A Knowledge Intensive Approach to Machine Learning in Music

Gerhard Widmer
Department of Medical Cybernetics and Artificial Intelligence,
University of Vienna
and
Austrian Research Institute for Artificial Intelligence,
Vienna

E−mail: gerhard@ai−vie.uucp

Abstract

The article describes the first results of an ongoing project that is being pursued at the Austrian Research Institute for Artificial Intelligence. The long−term goal of the project is the development of a new generation of flexible and adaptive musical systems. The central concern is therefore with techniques of Machine Learning, and with research into the role that general musical knowledge plays in a system that is to learn new musical concepts. The first test domain for our system was two−voice counterpoint composition. The starting point for the project was the realization that 'intelligent' learning requires a considerable amount of domain−specific knowledge. We will describe our approach to defining some basic knowledge about tonal music which can serve as the basis for learning processes. We will also briefly describe the integrated learning strategy which can take advantage of such knowledge during learning. Finally, the paper also suggests that intelligent, knowledge−based learning systems could be useful tools for testing general theories about music.

1 Introduction

This article describes a project whose long−term goal is the development of a new generation of flexible and adaptive musical systems. Flexibility requires the ability to learn. Thus, for the moment we concentrate on the development of learning methods that are suited to the task of learning musical concepts and rules. The first test domain for our system is two−voice counterpoint composition.

Two main ideas will be expounded in this paper: the first is that 'intelligent' learning requires a considerable amount of domain−specific, but possibly abstract, knowledge, that is, knowledge about the *structure* of the domain whose concepts are to be learned. This leads us to the problem of identifying and formalizing what we consider basic musical knowledge. It will be argued in this article that there is indeed something like common musical 'knowledge' shared by most Western music listeners, and a model that captures some of this knowledge (with respect to two−voice counterpoint) will be presented. The second problem then consists in devising learning algorithms that can take advantage of that knowledge to learn both more 'intelligently' and more effectively. We will present a new learning algorithm that has this desired property. Some examples of the algorithm at work will also be presented.

The second underlying idea is not new, but may not yet have been considered in connection with machine learning: we see intelligent learning systems as extremely useful tools for testing general theories in many domains. For a theory of perception of tonal music this means that the theory can be tested by experimenting with a learning system and analyzing how certain assumptions and *a priori* knowledge affect the 'learnability' of musical constructs and rules. This could lead to new insights concerning the connection between assumptions about perception and specific systems of musical rules or styles.

The article tries to address two groups of researchers with possibly very different areas of interest and equally different background, namely, specialists in Artificial Intelligence (and Machine Learning, in particular), and musicologists. It seems therefore necessary to give a short introduction to those concepts in the field of Machine Learning that are relevant to our project, in particular the concept of 'knowledge–based' learning. As our first application – two–voice counterpoint – is a very restricted and comparatively simple (some might say artificial) musical problem, we assume that the reader is familiar with the basic rules and concepts of that domain.

2 Machine Learning: some important concepts

Machine Learning (ML) is the discipline that tries to develop theories and models of learning processes, and to build working computer systems that can learn and adapt to new situations. 'Learning' is a very broad term; in principle, it covers adaptation to new environments and situations, building new concepts and relations from experience and observation, the acquisition of knowledge from teachers and textbooks, the acquisition of specialized skills through practice, learning to avoid mistakes, and many more such scenarios. In the field of Machine Learning, the idea of learning *general rules or concepts* from specific *examples* has received by far the most interest. If we define the notion of a 'rule' or 'concept' broadly enough, this research direction covers many of the learning scenarios listed above. The system to be presented here will learn general *problem solving rules* from specific instances of correct and incorrect counterpoint compositions.

2.1 Empirical vs. analytical learning

The field of Machine Learning has seen a major shift of interest in recent years. While in the early times of ML, most work went into the development of *inductive, empirical* learning algorithms, where a system is expected to learn some concept solely from positive and negative examples (see, for instance, Dietterich & Michalski 1981; Michalski 1983; Mitchell 1982; Winston 1975), the past few years have brought a tendency towards *knowledge–based*, i.e, essentially *deductive*, *analytical* learning.

The main motivation for this shift lies in the problems that are inherent to *logical induction*. Inductive generalization from examples is problematic for at least two reasons. First, it is not logically justifiable: we have no guarantee that an empirical generalization drawn from the observation of a finite number of examples of a concept will be correct. Coincidental similarities between the limited number of known examples may lead a learning system to consider lots of incorrect or non–sensical generalizations that have nothing to do with the concept to be learned. Given just the examples and no other information, a learner has no way of judging the plausibility or correctness of a generalization. A related problem is that of the search space: the number of possible generalizations is huge for any non–trivial concept description language (Rendell 1987), and 'blind' search of the space of generalizations is prohibitively expensive.

The solution to these problems lies in the use of *knowledge* to guide the learning process. There seem to be very few situations where people learn 'blindly', without trying to *explain* the observed phenomena with the help of some prior knowledge (or by way of analogy). Such explanations, if possible, both reduce the number of plausible generalizations and act as justifications of generalizations.

The main result that came out of the new interest in knowledge–based learning was a methodology known as *Explanation–Based Learning* or *EBL* (Mitchell *et al.* 1986; DeJong & Mooney 1986). Learning in EBL consists in *explaining* why and how an example belongs to the concept to be learned and *generalizing* this explanation. This process yields a description of a whole class of objects or situations which satisfy the same explanation structure and hence also belong to the goal concept. These explanations are in the form of *deductive proofs*. They are derived from the system's *a priori* knowledge about the problem domain; the sum total of this knowledge is called the system's *domain theory*.

The EBL method is the exact opposite of knowledge–free inductive learning in that it requires *complete knowledge* about the thing to be learned in order for learning to be possible. Learning is essentially reduced to re–expressing existing knowledge in a form that is more efficient and more directly applicable (or *operational*, in EBL terminology). While such an approach may be fruitful in some domains, it is clear that there are many more fields which do not permit the formulation of a complete *a priori* theory, and hence are not amenable to such purely deductive techniques. Music is but one of them. But even though EBL is not applicable in such domains, we do not want to resort to 'blind' inductive learning. What we are looking for are learning methods that use all the knowledge available, without requiring completeness and consistency of that knowledge.

In the field of music theory, some attempts at automatic learning of musical rules have already been made, but most of these projects were along the lines of purely inductive learning (e.g., induction of musical grammars from examples). We are pursuing an alternative methodology: we provide the system with basic musical knowledge (the kind of knowledge that is intuitively clear to us) and devise algorithms that can take advantage of that knowledge during the process of learning.

2.2 Learning Apprentice Systems

In order to give the reader an idea of the scenario in which our system learns, we briefly introduce the concept of *Learning Apprentice Systems*. Mitchell *et al*. (1985) define a Learning Apprentice as

"... an interactive knowledge−based consultant that directly assimilates new problem−solving knowledge by observing and analyzing the problem solving steps contributed by its users through their normal use of the system."

Two things to note here are that a human expert (teacher) is assumed to be present and that learning should occur, as far as possible, during *normal* use of the system, which means that the system should learn primarily by observing the problem solving steps of the expert. But if the learning system does not have a complete *a priori* theory, and if we want it to learn reliably, it will sometimes have to ask questions of the expert. By using domain−specific background knowledge in the process of interpreting the user's actions, the number of questions that have to be asked can be kept to a minimum, and the system can make many plausible generalizations without any help from the teacher.

The system we have constructed, then, is a Learning Apprentice for two−voice counterpoint composition, and the *a priori* knowledge we have endowed it with are the kinds of 'intuitive' perceptions that every ordinary person has acquired from years of (conscious or unconscious) exposure to tonal music.

3 The domain: Two−voice counterpoint composition

The problem for which the system is to learn rules is defined as follows:

Given: key, time signature, and melody (the *cantus firmus*) of a counterpoint piece.

Problem: complete the piece by writing a second line (the *counterpoint*) in such a way that the constraints of counterpoint style are satisfied.

Fig.1 shows a simple counterpoint piece of first species (whole notes against whole notes).

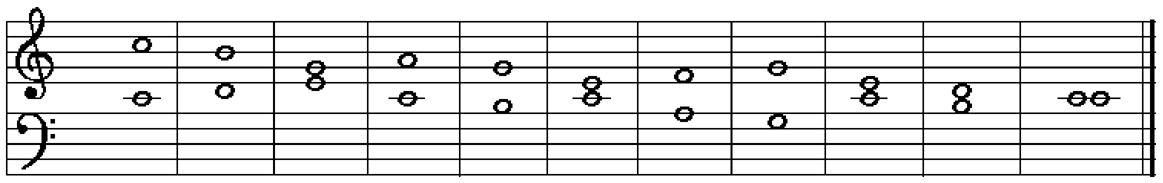


Fig.1: A simple counterpoint piece

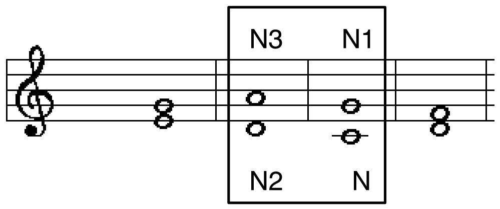
The system learns three classes of rules:

good (P,N) if <conditions>
  note N is a good solution in a particular context in piece P if <conditions> (a conjunctive expression) is satisfied. <conditions> specifies the structure of the context and the attributes that note and context must satisfy for N to be a good solution.

bad (P,N) if <conditions>
  note N is bad in a particular context in piece P if <conditions> hold.

unacceptable (P,N) if <conditions>
  note N is illegal in a certain context in piece P if <conditions> hold.

Fig.2 shows a simple rule and a musical situation to which it applies (succession of perfect consonances). Learned rules are ordinary PROLOG clauses; variables are capitalized; mnemonic variable names were substituted by the author.



```
unacceptable( Piece, Note) :–
   parallel_note( Note, ParNote),
   previous_note( Note, PrevNote),
   parallel_note( PrevNote, PrevParNote),
   is_interval( Note, ParNote, perfect_consonance),
   is_interval( PrevNote, PrevParNote, perfect_consonance).
```

Note = N
ParNote = N1
PrevNote = N2
PrevParNote = N3

Fig.2: A simple counterpoint rule

4 General musical knowledge as a basis for learning about music

In our system, the 'technical' basis for reasoning about music is a hierarchical knowledge base defining the basic musical concepts (notes, intervals, scales, keys, etc.) along with their intrinsic properties (pitch, duration, degree of consonance, etc.). The knowledge base is implemented as a hierarchical frame system.

Given this information and some examples of acceptable and unacceptable counterpoint pieces, the system could learn counterpoint rules *inductively*, i.e., by comparing the different examples and hypothesizing general rules on the basis of commonalities and differences. A large number of (carefully selected) examples would be necessary, and the system would come up with many nonsensical hypotheses.

But we can do better than that; surely nobody learns counterpoint without prior musical knowledge. In fact, every ordinary person 'knows' a lot about music (sometimes without knowing it) just from having been exposed to music (or Muzak, as the case may be) for years. This kind of tacit knowledge is usually called 'habits of perception' or 'musical intuition', or sometimes simply the 'musical ear'. It is this general knowledge that we want to model and give to the program as a basis for the learning process. We want to provide the computer with an 'ear', as it were, so that it has the same possibilities for learning as humans.

4.1 A hierarchical model of perception: Events and effects

We have devised a simple (and rather naive) model of the perception of two−voice counterpoint to serve as a basis for the formulation of simple musical knowledge (see Fig.3). The model rests on the notions of *events* and *effects*.

4.1.1 Events: the structure of musical situations

At the moment we are experimenting with the simplest species of counterpoint, i.e., whole notes against whole notes, so we can use a rather simple model of the structure of a piece. The lower half of Fig.3 depicts successively more abstract views of a musical situation: level 1, the level of the individual notes, is not adequate as a reasonable basis for the description of situations; it is too unstructured. It seems safe to postulate that people make at least very simple *local abstractions* when listening to music: a note can be perceived as a single entity in its own right; two simultaneous notes are often heard as a unit, a *vertical interval*, and two consecutive notes can be perceived as forming a unit of melodic motion, a *step* or *leap*. How we perceive notes depends on the musical context and on the phenomena we are concentrating on. Accordingly, our representation system rests on three types of *events* (level 2):

- *single note* (a single note is viewed as such)
- *vertical interval* (two simultaneous notes are represented as a unit)
- *step/leap* (two consecutive notes are viewed as a unit)

Any situation in a piece can then be represented as a composition (sequence or simultaneity) of events and any combination of these compositions (level 3). Because of the multiple role of single notes (atomic event as well as part of some composite events), more than one parsing (or *view*) of a situation is usually possible. Our representation language is a context−free grammar, very similar to Stephen Smoliar's tree representation for Schenkerian analyses (Smoliar 1980). When applied to a collection of specific notes, it returns a typed parse tree for each possible structural view of this musical situation.

Instead of listing the entire grammar, let us demonstrate the principle by way of an example: Fig.4 shows a simple situation in a piece. The four notes b, d, g, and e can be interpreted as a sequence of vertical intervals, a pair of two simultaneous horizontal steps/leaps, or any combination of these with single notes. This is expressed in the parse trees of Fig.4. The labels composite_situation, sequence, vertical_interval etc., correspond to non−terminal symbols in the grammar and specify the *type* of a subexpression. Strongly typed expressions permit clean formulation of rules and background knowledge: for every rule or

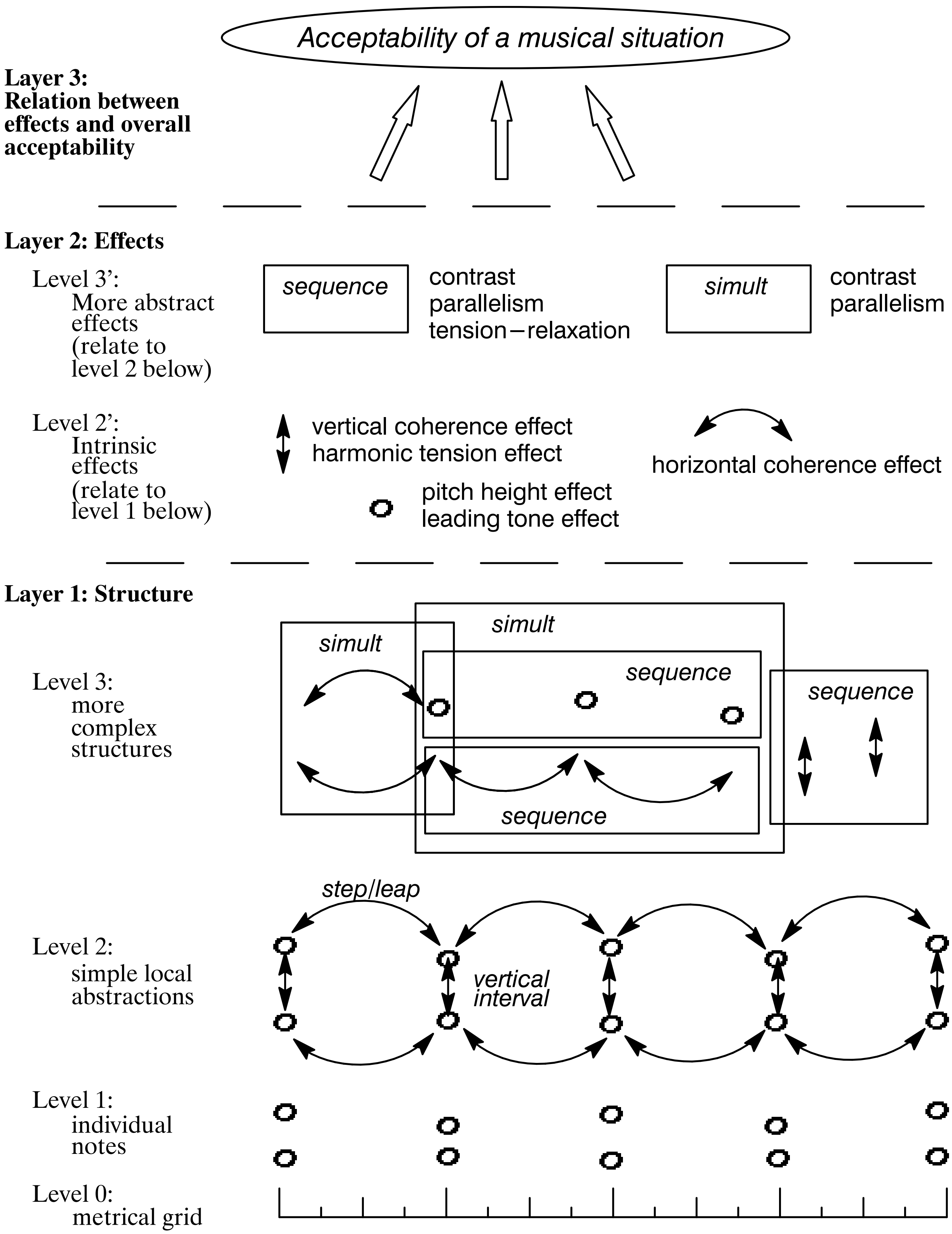
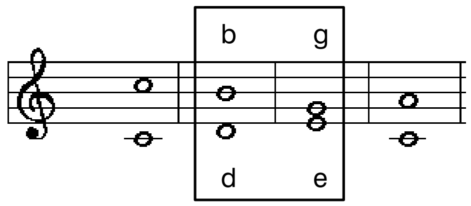


Fig.3: A simple model of the perception of two−voice counterpoint

other piece of knowledge, an attached type label identifies the kinds of objects or situations to which the rule is applicable.



```
composite_situation(
  sequence(
     event(vertical_interval(b,d)),
     event(vertical_interval(g,e))))

composite_situation(
  simult(
     event(step_leap(b,g)),
     event(step_leap(d,e))))

composite_situation(
  sequence(
     event(vertical_interval(b,d)),
     composite_situation(
          simult(event(single_note(g)),
               event(single_note(e))))))

composite_situation(
  simult(
     composite_situation(
          sequence(event(single_note(b)),
                    event(single_note(g)))),
     event(step_leap(d,e))))
```

Fig.4: A musical situation and four parse trees

4.1.2 Effects: the sound of musical situations

Given this structural view of musical situations, we can now attempt to describe how certain situations *sound*. The central notion around which our approach revolves is that of an *effect*; effects correspond to simple kinds of more or less direct *sensations* in the listener. Every event has some *intrinsic effects* (level 2' in layer 2 of Fig.3), depending on its type, and more complicated (and more abstract) effects emerge when events are juxtaposed (in sequences or simults) – this is level 3' in Fig.3.

For instance, all events of type *vertical interval* have intrinsic effects *harmonic tension* and *vertical coherence*. The former depends primarily on the degree of consonance of the interval, whereas the latter is defined through the interval's width. Events are lined up along *scales of intensity* for each effect; this is part of the system's *a priori* knowledge. It knows, for example, that perfect consonances have an extremely low degree of harmonic tension (that is, they sound 'empty', 'smooth'), whereas most dissonances (especially narrow, altered intervals) have a very high harmonic tension effect (they sound extremely 'tense').

Effects that arise from the juxtaposition of events correspond to more abstract sensations in a listener. Some of these 'second order' effects are *contrast*, *coherence*, *parallelism*, *tension*, and *relaxation*, where contrast, for instance, is defined as arising from a combination of intrinsic effect differences between the events involved in the situation.

4.2 The representation of musical knowledge

The kind of knowledge about effects listed above is mainly definitional. It is not sufficient to improve learning. Knowledge about *dependencies* between effects and the acceptability of a musical situation is needed. We believe that people have this knowledge, if only in a very general form. Even people with no explicit musical knowledge can usually say what it is that bothers them in an 'awkward' musical situation.

In our system, this kind of knowledge comes in three forms: *deductive rules*, *determinations*, and *plausibility heuristics*. The following sections will describe each of these in turn and demonstrate how they contribute to an effective learning process.

4.2.1 Deductive rules

Parts of the musical 'knowledge' referred to above can be formulated as a set of hierarchically dependent rules (an *incomplete domain theory* (Mitchell *et al.* 1986)). These rules relate certain lower– and higher–order effects to the general acceptability of a musical situation.

Fig.5 gives a simplified sketch of some of these rules and shows how they allow the system to learn a simple rule from just one training example (the note c in the training example was labelled as bad by the teacher). The rule that is learned says that perfect vertical consonances are bad and should be avoided (rationale: they sound 'dull' because of a lack of harmonic tension). The learning method is straightforward Explanation–Based Learning (EBL) – the system finds an explanation for why the c may sound bad, and then generalizes the explanation by turning constants into variables. Finally, a general rule is extracted from the explanation by collecting the leaves of the generalized explanation tree.

Many of these deductive rules describe factors that influence the way that music listeners intuitively rate the quality of a simple piece (for instance, that harmonic tension, contrast, and parallelism are related to the 'interestingness' of a musical situation). If this knowledge were complete in the sense that it could explain all the rules of counterpoint, our apprentice could learn all the rules perfectly by EBL. Not much to the surprise of anyone who knows anything about music, this is not the case. But after some introspection, it is easy to formulate some weaker forms of general musical knowledge.

4.2.2 Determinations

Determinations (Russell 1986) are a formalism for the expression of weaker knowledge. They are statements of *general dependencies* between attributes of a situation. They do not

```
bad(Piece,Note) :– in_context(Note,C), incoherent(C).        Knowledge relating
bad(Piece,Note) :– in_context(Note,C), dull(C).              abstract effects to
...                                                          the acceptability of
dull(C) :– intrinsic_effect(C,E,S), dull_effect(E,S).        a solution
...

intrinsic_effect(vertical_interval(N1,N2),harmonic_tension,low) :–
   is_interval(N1,N2,perfect_consonance).                     Knowledge about
...                                                           perceivable effects
dull_effect(harmonic_tension,low).                            of simple music
...

in_context(N,vertical_interval(N,N1)) :–                     Grammar capable of
   parallel_note(N,N1).                                      describing the structure
...                                                          of simple pieces
```

Fig.5.a: Set of given deductive rules (simplified)

Piece p:

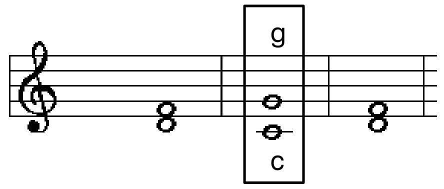


Fig.5.b: A training instance

$$\begin{tikzcd}
 & \text{bad(p,c)} & & & \\
\text{in\_context(c,vertical\_interval(c,g))} \arrow[ur] & & & \text{dull(vertical\_interval(c,g))} \arrow[ull] & \\
\text{parallel\_note(c,g)} \arrow[u] & & \text{intrinsic\_effect(vertical\_int(c,g), harmonic\_tension,low)} \arrow[ur] & & \text{dull\_effect (harmonic\_tension, low)} \arrow[ul] \\
 & & \text{is\_interval(c,g,perfect\_consonance)} \arrow[u] & & \\
 & & \text{is\_interval(c,g,‘P5’)} \arrow[u] & &
\end{tikzcd}$$

Fig.5.c: Sketch of the explanation tree

```
bad(P,N) :– parallel_note(N,N1),
            is_interval(N,N1,perfect_consonance).
```

Fig.5.d: Rule extracted from generalized explanation

contain information that is specific enough for problem solving or for explaining new situations, but they can serve as a basis for the search for plausible explanations.

Fig. 6.a shows a rather general and powerful determination from our system, and the rest of Fig.6 sketches how it is used in the process of learning the ‘parallel fifths’ rule (actually, a generalization thereof). The learning method used here is a combination of EBL and determinations. The determination’s role is essentially to point to possible explanations for a phenomenon; these can then be verified by the user or by reference to other, known situations which satisfy the same determination in the same way (Davies & Russell 1987).

Determinations are a natural and attractive form for the formulation of general, abstract knowledge. The one depicted in Fig.6 is a very simple, intuitively clear musical heuristic; when writing it down, one need not know in advance for which cases and combinations of events and effects it will hold. This will be learned by the system.

4.2.3 Plausibility heuristics

The weakest form of knowledge in our system comes in the form of *heuristics for inductive generalization*. These are meant to describe very weak and overly general intuitions about music.

Here is an example of a plausibility heuristic:

> “When trying to explain why a solution is bad or unacceptable, prefer hypotheses that contain some extreme or unusual event/effect.”

This allows for a *heuristic search* for the best explanation and hence for the best inductive generalization (see below). The main effect is that more ‘plausible’ hypotheses are tried first. Not only does this speed up the learning process, but it also saves the user the trouble of having to reject a lot of – in his/her eyes – ‘nonsensical’ and unexpected hypotheses presented by the system for verification.

```
determines( bad_effect( Event1, Effect1, Degree1) &
          bad_effect( Event2, Effect2, Degree2),
        unacceptable_situation( sequence( Event1, Event2))).
```

A sequence *of two events* Event1, Event2 *which are both* bad *may be* unacceptable; *whether or not this is the case depends on exactly what the effects are that make the two events bad* (Effect1, Effect2), *and how strong they are* (Degree1, Degree2).

Fig.6.a: A determination

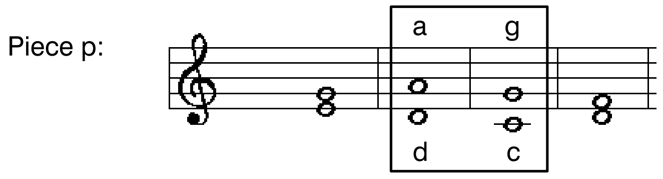


Fig.6.b: A training instance

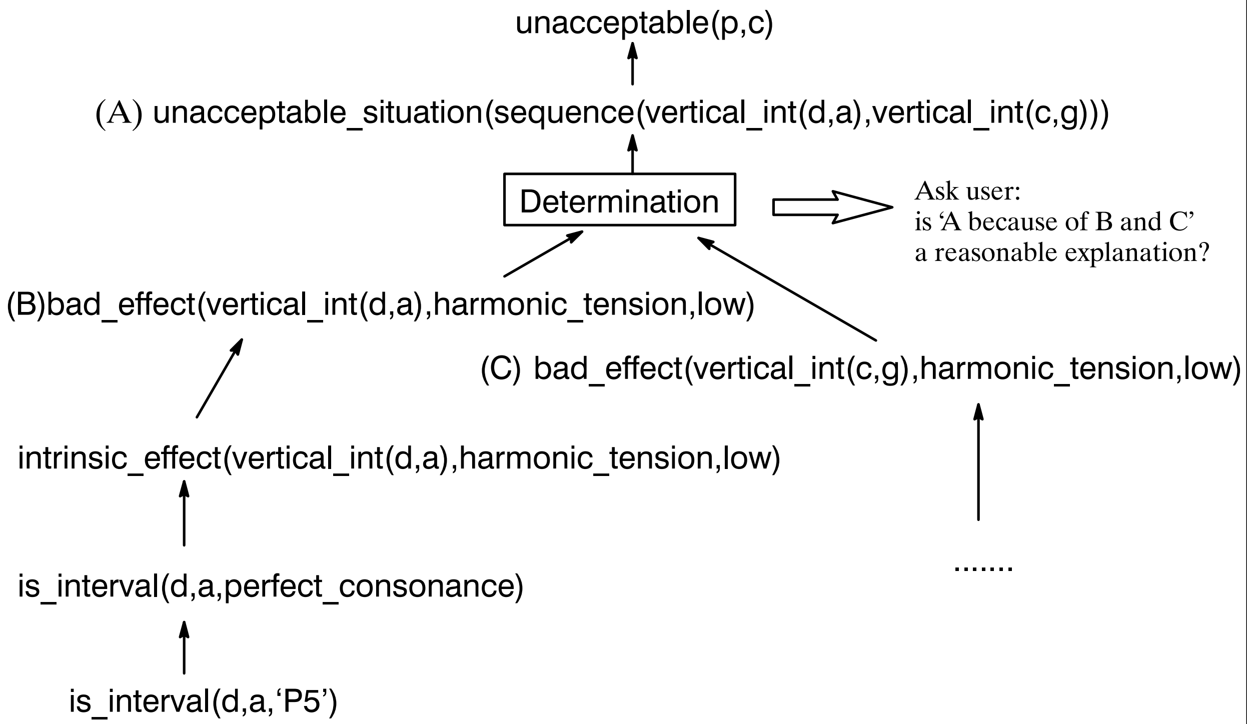

unacceptable(p,c)
(A) unacceptable_situation(sequence(vertical_int(d,a),vertical_int(c,g)))
Determination
Ask user:
is 'A because of B and C'
a reasonable explanation?
(B)bad_effect(vertical_int(d,a),harmonic_tension,low)
(C) bad_effect(vertical_int(c,g),harmonic_tension,low)
intrinsic_effect(vertical_int(d,a),harmonic_tension,low)
is_interval(d,a,perfect_consonance)
....
is_interval(d,a,'P5')

Fig.6.c: Explanation process (combination of determination and EBL)

```
unacceptable(P,N) :- parallel_note(N,N1),
                     previous_note(N,N2),
                     parallel_note(N2,N3),
                     is_interval(N,N1,perfect_consonance),
                     is_interval(N2,N3,perfect_consonance).
```

Fig.6.d: Rule extracted from generalized explanation

5 The integrated learning strategy

5.1 The learning algorithm

The three types of explicit background knowledge – deductive rules, determinations, heuristics – require three different learning methods, namely, EBL, determination–based learning, and inductive generalization. Our apprentice combines and integrates these rather different methods in a framework of *explaining and generalizing*: given a training example (a musical situation) classified as good, bad, or unacceptable by the user, the system tries to explain why this might be so. An explanation is a tree reducing the goal (say, bad(P,N)) to simpler, easily testable ('operational') conditions. These are then generalized to yield the conditions for a generally applicable rule.

In every step of the explanation process, the system tries to apply the strongest forms of knowledge available:

(a) if a deductive rule is applicable to the current explanation subgoal, an EBL step is performed (see section 4.2.1).

(b) if the current subgoal matches the latter part of a determination, the determination's preconditions are evaluated recursively (using methods (a), (b), or (c)), and the user is asked to verify the resulting explanation (see section 4.2.2).

(c) if neither (a) nor (b) are possible, the system tries to 'explain' the current subgoal on the basis of its *similarity* to other known situations or to rules that have already been learned. Since there are usually several alternatives, all the available heuristics are used to determine which of these is the most plausible one (see section 4.2.3). Again, the user is asked to verify such explanation steps. This type of 'explanation' step leads to effects of inductive generalization.

Explanations based on 'weak' knowledge (steps (b) and (c) above) are presented to the user for confirmation or rejection. In this way, the learned rules will be strongly justified, even if the underlying knowledge was unreliable. However, using all its knowledge, the system tends to come up with plausible explanations very quickly, so that this is not too much of a burden on the human teacher. The reader interested in Machine Learning is referred to (Widmer 1989) for details of the learning algorithm.

5.2 An example

The following example demonstrates how the various learning modes are integrated in a more complex learning situation. Suppose that the apprentice has already learned the very specific rule depicted in Fig.7.b, which says that a note is bad if it forms an interval (jump) of a minor seventh (min7) with its predecessor. Suppose also that at some point during problem solving, the system creates the solution depicted in Fig.7.c, at which point the teacher interrupts it to say that note g2 is unacceptable. Fig.7.c shows the explanation tree built by the system. The tree is constructed as follows:

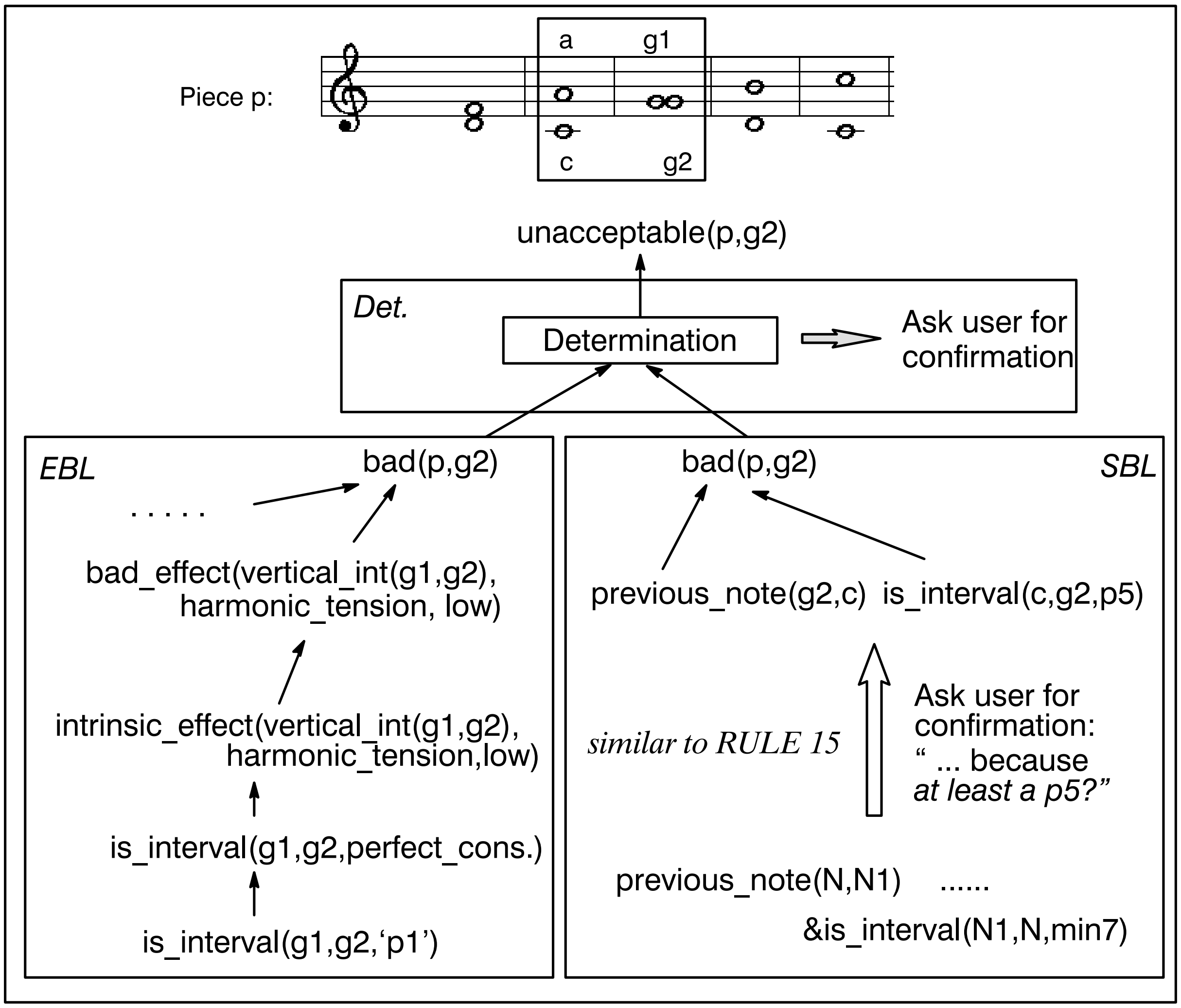
```
determines((bad( Piece, Note) :- Reason1,
                 (bad( Piece, Note) :- Reason2,
                  notequal( Reason1, Reason2))),
            unacceptable( Piece, Note)).
```

A note that is bad for two different reasons may be totally unacceptable; whether or not this is the case is determined by the particular reasons for which the note is bad.

Fig.7.a: A given determination

```
RULE 15:  bad(N) :- previous_note(N,N1) & scale_degree(N, second) &
                     scale_degree(N1,root) & is_interval(N1,N,min7).
```

Fig.7.b: A specific rule already learned



Piece p:
a
g1
c
g2
unacceptable(p,g2)
Det.
Determination
Ask user for confirmation
EBL
bad(p,g2)
. . . . .
bad_effect(vertical_int(g1,g2), harmonic_tension, low)
intrinsic_effect(vertical_int(g1,g2), harmonic_tension,low)
is_interval(g1,g2,perfect_cons.)
is_interval(g1,g2,'p1')
bad(p,g2)
SBL
previous_note(g2,c) is_interval(c,g2,p5)
similar to RULE 15
Ask user for confirmation: " ... because at least a p5?"
previous_note(N,N1) ......
&is_interval(N1,N,min7)

Fig.7.c: A training instance and its explanation

```
unacceptable(P,N) :- parallel_note(N,N1) & is_interval(N,N1,perfect_consonance) &
                     previous_note(N,N2) & at_least_interval(N2,N,p5)
```

Fig.7.d: New rule extracted from generalized explanation

- Trying to find an explanation for unacceptable(g2), the system finds the determination of Fig.7.a and tries to establish its preconditions, that is, it attempts to show that g2 is bad for two different reasons.
  - First, it succeeds in proving bad(g2) using the deductive rules in Fig.5.a. This results in the left subtree in Fig.7.c and establishes an EBL−type sub−explanation ("g2 is bad because it forms a perfect consonance − specifically, a perfect unison (p1) − with g1, and perfect consonances have low harmonic tension, which sounds dull").
  - Trying to show that bad(g2) holds for a second reason, the system comes across the specific rule already learned and listed in Fig.7.b and, using some heuristics, finds that the rule is rather similar to one aspect of the current situation (namely, the jump c−g2, which is an interval of a perfect fifth (p5)). The two situations are similar in that in both cases, there is a melodic jump which is *at least as big as a p5*. This similarity is presented to the teacher as a hypothesis for bad(g2) ("Is the jump from c to g2 bad because it is at least a p5?"), and the teacher agrees, which establishes precondition 2 of the determination and introduces an inductive generalization into the explanation.
- Finally, the need to establish the relevance of the determination prompts another question to the teacher ("unacceptable(g2) because <sub−explanation1> and <subexplanation2>?"). The teacher agrees again, and this completes the explanation.

The generalized rule extracted from the explanation is shown in Fig.7.d. The rule basically says that it is prohibited to 'jump into' a perfect consonance (perfect consonances should be introduced by stepwise motion). Note that this rather general rule was learned on the basis of one training instance, using three different kinds of background knowledge plus a previously learned rule and two very specific questions to the teacher.

Examples of the kinds of rules the system can learn appear in the appendix, where a particular experiment with the system is presented.

6 Conclusion

We believe that our learning apprentice project represents a case of a fruitful marriage of Artificial Intelligence and Music. On the one hand, it leads to the development of more flexible and reliable learning algorithms, which is beneficial to the entire field of Machine Learning. On the other hand, attempts to model some basic features of musical perception and experiments with a system that uses this model as a basis for learning may lead to interesting insights into the structure of tonal music. Such attempts may provide us with indications concerning the logical connection between musical perception theories and specific musical conventions by answering questions like the following:

- How do different sets of *a priori* knowledge affect the 'learnability' of the rules of a given musical style?
- Is there a certain *regularity* to the connection between the underlying theory and the operational 'surface' rules?
- Which part of the rule system of a particular musical style follows 'naturally' from which set of underlying assumptions about perception?

We believe that the kinds of knowledge structures described in section 4 represent a legitimate first step to providing a computer with the equivalent of a simple ‘ear’. We are still far from having a satisfactory theory of musical events and effects. In particular, first species counterpoint is too simple a musical problem to allow for musicologically interesting experiments. In the meantime (1990−1991), the project has advanced beyond this rather simple type of music; a new, more ambitious system has been implemented that can learn to harmonize given melodies. That system is based on a more complex and also more general theory of musical perception (see Widmer 1990a,b,c). First experimental results are very encouraging. The ultimate goal is to construct a general (qualitative) model of human musical perception, based on common−sense observations and intuitions. That kind of musical model could then be used in experiments to yield meaningful answers to the kinds of questions hinted at above.

Acknowledgements

I would like to thank Bernhard Pfahringer, Peter Kerstan, Paolo Petta, and John Matiasek for interesting discussions. Thanks are also due to Prof. Robert Trappl for his support of the project. Support for the Austrian Research Institute for Artificial Intelligence is provided by the Austrian Federal Ministry for Science and Research. Parts of this research were sponsored by the Austrian *Fonds zur Foerderung der wissenschaftlichen Forschung*.

The Learning Apprentice has been implemented in Quintus Prolog and runs on an Apollo DN3000 workstation. It communicates with the user via a graphical, mouse−sensitive music editor (written in C).

References

Davies, T. and Russell, S. (1987). A Logical Approach to Reasoning by Analogy. In *Proceedings of IJCAI−87, Milan.* Morgan Kaufmann Publishers.

DeJong, G. and Mooney, R. (1986). Explanation−Based Learning: An Alternative View. *Machine Learning 1(1).*

Dietterich, T. and Michalski, R. (1981). Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods. *Artificial Intelligence 16(3)*.

Michalski, R. (1983). A Theory and Methodology of Inductive Learning. In R.Michalski, J.Carbonell, T.Mitchell, eds., *Machine Learning: An Artificial Intelligence Approach*, Palo Alto, CA: Tioga.

Mitchell, T. (1982). Generalization as Search. *Artificial Intelligence 18(2)*.

Mitchell, T., Mahadevan, S. and Steinberg, L. (1985). LEAP: A Learning Apprentice for VLSI Design. In *Proceedings IJCAI−85, Los Angeles.* Morgan Kaufmann Publishers.

Mitchell, T., Keller, R. and Kedar−Cabelli, S. (1986). Explanation−Based Generalization: A Unifying View. *Machine Learning 1(1)*.

Rendell, L. (1987). Similarity−Based Learning and its Extensions. *Computational Intelligence 3*.

Russell, S. (1986). *Analogical and Inductive Reasoning.* Ph.D. Thesis, Report STAN−CS−87−1150, Stanford University, Stanford, CA.

Smoliar, S. (1980). A Computer Aid for Schenkerian Analysis. *Computer Music Journal 4(2)*.

Widmer, G. (1989). A Tight Integration of Deductive and Inductive Learning. In *Proceedings of the Sixth International Workshop on Machine Learning.* Ithaca, N.Y.: Morgan Kaufmann. Extended version available as Technical Report TR−89−3, Austrian Research Institute for Artificial Intelligence, Vienna.

Widmer, G. (1990a). The Usefulness of Qualitative Theories of Musical Perception. In *Proceedings of the International Computer Music Conference (ICMC−90)*, Glasgow, U.K.

Widmer, G. (1990b). *A Qualitative Model of the Perception of Melodies and Harmonizations*. Report TR−90−12, Austrian Research Institute for Artificial Intelligence, Vienna.

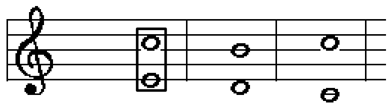
Widmer, G. (1990c). *Using a Qualitative Perception Model to Learn Harmonization Rules for Melodies*. Report TR−90−13, Austrian Research Institute for Artificial Intelligence, Vienna.

Winston, P. (1975). Learning Structural Descriptions from Examples. In P.Winston, ed., *The Psychology of Computer Vision.* New York: McGraw Hill.

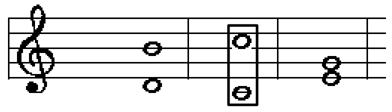
Appendix: An experiment

The experiment described below demonstrates how the availability of general musical knowledge greatly improves the efficiency of the learning process. The setup was as follows: we settled on a set of eight rules which the system should learn. They are sketched below.

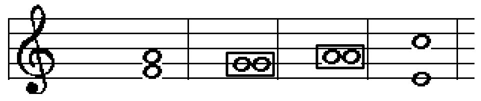
Rule 1: *"A first species counterpoint piece must not begin with an imperfect consonance"*
(unacceptable)



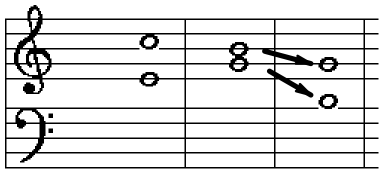
Rule 2: *"Perfect consonances should rarely be used"*
(bad)



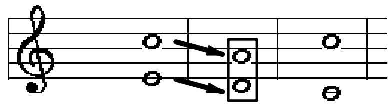
Rule 3: *"Sequences of perfect consonances are illegal"*
(unacceptable)



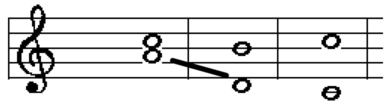
Rule 4: *"Parallel motion tends to be bad"*
(bad)



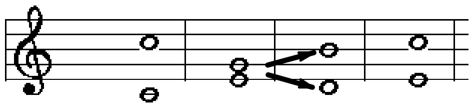
Rule 5: *"Introducing perfect consonances by parallel motion is not allowed"*
(unacceptable)



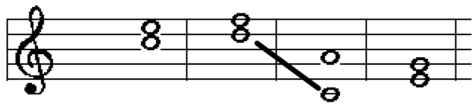
Rule 6: *"Melodic jumps bigger than a perfect fourth tend to be bad"*
(bad)



Rule 7: *"Motion in opposite directions is good"*
(good)



Rule 8: *"Melodic jumps bigger than an octave are not allowed"*
(unacceptable)



The system was then run in two modes: once with the full domain theory (perception model), the second time without it (only the musical heuristics – see section 4.2.3 – were left in the system). That is, in the second case (without domain theory), all the rules would have to be learned in a purely *inductive* way. We then measured the number of training examples the apprentice would require, and the number of questions it would have to pose to the teacher, in order to learn just this set of rules. The results show the positive effect of *a priori* knowledge very clearly: the eight rules were completely learned in both cases. However, in case 1 (with domain theory), the apprentice needed *10 examples* and asked a total of *18 questions*, whereas in case 2 (almost no musical knowledge), *24 examples* and *42 questions* were needed for the system to learn the same set of rules. It must be stressed that no changes were made to the system – we just varied the amount of available musical knowledge.