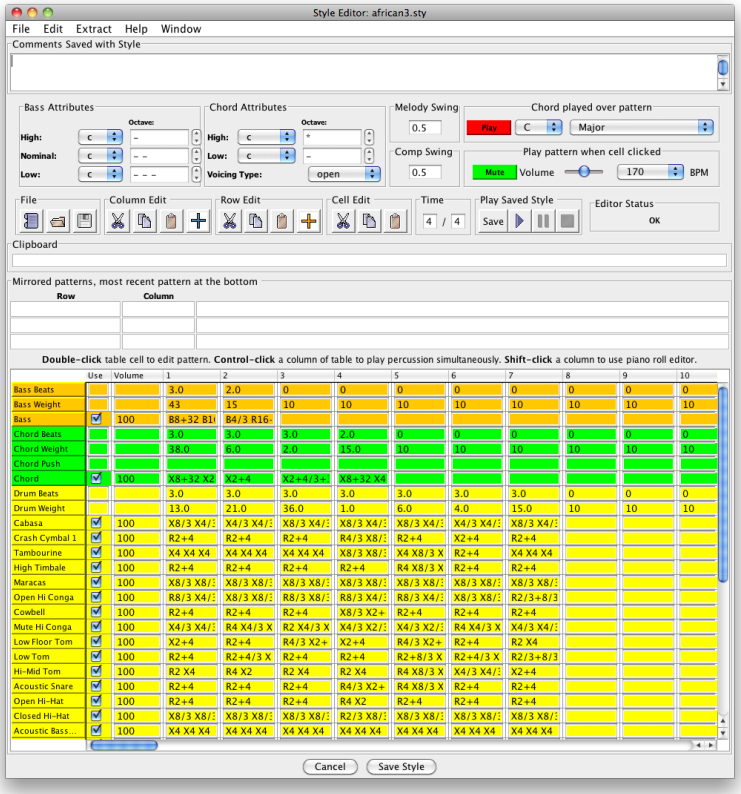
Other Learning in Impro-Visor

- 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** (needed for understanding bass patterns).
- As with grammar learning, clustering is used.

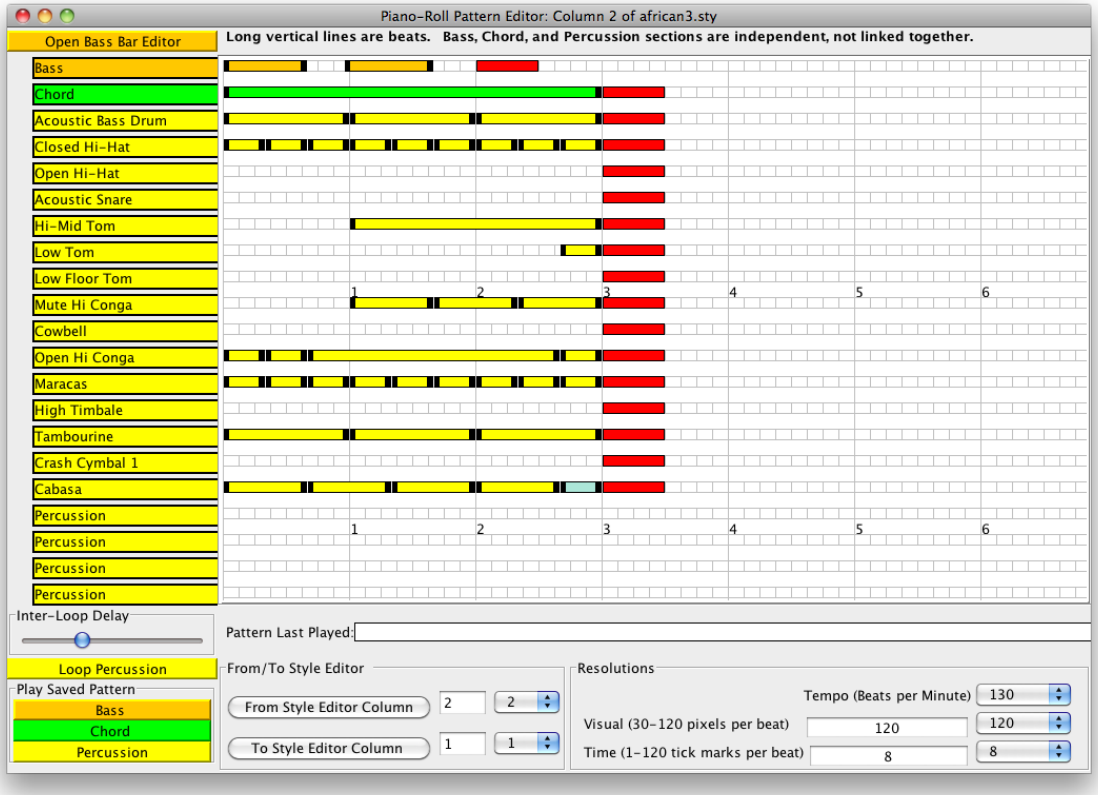
- A research problem is to **eliminate the second requirement**, that chords be provided.

Style Patterns Represented Graphically

Style Spreadsheet



"Piano roll" for one column of spreadsheet



Creativity, Emotion

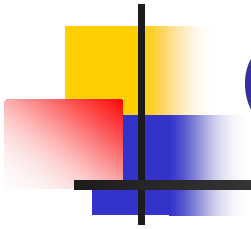
additional traits that we might desire intelligent software to exhibit

Emerging Academic Area: Computational Creativity

- Computers create, or help humans better create:
  visual art, music, stories, jokes, …
- 10 years of workshops, leading to

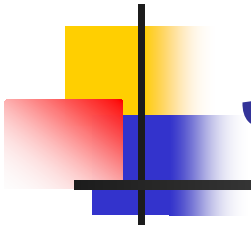
  International Conference on Computational Creativity (ICCC):

  - Lisbon, 2010
  - Mexico City, 2011
  - Dublin, 2012



Creativity Evaluation

- Anna Jordanous (DPhil, U. of Sussex, 2011) consulted six judges who individually evaluated creativity of
  - GenJam
  - Impro-Visor
  - George Lewis' Voyager program
  - Her own genetic system



Jordanous, 2011 summarizes:

"Impro-Visor was considered the system with highest value and again it had a good ability to create results.

Much poorer scores were recorded for Impro-Visor's ability to develop its improvisations and to express emotions and intention; this last point was prioritised by survey participants alongside more expected abilities such as domain expertise and the ability to communicate and interact with other musicians and the audience."

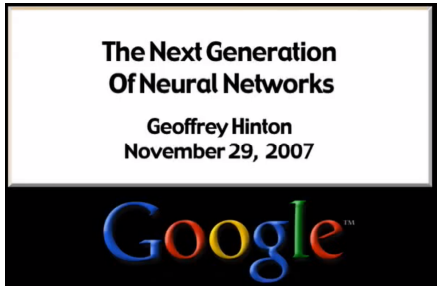
A Different Approach to Learning: **RBM-provisor**

- Problem: How to learn to improvise with **minimal musical knowledge**.
- We applied **Restricted Boltzmann Machines** (RBMs) in the form of ***Deep Belief Networks***.
- 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

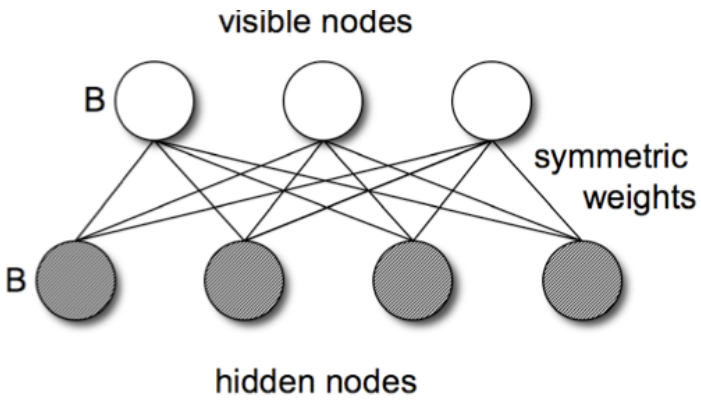


- We applied a similar idea to learning concepts that produce melodies from chord progressions.

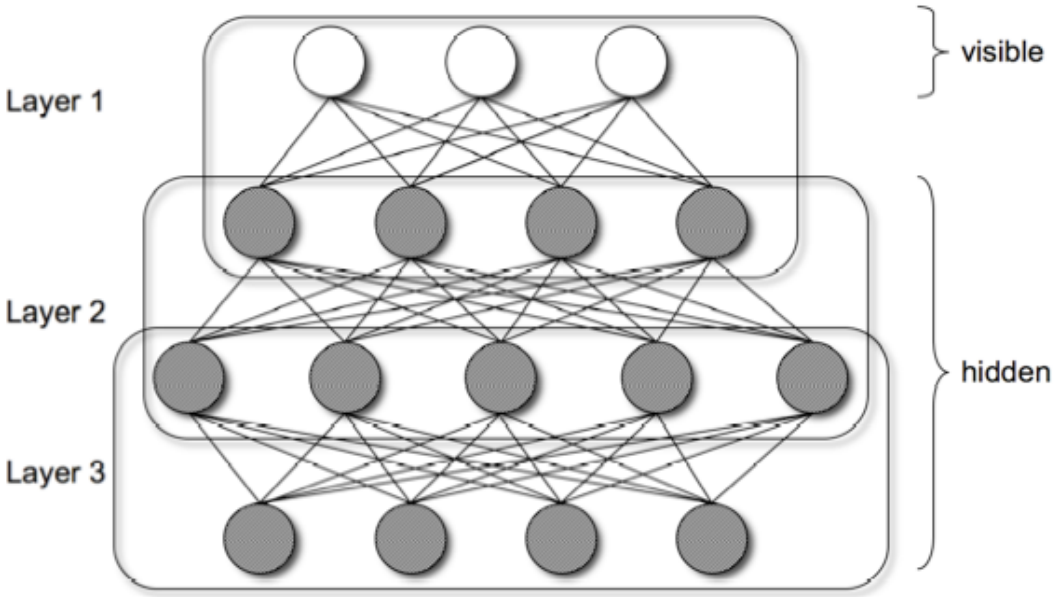
- The idea was to **build in as little** musical knowledge as possible.

Restricted Boltzmann Machines vs. Deep Belief Networks

RBM

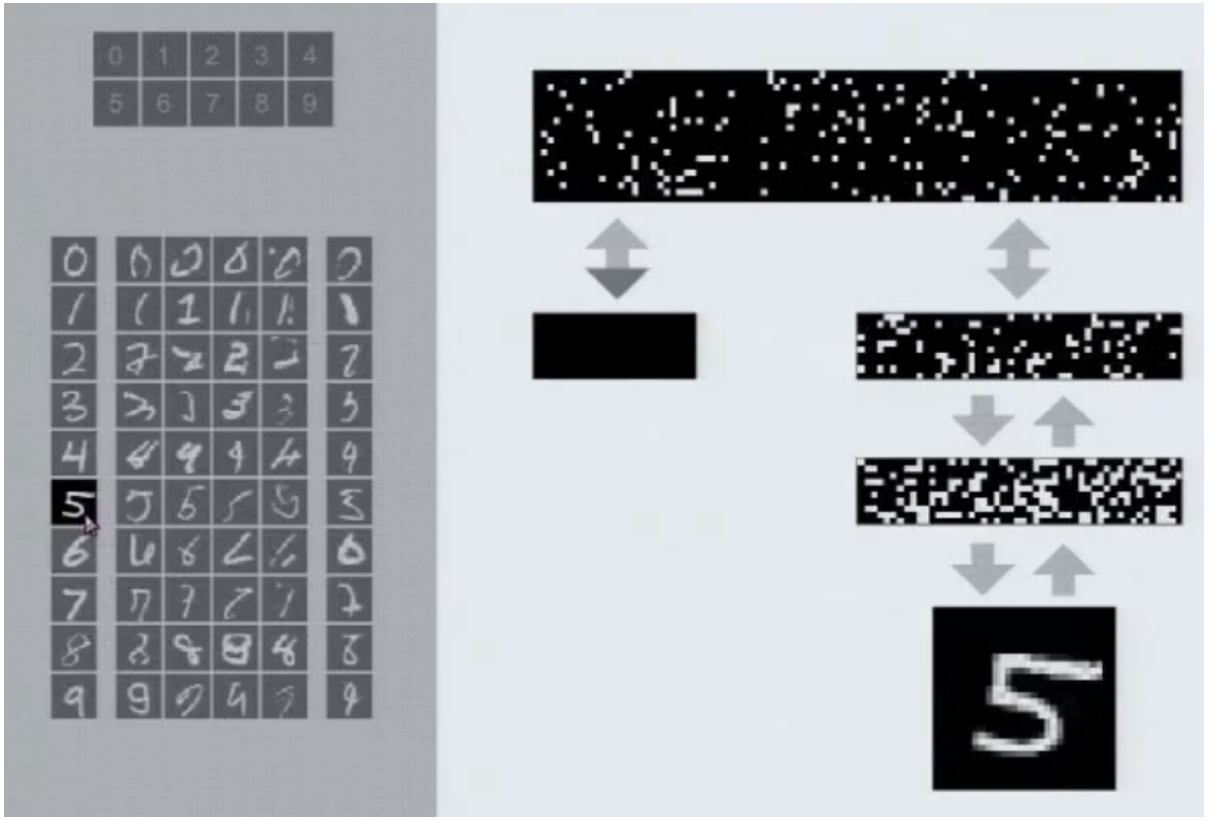


DBN (3-layer)



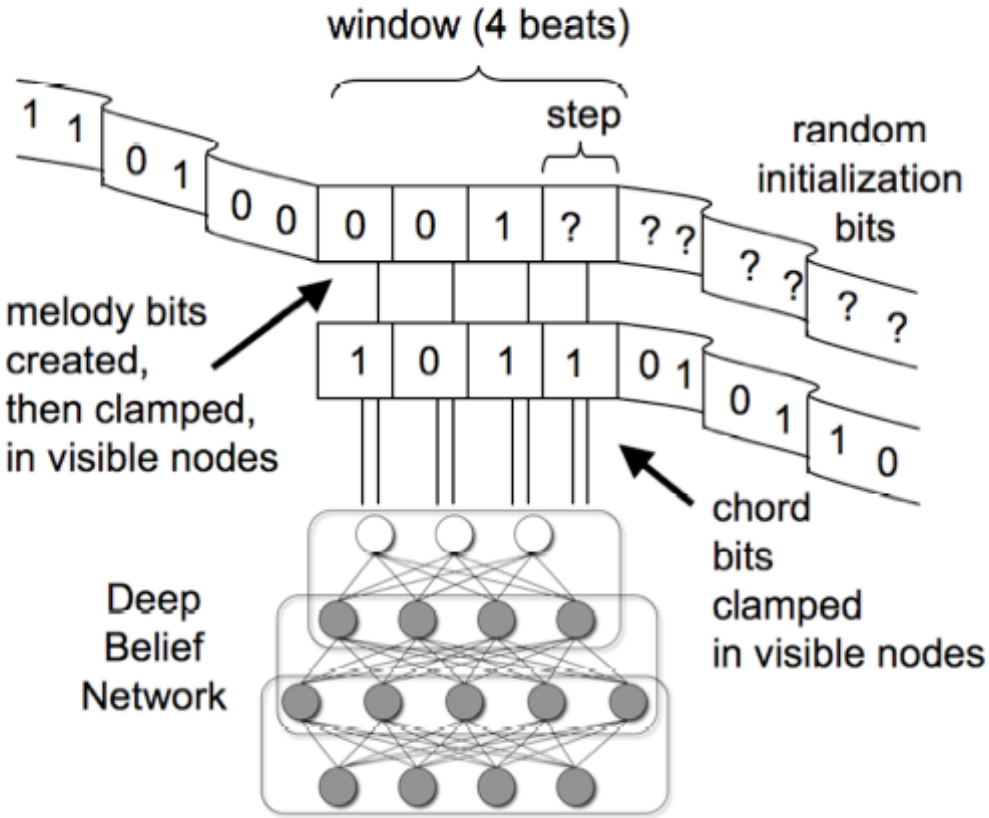
Deeper Concepts

Hinton's approach to pattern learning and generation



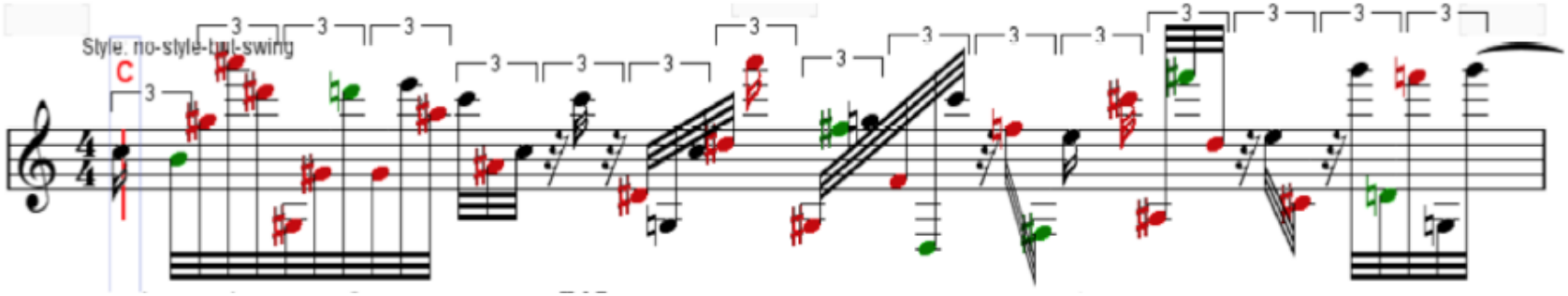
Improvising Jazz with a Deep Belief Network

Melody Out ⟵ Chord Progression In

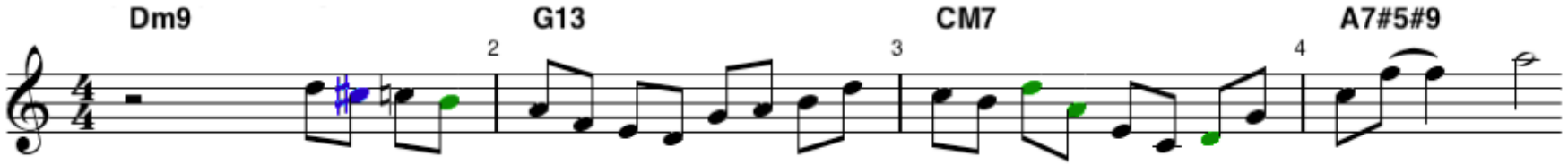


RBM-provisor Examples

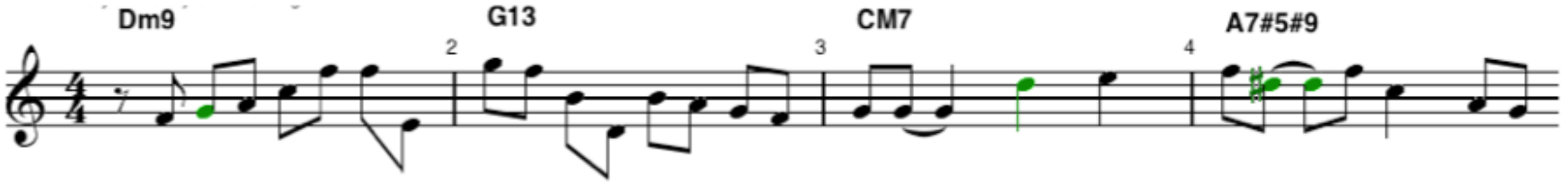
Output from Unlearned Network

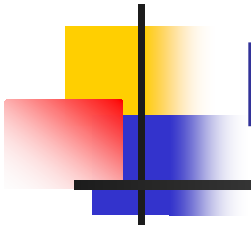


Example from Training Set



Output from Trained Network

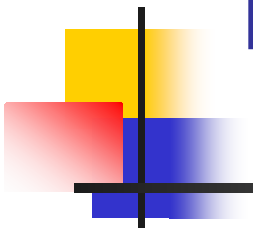




Recent Work

- Automate analysis to representing and manipulating idiomatic harmonic sequences (“chord bricks”) and key centers.
- Helps musicians understand tune construction.
- Helps players recognize the importance of key centers in improvisation.

Road Maps



- Road maps are Impro-Visor's newest feature.
- Tunes are automatically analyzed into keys and "bricks".
- Bricks are idiomatic chord progressions, such as cadences and turnarounds.
- The idea of bricks is from pianist/author Conrad Cork in the U.K.
- Bricks help the beginner to intermediate player understand the tune.

Analyzing a Tune using Bricks

Input Chord Progression

(phrase)
Cm69 | / | Fm7 | / |

(section)
Dm7b5 | G7alt | Cm69 | / |

Ebm7 | Ab7 | DbM7 | / |

Dm7b5 | G7alt | Cm69 | Dm7b5 G7alt |

Impro-Visor

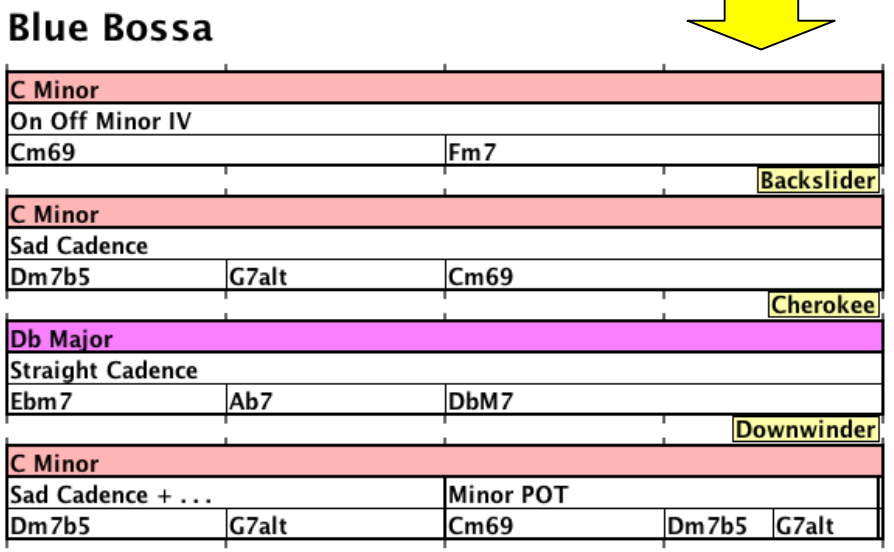
Output Roadmap

Join Names (yellow tags)

Input Chords

Inferred Key

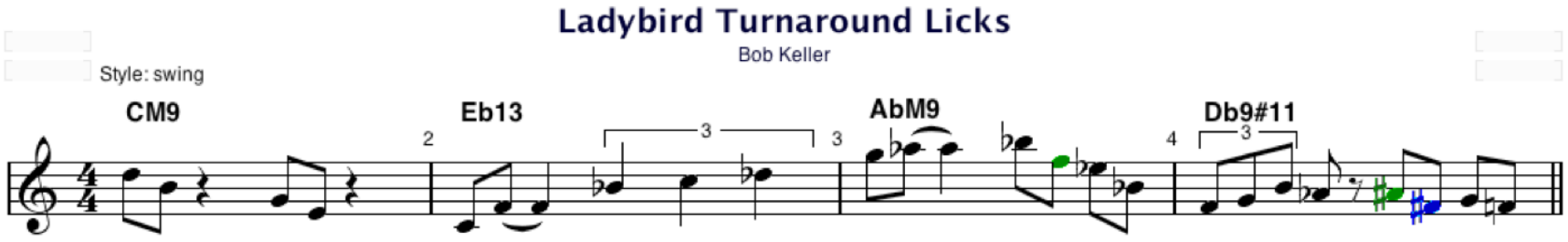
Brick Name

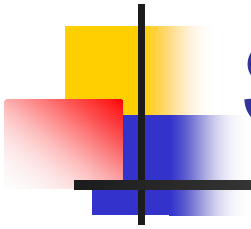


Future Work:
Bricks as a Basis for Grammar Learning

See: http://www.cs.hmc.edu/~keller/jazz/improvisor/licks/

Name	Chord Progression			
Cadence + Dropback	IIm^7	V^7	I	VI^7
Sad Cadence + Dropback	IIm^{7}b5	V^7	Im	VIm^{7}b5
POT (*Plain Old Turnaround*)	I	VI^7	IIm^7	V^7
Minor POT (*Minor Plain Old Turnaround*)	Im	VI^7	IIm^{7}b5	V^7
Pullback	IIm^7	V^7	IIIm^7	VI^7
Ladybird Turnaround	I	bIII^7	bVI	bII^7





Some References

- Keller, Hunt, Jones, Morrison, Wolin, and Gomez, **Blues for Gary: Design Abstractions for a Jazz Improvisation Assistant**, *ENTCS (Electronic Notes in Theoretical Computer Science)*, 193 (2007) 47-60.

- 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.

Some Links Related to This Talk

- This Talk: http://www.impro-visor.com/IEEE or http://www.impro-visor.com/IEEE6up
- http://en.wikipedia.org/wiki/Free_and_open_source_software FOSS
- http://audacity.sourceforge.net/about/ Audacity
- http://en.wikipedia.org/wiki/Audio_file_format Audio formats
- http://www.smartmusic.com/ Smart Music
- http://www.digital-ear.com/digital-ear Digital Ear
- http://www.intelliscore.net/download.html Intelliscore
- http://www.celemony.com/cms/ Melodyne
- http://www.pgmusic.com/ Band-in-a-Box
- http://www.ist.rit.edu/~jab/GenJam.html GenJam
- http://www.music.informatics.indiana.edu/~craphael/music_plus_one/index.html Plus One
- http://www.miller-mccune.com/culture/triumph-of-the-cyborg-composer-8507/ Emily Howell
- http://artsites.ucsc.edu/faculty/cope/ David Cope
- http://www.csl.sony.fr/downloads/papers/2002/pachet02b.pdf Francois Pachet
- http://www.youtube.com/watch?v=AyzOUbkUf3M Geoffrey Hinton
- http://kcl.academia.edu/AnnaJordanous/Papers/859375/Evaluating_Computational_Creativity_A_Standardised_Procedure_for_Evaluating_Creative_Systems_and_its_Application Anna Jordanous
- http://www.impro-visor.com Impro-Visor