音 楽 情 報 科 学 7－4
（1 9 9 4． 8． 7）

Logic of Music and its Mechanization

Hajime Sawamura

Institute for Social Information Science (ISIS), FUJITSU LABORATORIES LTD.
140 Miyamoto, Numazu, Shizuoka 410-03, JAPAN
hajime@iias.flab.fujitsu.co.jp

abstract

HOL(Higher-Order Logic Theorem Prover) has been mainly applied to exact sciences such as mathematics and logics of programs in which human's rational thinking is actively engaged. As another direction of applications of the HOL theorem prover, we explore arts which is involved in human emotion, in particular, music. First, we present a logical music system which is supposed to be an underlying logic for both reasoning about music and composing music. This is based on two previous work on music theory in which logic plays a key role: Kurkela's semantical analysis of music based on Montague's intensional logic and Kassler's logical system for Schoenberg's twelve-tone music. Next, a mechanization of such a logical music system is provided very smoothly by using the expressiveness and proof facilities of HOL and by programming ML functions representing the symbol manipulations to be needed in the logical music system.
We believe that in the future, theorem proving technology would help human's emotional activities as well, in addition to human's rational information processing. This paper may be viewed as a first attempt toward relating music to logic in the area of automated theorem proving.

音楽の論理とその機械化

沢村　一

（株）富士通研究所　情報社会科学研究所

高階論理定理証明機HOLは数学，プログラムの論理など人間の合理的思考に関わる厳密科学に対して適用されてきた．本論文では，HOL定理証明機の他の方向の応用として，人間の感性に関わる領域，特に音楽への応用を探求する．先ず，音楽について論じたり音楽を論理的かつ形式的に作曲するための基礎となる論理体系を提案する．これは音楽理論に関するこれまでの二つの研究：KurkelaのMontagueの内包論理による音楽の論理分析とKasslerのSchoenbergの12音音楽に対する論理体系，に基づいて形式化された．次に，このような音楽の論理における論証／計算を機械的に遂行するために，HOL定理証明機の上あるいは中に埋め込む方法を与えた．これらによって，音楽活動を定理の証明プロセスと見るこれまでとは異なった新しい視点がありうることを明らかにした．

1. Introduction

Mathematical methods of the kind studied in logic are extensively used and applied to a broad class of questions of a logical nature. Interestingly, arts like music, picture, and so on are no exception. It is no longer misleading that logic plays an important and even essential rôle not only in computer science and artificial intelligence, but also in aesthetics which has been thought of as being in a directly opposite position to logic.

In her paper [Langer 25], Langer attempted to formalize music as a logical system, based on the recognition that every universe of discourse has its logical structure. This seems to be the first treatise to view music from a logical viewpoint, as far as I know. After that, we can see a few independent results which relate music to logic, based on logical analysis of music (e.g., [Kassler 63], [Kunst 76], [Rahn 79], [Kurkela 86]).

The HOL system [Gordon 93], on the other hand, is a proof development system developed by Gordon and intended for applications to both hardware and software. It is principally used in two ways: for directly proving higher-order theorems, and as theorem-proving support for other logics. In particular, HOL has allowed to mechanize various logics appearing in logical studies of computer science and artificial intelligence (e.g., [Gordon 89], [Melham 92], [Nesi 92], [Jacobs 93]), which belong to, so to speak, exact sciences that require human rational thinking. The mechanization has been essentially done by embedding original semantics of logics into the HOL logic, a variety of higher-order logic based on Church's formulation of the simple theory of types.

In this paper, we are more interested in the latter aspect of HOL above, and challenge to apply the HOL theorem prover to music which is involved in human emotion and to reason about music through HOL. The logical music system we consider here is based on two previous work on music theory in which logic plays a key role. The first one is Kurkela's work [Kurkela 86] on the semantical analysis of a musical score. He exploited Montague's intensional logic to give a meaning to notes and tones which are constituents of music scores. The second one is Kassler's system R_1 [Kassler 63] which he presented as a logical system characterizing Schoenberg's twelve tone music. What we are about to undertake in this paper is a marriage of these two independent theories in such a way that the resulting musical system allows us to discuss meaning of music and to mechanize it in the HOL theorem prover.

The contents of the remaining sections of the paper are as follows: Section 2 describes a logical music system which is supposed to be an underlying logic for reasoning about music. It includes descriptions of intensional logic and Kassler's music system. Section 3 discusses how such a logical music system can be mechanized using HOL. Final Section contains concluding remarks and a brief discussion of future work.

2. Logical music system

In this section, we describe a simple logical music system which is constructed from two former results on formalized music by Kurkela [Kurkela 86] and Kassler [Kassler 63]. Kurkela first applied the Montague's semiotic program [Dowty 81]for natural languages to another formalized language, musical language. He supposed that a musical score should be analyzed via the transformation to intensional logic just as a natural language sentence is first transformed into an intensional logic term and then interpreted over possible worlds. On the other hand, Kassler presents the logistic system R_1 and some interpretations for it. From music-theoretic point of view, his formalization of music as a logical system seems to be founded on the two influential ideas: Schoenberg's twelve-tone system and Schenkerian analysis of music [Forte 82]. Early this century, a kind of an axiomatic definition for tonal music was presented by Heinrich Schenker who viewed music as being able to derivable from certain fundamental structures (Ursätze) regarded as axioms by music development techniques called prolongation. Surprisingly, this theory bears many similarities to the ideas of Chomsky's transformational generative grammar theory.

2.1 Intensional logic (IL) as an underlying logic of music

We employ the intensional logic [Gallin 75] as an underlying logic to formalize music. Intensional logic is a higher-order modal logic with the special operators: intension (^) and extension ($^\vee$) added to the simple type theory by Church. Intensional logic also has possible worlds semantics for the language, which allows us to interpret music assertions in possible worlds (aesthetic worlds in this case). Here we follow Gallin's formulation in describing intensional logic, except for the idiosyncratic part for music.

Definition 1. (Types) The set of types of IL is the smallest set T satisfying: (i) $e, t \in T$, (ii) $\alpha, \beta \in T$ imply $<\alpha, \beta> \in T$ and (iii) $\alpha \in T$ implies $<s, \alpha> \in T$.

Objects of type e are entities (sound events), objects of type t are truth-values, objects of type $<\alpha, \beta>$ are functions from objects of types α to objects of type β, and objects of type $<s, \alpha>$ are functions from possible worlds to objects of type α, senses appropriate to denotations of type α.

For our purpose, we consider the twelve octave-equivalent note-class for music, as in Fig. 1. Thus, the note-class C, for instance, stands for the note-class {all C's, all B-sharp's, D-double flat's, etc.} or its representative.

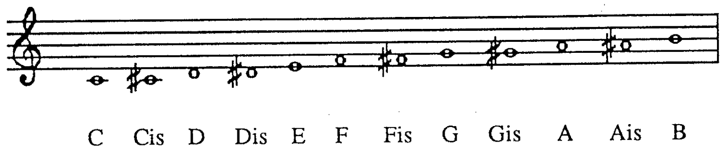

Fig. 1 Twelve-note-class

Definition 2. (Primitive symbols) We prepare four kinds of primitive symbols for music.

(i) The improper symbols: $\equiv$, λ, $^\wedge$, $^\vee$, [,].

(ii) For each $\alpha \in T$, we have a denumerable list of variables.

(iii) For each note-class, we have predicate constants of type <e,t>: c, cis, d, dis, e, f, fis, g, gis, a, ais, b which specify the twelve-note-class above.

(iv) For each note, we have predicate constants of type <e,t>: ♩, ♪, 𝅗𝅥, 𝅝, etc., which specify the relative lengths (duration) of the quarter note, eighth note, half note, whole note, etc., respectively.

The following terms of IL are used for music assertions as well as for natural language in Montague's semiotic program.

Definition 3. (Terms) The set Tm_α of terms of IL of type α are recursively defined as follows :

(i) Every variable of type α belongs to Tm_α.
(ii) Every constant of type <e,t> belongs to $Tm_{<e,t>}$.
(iii) $A \in Tm_{<\alpha,\beta>}$, $B \in Tm_\alpha$ imply $[AB] \in Tm_\beta$.
(iv) $A \in Tm_\beta$, x a variable of type α imply $\lambda xA \in Tm_{<\alpha,\beta>}$.
(v) $A, B \in Tm_\alpha$ imply $[A \equiv B] \in Tm_t$.
(vi) $A \in Tm_\alpha$ implies $^\wedge A \in Tm_{<s,\alpha>}$.
(vii) $A \in Tm_{<s,\alpha>}$ implies $^\vee A \in Tm_\alpha$.

We write A_α for A when $A \in Tm_\alpha$, but we sometimes omit the types if no confusion arises. We adopt the usual conventions regarding grouping in terms. In particular, parentheses (,) in place of brackets [,] sometimes are adopted, and outermost brackets may be omitted.

Definition 4. (Definitional constants) We introduce the usual logical constants, connectives and quantifiers in IL by definition:

$T =_{def} [\lambda x_t x_t \equiv \lambda x_t x_t]$

$F =_{def} [\lambda x_t x_t \equiv \lambda x_t T]$

$\neg =_{def} \lambda x_t [F \equiv x_t]$

$\wedge =_{def} \lambda x_t \lambda y_t [\lambda f_{<t,t>}[fx \equiv y] \equiv \lambda f_{<t,t>}[fT]]$

$\supset =_{def} \lambda x_t \lambda y_t [[x \wedge y] \equiv x]$

$\vee =_{def} \lambda x_t \lambda y_t [\neg x \supset y]$

$\forall x_\alpha A_t =_{def} [\lambda x_\alpha A_t \equiv \lambda x_\alpha T]$

$\exists x_\alpha A_t =_{def} \neg \forall x_\alpha \neg A_t$

$\Box A_t =_{def} [^\wedge A_t \equiv {}^\wedge T]$

$\Diamond A_t =_{def} \neg \Box \neg A_t.$

Following the usual notations, we write $[A \wedge B]$ instead of $[[\wedge A]B]$ where A and B are formulas, similarly for the other binary connectives.

A subset of terms called modally closed terms is needed to describe the axiomatic system of IL, which is defined as follows.

Definition 5. (Modally closed terms) The class MC of modally closed terms is the smallest class such that
(i) $x_\alpha \in MC$ for every variable $x_\alpha \in Tm_\alpha$
(ii) $^\wedge A_\alpha \in MC$ for every term A_α
(iii) $A_{\alpha\beta} B_\alpha \in MC$ if $A_{\alpha\beta}, B_\alpha \in MC$
(iv) $A_\alpha \equiv B_\alpha \in MC$ if $A_\alpha, B_\alpha \in MC$

(v) $\lambda x_\alpha A_\beta \in$ MC if $A_\beta \in$ MC.

Informally, a modally closed term is such a term that its denotation is independent of possible worlds.

Definition 6.(Gallin's system of IL [Gallin 75]) IL has an equational deductive system consisting of six axioms and one replacement rule.

Axioms:

A1: $g_{<t,t>}T \wedge g_{<t,t>}F \equiv \forall x_t[g_{<t,t>}x_t]$
A2: $x_\alpha \equiv y_\alpha \supset f_{<\alpha,t>}x_\alpha \equiv f_{<\alpha,t>}y_\alpha$
A3: $\forall x_\alpha[f_{<\alpha,\beta>}x_\alpha \equiv g_{<\alpha,\beta>}x_\alpha] \equiv [f_{<\alpha,\beta>} \equiv g_{<\alpha,\beta>}]$
A4: $(\lambda x_\alpha A_\beta(x_\alpha))B_\alpha \equiv A_\beta(B_\alpha)$ (β-conversion axiom), where $A_\beta(B_\alpha)$ come from $A_\beta(x_\alpha)$ by replacing all free occurrences of x_α by the term B_α, and providing that B_α is free for x_α in $A_\beta(x_\alpha)$, and either no free occurrence of x_α in $A_\beta(x_\alpha)$ lies within the scope of ^, or else $B_\alpha \in$ MC.
A5: $\Box[{}^\vee f_{<s,\alpha>} \equiv {}^\vee g_{<s,\alpha>}] \equiv [f_{<s,\alpha>} \equiv g_{<s,\alpha>}]$
A6: ${}^\vee{}^\wedge A_\alpha \equiv A_\alpha$ (cancellation axiom)

Inference rule R:

From $\vdash A_\alpha \equiv A'_\alpha$ and $\vdash B_t$ to infer the formula B_t', where B_t' comes from B_t by replacing one occurrence of A_α by the term A_α'.

A proof is defined in the usual way. In this paper, we will not describe the formal model theory for the intensional language. We just touch upon it when necessary. Here we just note that to a constant c of type α of IL, the meaning function *m* in a model assigns an intension (or sense) $m(c) \in D_{<s,\alpha>}$.

2.2 Notes as higher-order terms of intensional logic

Having the above intensional language in mind, we begin with defining notes in an abstract way. A note has as its properties the pitch, duration, tone, intensity and so on. Only the pitch and duration of a note, here, are taken into account for our treatment of notes. First we introduce the abstract note which skeletonize the pitch and duration of a note.

Definition 7. (Abstract note) The abstract note *n* is a function over possible worlds, whose type is <s,<<s,<e,t>>, <s,<<s,<e,t>>,t>>>, $^\wedge\lambda P^\wedge\lambda Q[\exists x(P\{x\} \wedge Q\{x\})]$, where
$A_{<s,\alpha>}\{B\} =_{def} [{}^\vee A_{<s,\alpha>}]B$ (we call this brace convention).

Definition 8. (Abstract note-class and abstract duration-class) The term of the form $^\wedge\lambda y_e[N_{<e,t>}y_e]$ of IL is called an abstract note-class, where $N_{<e,t>}$ denotes any of the primitive predicate constants of type <e,t>: c, cis, d, dis, e, f, fis, g, gis, a, ais, b. The term of the form $^\wedge\lambda z_e[D_{<e,t>}z_e]$ of IL is called an abstract duration-class, where $D_{<e,t>}$ denotes any of the primitive constants of type <e,t>: ♩, ♪, 𝅗𝅥, 𝅝, etc.

An abstract note-class is thought of as representing a sound image that we would have in mind, so to speak, the deep structure of a tone, and an abstract duration-class likewise. Then, the abstract note together with an abstract-note class and an abstract duration-class gradually yields somewhat concrete sound image, a kind of a surface structure of sound. For example, the note class A with the duration, quarter note

is defined as a term $n\{^\wedge\lambda y[a\ y]\}\{^\wedge\lambda z[♩z]\}$ of IL which is reduced to $\exists x(a\ x \wedge ♩x)$, within the framework of intensional logic, as follows (the proof is abbreviated in a series of rewriting, not in the primitive inference steps of IL):

$n\{^\wedge\lambda y[a\ y]\}\{^\wedge\lambda z[♩z]\}$

$= [{}^\vee{}^\wedge\lambda P^\wedge\lambda Q[\exists x(P\{x\} \wedge Q\{x\})]](^\wedge\lambda y[a\ y])\{^\wedge\lambda z[♩z]\}$ (by brace convention)

$= \lambda P^\wedge\lambda Q[\exists x(P\{x\} \wedge Q\{x\})](^\wedge\lambda y[a\ y])\{^\wedge\lambda z[♩z]\}$ (by cancellation axiom of IL)

$= {}^\wedge\lambda Q[\exists x([^\wedge\lambda y(a\ y)]\{x\} \wedge Q\{x\})]\{^\wedge\lambda z[♩z]\}$ (by β-conversion axiom of IL)

$= {}^\wedge\lambda Q[\exists x([{}^\vee{}^\wedge\lambda y(a\ y)]x \wedge Q\{x\})]\{^\wedge\lambda z[♩z]\}$ (by brace convention)

$$= {}^\wedge\lambda Q[\exists x([\lambda y(a\ y)]x \wedge Q\{x\})]\ \{{}^\wedge\lambda z[\text{♩}z]\}\ \text{(by cancellation axiom of IL)}$$

$$= {}^\wedge\lambda Q[\exists x([\lambda y(a\ y)]x \wedge Q\{x\})]\ \{{}^\wedge\lambda z[\text{♩}z]\}\ \text{(by } \beta\text{-conversion axiom of IL)}$$

$$= {}^\vee{}^\wedge\lambda Q[\exists x(a\ x \wedge Q\{x\})]\ ({}^\wedge\lambda z[\text{♩}z])\ \text{(by brace convention)}$$

$$= \lambda Q[\exists x(a\ x \wedge Q\{x\})]\ ({}^\wedge\lambda z[\text{♩}z])\ \text{(by cancellation axiom of IL)}$$

$$= \exists x(a\ x \wedge {}^\wedge\lambda z[\text{♩}z]\{x\})\ \text{(by } \beta\text{-conversion axiom of IL)}$$

$$= \exists x(a\ x \wedge [{}^\vee{}^\wedge\lambda z[\text{♩}z]]x\})\ \text{(by brace convention)}$$

$$= \exists x(a\ x \wedge [\lambda z[\text{♩}z]]x\})\ \text{(by cancellation axiom of IL)}$$

$$= \exists x(a\ x \wedge \text{♩}z)\ \text{(by } \beta\text{-conversion axiom of IL).}$$

Note that β-conversion are safely applied since it satisfies the side conditions of β-conversion axiom of IL. The quantified variable x of type e is thought of as ranging over (actual or imaginative) sound entities (events). The last term is, as we can see, quite a simple and well-formed quantified sentence of IL. It can be read: there exists at least one sound event x such that it belongs to an octave-equivalent pitch-class A and a quarter note duration. The extension of this term is, as usual, a truth value. If, given a world w, there exists at least sound event x such that it belongs to a pitch-class A with a quarter note duration-class, then the sentence become true at that world. In what follows, we simply consider the twelve note-class, $\mathcal{n}\{{}^\wedge\lambda y_e[N_{<e,t>}y_e]\} = {}^\wedge\lambda Q[\exists x(Nx \wedge Q\{x\})]$, abstracting the part of abstract duration-class. For instance, the note-class C above denotes the term ${}^\wedge\lambda Q[\exists x(c\ x \wedge Q\{x\})]$, the note-class Cis denotes the term ${}^\wedge\lambda Q[\exists x(cis\ x \wedge Q\{x\})]$, and so on.

In Section 3 below, we will consider how to deal with notes as intensional terms and how to prove formulas appeared in this section, using the HOL theorem prover.

2.3 Abstract music as a sequence of twelve-note-class

Thus far we have regarded the twelve-note-class as intensional terms which are to be undergone interpretations in possible worlds. Having those notes in mind, we proceed to music composition consisting of the twelve-note-class. In doing so, we follow Kassler's work [Kassler 63], who attempts to reconstruct Schoenberg's twelve-tone music theory in a logical setting.

For simplicity, each note-class in the twelve-note-class is identified with each of ten numerals and two letters, like in [Kassler 63] as follows:

C	Cis	D	Dis	E	F	Fis	G	Gis	A	Ais	B
0	1	2	3	4	5	6	7	8	9	a	b

Under this correspondence, we recapitulate Kassler's logical system R_1 (he originally called it logistic system), referring to Sharvy's review on Kassler's papers [Sharvy 75]. The language system of R_1 has the following syntactic classes.

Definition 9. (Primitive symbols, formulas, well-formed formulas and basic wffs) Formulas and well-formed formulas are descriptively defined from twelve primitive symbols as follows.

(i) Primitive symbols of R_1: the numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and two letters a, b.

(ii) Formulas of R_1: α is a formula iff α is a finite sequence of one or more primitive symbols of R_1.

(iii) Well-formed formulas (wff) of R_1: α is a wff iff α is a sequence over the primitive symbols containing at least one occurrence of each primitive symbol.

(iv) α is a basic wff iff α is a wff of length twelve, equivalently, iff each primitive symbol occurs exactly once in α.

(v) α is the basic wff of β iff α consists of the twelve primitive symbols in order of their first occurrence in β.

In the twelve-tone music theory, a basic wff corresponds to what is called "serie" (row in English).

Definition 10. (Permutation, segment and unprecedented segment) For any basic wff α and wff β, the permutation Pαβl is the wff obtained by substituting the ith primitive symbol of α for every occurrence in β of the ith primitive symbol of the basic wff of β, for $1 \le i \le 12$. α is a segment of β iff α is not the null string and for some string σ and π, β is the concatenation σαπ. If, in addition, α is not a segment of σ, then α is an unprecedented segment of β.

Definition 11. (Axiom and inference rules) R_1 has one axiom and three inference rules.
Axiom: 0123456789ab
Inference rules:

Restatement

$$\frac{\sigma\beta\pi}{\sigma\beta\beta\pi}$$, where β is a segment of the basic wff of the premise $\sigma\beta\pi$, and β is an unprecedented segment of the premise $\sigma\beta\pi$.

Repetition

$$\frac{\sigma\beta\pi}{\sigma\beta\beta\pi}$$, where β is a primitive symbol.

Permutation

$$\frac{\beta}{P\alpha\beta|}$$, where α is a basic wff.

2.4 Interpretation of R_1

In this subsection, we discuss the semantical aspects of R_1, and then an interpretation of music as a sequence of intensional terms representing abstract note-class. In his paper [Kassler 63], Kassler sketches three interpretations of R_1: ones for written music, produced music, and received music. In particular for written music, we introduce a somewhat modified interpretation along with his primary principal interpretation.

Definition 12. (Rhythm-free-associated vector and interpretation of wffs) For any monophonic composition, its rhythm-free-associated vector is the formula of R_1 whose ith position is filled by the primitive symbols associated with the abstract note-class to which the ith note-token of that composition belongs. Then, to every wff α is assigned the class of monophonic compositions whose rhythm-free-associated vector is α.

In this interpretation, therefore, starting with the axiom which denotes a set of compositions whose rhythm-free-associated vectors are 0123456789ab, the inference rules generate a wff α, that is, the class of monophonic compositions whose rhythm-free-associated vector is α. On the contrary, R_1 does not always yields a theorem as a wff which is a rhythm-free-associated vector of some class of monophonic compositions. Put differently, the interpretation is too rough for R_1 to be complete. Another deficiency of the interpretation is that nothing inherently restricts interpretation to the wffs as music assertions. We will discuss elsewhere how the note-class as intensional terms to be interpreted as senses and a music as a sequence of their note-class can remedy these difficulties encountered in the semantics of music.

For all that, we think it worthy to cite a music as a theorem in R_1 from [Kassler 63]. This yields an interesting musical evidence irrespective the interpretation above as well.

Proof: 0123456789ab (by axiom)
0b78312a6549 (by permutation)
0b78312a656549 (by reinstatement)
0b783112a656549 (by repetition)
0b7831112a656549 (by repetition)
0b78311112a656549 (by repetition)
0b783111112a656549 (by repetition)
0b7831111112a656549 (by repetition)
0b78311111122a656549 (by repetition)
0b78311111122a6656549 (by repetition)
0b78311111122a66656549 (by repetition)
0b78311111122a666656549 (by repetition)
0b78311111122a6666556549 (by repetition)
0b78311111122a66665565549 (by repetition)
Theorem: 0b78311111122a666655655449 (by repetition)

Arnold Schoenberg's Fourth String Quartet [Schoenberg 39] begins with a rhapsodic recitative which is performed by the four stringed instruments in unison of the same pitch. Kassler found that under the above interpretation, the first 4.75 measures of the third movement of it is a member of the set denoted by the theorem. In this interpretation, the proved theorem represents an intermediate structure of that passage, which lies between the deep and surface structures. In staff notation, the theorem sounds like

There is one more important question left: How can one map such an intermediate structure to a real music score? By some transformation? Or is it beyond logic? Let us leave the question up to musicologists for the time being, and go on to our second theme of this paper.

3. Mechanizing the logical music system in HOL

In this section, we consider how to implement the logical music system on reasoning systems, in particular in HOL. In doing so, we prefer directly implementing it on general systems to designing a new prover for music, taking the advantage of generality of general reasoning systems. Candidates for implementing it would be HOL [Gordon 93], EUODHILOS [Sawamura 90][Sawamura 91] [Ohtani 93], etc. In this paper, we consider the HOL implementation since the implementation of the logical music system presented in the previous sections is straightforward using HOL, in particular with the help of ML. The reason is twofold: tones are defined in higher-order representation suitable to HOL treatment, and the treatment of the R_1 part is suitable to ML.

The implementation is now in progress, and here we discuss only the basic ideas of implementing the logical music system. The implementation consists of two phases: encoding the IL part into higher-order logic and programming the R_1 part in ML.

3.1 Approximately encoding IL into higher-order logic

We will not directly encode IL to HOL by semantically embedding the possible worlds semantics for IL to the HOL higher-order language. We would rather be content with an approximate encoding of IL to HOL since the type system of IL is slightly different from the HOL's type system. Thus, although s itself is not a type but the form $<s,\alpha>$ can be type if α is a type, we dare to introduce the new type definition with care:

```
#new_type 0 's';;
```

in addition to

```
#new_type 0 'e';;
```

Note that for the type t in IL, the type bool in HOL is substituted. For logical constants, we set

```
#new_constant('int',":*->(s->*)");;
#new_constant('ext',":(s->*)->*");;
#new_constant('nec',":bool->bool");;
```

Here we write α->β for a type $<\alpha,\beta>$ in IL. For each primitive predicate constants of type $<e,t>$: c, cis, d, dis, e, f, fis, g, gis, a, ais, b, in Definition 2, we introduce the constant definitions:

```
#new_constant('c',":e->bool");;
#new_constant('cis',":e->bool");; ,etc.
```

For each abstract note-class in Definition 8, we introduce the definitions:

```
#new_definition('C',"C = int(\y:e.(c:e->bool) y)");;
#new_definition('Cis',"Cis = int(\y:e.(cis:e->bool) y)");; ,etc.
```

Then, the realization of IL as a HOL theory becomes almost straightforward except for the β-conversion axiom of IL. For the β-conversion axiom of IL is BETA_CONV substituted, provided that the condition " either no free occurrence of x_α in $A_\beta(x_\alpha)$ lies within the scope of ^, or else B_α is modally closed" is satisfied. The remaining five axioms of IL in Definition 6 looks like:

```
#new_axiom('A1',"!g.(g:bool->bool T:bool /\ g F:bool=!x.(gx:bool))");;
#new_axiom('A2',"!f x y.((x:* = y:*) ==> ((f:*->t) x = f y)");;
#new_axiom('A3',"!f g.((!x.((f:*->**) x = (g:*->**) x)) = (f = g))");;
#new_axiom('A5',"!f g.(nec(ext(f:s->*) = ext(g:s->*)) = (f = g))");;
#new_axiom('A6',"!A.int(ext A:s->*) = A") ;;
```

Finally, for the only rule of IL is the HOL tactics SUBST or SUBST_TAC substituted. With this encoding, we can easily prove such theorems as exemplified in the subsection 2.2, using the HOL reasoning apparatus. A number of elementary theorems of IL which appeared in Gallin's monograph [Gallin 75] were proved in this setting.

3.2 Programming R_1 in ML

Here we deal with a sequence of abstract note-class as a music assertion simply by the list processing in ML. Then, the axiom of R_1 is represented as a list whose elements' types are s->(e->bool) as follows:

```
[C; Cis; D; Dis; .........; B]
```

For the music derivation, it suffices to define ML functions representing rules of inference: Restatement, Repetition and Permutation in Definition 11. But this is easy.

3.3 Identification of music composition with theorem proving

We should mention how to enjoy music with HOL under R_1, or how to identify music composition with theorem proving task. There are two ways for that which are closely related to the proof styles of HOL:

(i) The style of forward proof

This leads us to constructing a musical assertion (theorem) step by step, beginning with the axiom and deriving intermediate results by repeated applications of inference rules. Of course musical imagination would be carried out in our mind and brain during this process.

(ii) The style of backward (goal-directed) proof

This leads us to testing if a putative composition is an acceptable music or not. R_1 is known to be decidable [Kassler 67]. So, this algorithm would be helpful for the parts of a whole composition.

As a matter of fact, the mixture of these proof styles could lead us to our final theorem (a music). Obviously, the use of automated theorem proving in music differs considerably from the mainstream of past and modern computer music studies, and might bring us a new insight to music, in particular meanings of music.

4. Concluding remarks and future work

Music is really a great fun! In this paper, we have outlined work in progress on how to deal with music theory, how to enjoy soundless music and hopefully how to compose music using the HOL theorem prover. We expect that these could help to open a new application area for the HOL theorem prover. Of course, these enterprises are far from complete. We just explored a very different direction for applications of the HOL theorem prover. We now, however, come to believe that the HOL-like theorem proving technology might become one of promising tools to support human's emotional activities and reasoning about emotion, sentiment, etc.

A lot of work is left, to be done in the future. Some of them are:

- performing proved theorems with an acoustic system
- realizing a whole story of Schenkerian analysis [Forte 82][Kassler 75] in HOL
- applying HOL to other arts, e.g., picture [Howell 74][Hintikka 75] as well
- explaining musical beauty in a logical framework [Kunst 76].

References

[Dowty 81] Dowty, D. R., Wall, R. E. and Peters, S.: Introduction to Montague Semantics, D. Reidel Pub. Co., 1981.

[Forte 82] Forte, A. and Gilbert, S. E.: Introduction to Schenkerian Analysis, W. W. Norton & Company, Inc., 1982.

[Gallin 75] Gallin, D.: Intensional and Higher-Order Modal Logic, with Applications to Montague Semantics, N-Holland Pub. Co., 1975.

[Gordon 89] Gordon, M. J. C.: Mechanizing Programming Logics in Higher Order Logic, Technical Report No. 145, Computer Laboratory, University of Cambridge, 1989.

[Gordon 93] Gordon, M. J. C. and Melham, T. F.: Introduction to HOL, Cambridge University Press, 1993.

[Hintikka 75] Hintikka, J.: The Intensions of Intensionality and Other New Models for Modalities, D. Reidel Pub. Co., 1975.

[Howell 74] Howell, R.: The Logical Structure of Pictorial Representation, Theoria 2, 1974, pp. 76-109.

[Jacobs 93] Jacobs, B. and Melham, T.: Translating Dependent Type Theory into Higher Order Logic, LNCS 664, Springer, 1993, pp. 210-229.

[Kassler 63] Kassler, M.: A Sketch of the Use of Formalized Languages for the Assertion of Music, Perspectives of New Music, Vol. 1, No. 2, 1963, pp. 83-94.

[Kassler 67] Kassler, M.: Toward a Theory that is the Twelve-Note-Class System, Perspectives of New Music, Vol. 5, No. 2, 1967, pp. 1-80.

[Kassler 75] Kassler, M.: Proving Musical Theorems I: The Middleground of Heirinch Schenker's Theory of Tonality, TR No. 103, Basser Dept. of Comp.Sci., The Univ. of Sydney, 1975.

[Kunst 76] Kunst, J.: Making Sense in Music I - The Use of Mathematical Logic, Interface, Vol. 5, 1976, pp. 69-86.

[Kurkela 86] Kurkela, K.: Note and Tone - A Semantical Analysis of Conventional Music Notation, Acta Musicologica Fennica 15, 1986.

[Langer 25] Langer, S. K.: A Set of Postulates for the Logical Structure of Music, Monist 39, 1925, pp. 561-570.

[Melham 92] Melham, T. E.: A Mechanized Theory of the Π-Calculus in HOL, Technical Report No. 244, Computer Laboratory, University of Cambridge, 1992.

[Nesi 92] Nesi, M.: A Formalization of the Process Algebra CCS in Higher Order Logic, Technical Report No. 278, Computer Laboratory, University of Cambridge, 1992.

[Ohtani 93] Ohtani, T., Sawamura, H. and Minami, T.: EUODHILOS-II on top of GNU Epoch, CADE-12, 1994 (to appear).

[Rahn 79] Rahn, J.: Logic, Set Theory, Music Theory, College Music Symposium, Vol. 19, No. 1, 1979, pp. 114-127.

[Sawamura 90] Sawamura, H., Minami, T., Yokota, K. and Ohashi, K.: A Logic Programming Approach to Specifying Logics and Constructing Proofs, Proc. of the Seventh International Conference on Logic Programming, edited by D. H. D. Warren and P. Szeredi, The MIT Press, 1990, pp. 405-424.

[Sawamura 92] Sawamura, H., Minami, T. and Ohashi, K.: EUODHILOS: A General Reasoning System for a Variety of Logics, Proc. of Logic Programming and Automated Reasoning, Lecture Notes in Artificial Intelligence 624, Springer, 1992, pp. 501-503.

[Schoenberg 39] Fourth String Quartet, OP. 37, G. Schirmer Inc., New York, 1939.

[Sharvy 75] Sharvy, R.: Review on Kassler's papers, The Journal of Symbolic Logic, Vol. 40, No. 4, 1975, pp. 576-577.