音 楽 情 報 科 学　42－9
（２００１．１０．２７）

ヘッドの概念を用いた和声学の文法

北陸先端科学技術大学院大学　情報科学研究科
東条　敏
tojo@jaist.ac.jp

概 要

和声学においては階層的な知識表現が数多く見られる．本稿では，この階層的な知識を HPSG (主辞駆動文法) を用いて表現することを目的とする．この文法理論では，各々のカテゴリーは属性構造で表わされるため，音楽の情報のさまざまな制約関係を記述するのが容易となる．さらにこの文法理論では，子カテゴリーの中に親カテゴリーの重要な情報を受け継ぐヘッドの概念が組み込まれており，これが和声学の中の和音やカデンツの知識表現に有用となる．音楽を解析する過程とは，この属性構造の中の複雑に絡み合った制約を解くことである．本稿では和音の表現とカデンツの認識例を示し，さらに GTTM との結び付きにおいて楽曲の自動解析の可能性について論じる．

Harmony Analysis in HPSG

Satoshi Tojo
Japan Advanced Institute of Science and Technology

Abstract

The formal theory of music consists of hierarchical knowledge. In this paper, we try to represent the theory of harmony, viz. chords and cadences, in HPSG. The grammar represents each category by a feature structure that is adequate for us to implement complicated inheritance of tonal information. Especially, the grammar provides the notion of 'head daughter' that mainly inherits the important information of its mother category, and we show how the notion fits to represent cadences as grammar rules. The parsing process of music is to solve complicated constraints between features. We discuss the examples of the recognition of a chord and that of a cadence. Also, we discuss the possibility of bidirectional parser/generator of music, together with the grouping algorithm of GTTM.

1　動機

音楽の理論の多くは階層的な知識表現からなる．例えば，三和音というのは三つの単音からなり，音階は西洋の近代音楽では七つの音からなり，またカデンツは三つあるい四つの和音の連なりからなっている．このようなことから音楽の理論は GTTM (Generative Theory of Tonal Music) [1] 以来，自然言語の文法との類似性・相違性が論じられている．平田ら [2] はこのような知識表現をするのに直接文法理論を使うのは避け，DOOD (演繹オブジェクト指向データベース) [3, 4] を用いた形式化を提案している．この方法はやはり属性構造を組み立てるパラダイムであるが，階層的な文法の概念やその中のヘッドの概念などを自分で定義する必要があるため，そのような概念が既に組み込まれた制約ベースの文法を用いれば記述量に大幅な改善が見込まれる．

制約ベースの文法の先駆としては GPSG (Generalized Phrase Structure Grammar) [5, 6] があり，これは各カテゴリーが属性構造の形をしているために文法規則の数を減らすことに寄与できる．HPSG (Head-driven Phrase Structure Grammar) [7, 8] は属性構造の考え方にさらにヘッドの概念を中心に据えた文法理論である．ヘッドとは子カテゴリーの中で親カテゴリーの重要な情報を受け継ぐものであり，また逆に，どのような階層構造であれ上位階層を形成する上では必ずこのような中心となる下位階層があるという考え方でできている．HPSG による構文解析とは，属性間の制約を解くことである．それは属性間の間で整合的な単一化を行うプロセスである．

以上のように，本稿では，HPSG は (i) 階層構造の表現，(ii) 属性構造，(iii) ヘッドの概念などが内

在しているために音楽の諸知識を表現をする上で有利であるという視点に立つ．次の節ではまず HPSG の概略を説明し，続く節では和声学の基礎を記述する試みを行う．次に，制約を実際に解くプロセスを例で示し，最後に本稿の寄与するところをまとめる．

2 HPSG 入門

本節では HPSG について概観する．文法規則は ID-スキーマといくつかの原則からなる．前者は通常の意味での生成規則に相当し，後者はカテゴリー間の制約である．HPSG では文法規則も個々のカテゴリーも属性構造で記述される．

2.1 属性のシステム

属性構造は属性とその値をペアにしたものを集めて角かっこでまとめた構造である．構造全体には一つのタイプが付与される．本稿ではタイプは '~' に先導される．*phon* は音声情報を，*synsem* (syntax - semantics) は構文とその意味に関する情報を意味する．その中は局所的な *loc* (local) と *nonlocal* な情報に分かれる．

$$\left[\begin{array}{ll} \tilde{}type & \\ phon & \langle\,\rangle \\ synsem & \left[\begin{array}{ll} loc & \left[\begin{array}{ll} cat & \left[\begin{array}{l} head \\ subcat \end{array}\right] \\ content & \end{array}\right] \\ nonlocal & \left[\begin{array}{l} \cdots \\ \cdots \end{array}\right] \end{array}\right] \end{array}\right] \tag{1}$$

以下では '$\boxed{\text{i}}$' のようなボックスで囲んだ数字によるインデックスを用いる．これは，このボックスのあとに属性構造を伴う場合，以後，このインデックスでその構造全体を表わし，一つの大きな構造の中で異った場所に現れても常に同じ同じ構造を指すポインタである．またインデックスの後に異る属性構造を伴えば，矛盾のない限りそれらは統合される．

$$\boxed{\text{i}}\left[\begin{array}{ll} f_1 & v_1 \end{array}\right] \equiv \boxed{\text{i}}\left[\begin{array}{ll} f_2 & v_2 \end{array}\right] \equiv \boxed{\text{i}}\left[\begin{array}{ll} f_1 & v_1 \\ f_2 & v_2 \end{array}\right]$$

(1) の構造はそれ全体を構成するような部品の構造を持っていない．以下に，下位構造を明示した構造を示す．ある構造の下位構造は *dtrs* (daughter) という属性で示される．この中の一つがヘッド *head-dtr* (head-daughter; 主となる子構造) である．

$$\left[\begin{array}{ll} \tilde{}type & \\ phon & \langle\,\rangle \\ synsem & \left[\begin{array}{l} \cdots \end{array}\right] \\ dtrs & \left[\begin{array}{l} head\text{-}dtr \\ comp\text{-}dtrs \end{array}\right] \end{array}\right] \tag{2}$$

例えばある文法規則が存在して，伝統的に矢印を用いて以下のように表わされるものとする．

$$\boxed{1}\left[\begin{array}{l} \tilde{}cat1 \\ \cdots \end{array}\right] \rightarrow \boxed{2}\left[\begin{array}{l} \cdots \end{array}\right], \boxed{3}\left[\begin{array}{l} \cdots \end{array}\right].$$

このとき，この文法規則は角かっこを用いて以下のように表わす．ここでは $\boxed{2}$ が $\boxed{1}$ のヘッドであると仮定している．

$$\left[\begin{array}{ll} \tilde{}cat1 & \\ phon & \\ synsem & \boxed{1} \\ dtrs & \left[\begin{array}{ll} head\text{-}dtr & \boxed{2} \\ comp\text{-}dtrs & \langle\boxed{3}\rangle \end{array}\right] \end{array}\right]$$

ただし，本稿においては文法規則であることを明示した方が理解を容易する場合において矢印を用いた記法も併用する．

属性構造のシステムというのは常に (2) であり，その *synsem* の値は常に (1) なのであるが，今後本稿では直接関係ない属性表示を省いて見やすくするために，連続した属性名は縦棒をはさんで '$f_1|\cdots|f_n$,' あるいは '$f_1||f_n$' のように表示する．

2.2 ID-スキーマと原則

自然言語の文法規則は ID-スキーマと呼ばれ，以下の三種類が定義されている．最初はヘッドでない子構造がヘッドの前につくスキーマである．次はヘッドでない子構造がヘッドの後にくるスキーマである．

$$\left[\begin{array}{ll} head & \boxed{1} \\ subcat & \langle\,\rangle \end{array}\right] \rightarrow \boxed{2}, \left[\begin{array}{ll} head & \boxed{1} \\ subcat & \langle\boxed{2}\rangle \end{array}\right].$$

$$\left[\begin{array}{l} head\,\boxed{1} \\ subcat\,\langle\boxed{2}\rangle \end{array}\right] \rightarrow \left[\begin{array}{l} head\,\boxed{1} \\ subcat\,\langle\boxed{2},\boxed{3},\cdots\rangle \end{array}\right], \boxed{3}, \cdots.$$

上記の ID-スキーマの中には共通に以下の二つの原則 (principles) が内在している．最初の**ヘッドフィーチャー原理**:

$$\left[\ dtrs\ \right] \Rightarrow \left[\begin{array}{ll} ||head & \boxed{1} \\ ||head\text{-}dtr||head & \boxed{1} \end{array}\right]$$

では，ヘッドの $head$ 構造が親構造の中の $head$ と同一であることが述べられている．また次の**下位範疇化原理**:

$$\left[\ dtrs\ \right] \Rightarrow \left[\begin{array}{ll} ||subcat & \boxed{2} \\ dtrs & \left[\begin{array}{ll} head\text{-}dtr||subcat & \langle \boxed{1}, \boxed{2} \rangle \\ comp\text{-}dtrs & \boxed{1} \end{array}\right] \end{array}\right]$$

においては，ヘッドが下位範疇化するカテゴリーは，親の下位範疇化するカテゴリーと自身が下位範疇化するカテゴリーを合わせたものであることが述べられている．[1]

3　和音とカデンツの文法

一つの音楽作品とはカデンツの連鎖である．カデンツとは和音がある規則で連続したものである．自然言語解析のアナロジーで言えば，和音の機能がカテゴリーに相当し，具体的な和音は語彙項目に相当する．カデンツはその結果できる句である．以下，われわれは和音を HPSG で形式化し，そのあとカデンツに関する ID- スキーマを提案する．

3.1　音程の計算

和音とカデンツの形式化に先立ち，それらの基礎となる音程の計算のしかたを定義しておく．ここでは例えば「完全五度」や「減七度」と言った音程を短くそれぞれ $mg(5)$，$dm(7)$ などと記述するために以下のように略記法をまとめる．

[1] すべての属性構造はミート '⊓' (あるいは単一化) に関してラティスを構成している．このとき $a \Rightarrow b$ は a の b に対する相対擬補元と呼ばれ，a とミートをとって b の下位に来るような最大の元である．例えば，a の否定は $a \Rightarrow \bot$ と表わすことができる．このラティスに最小元が存在するときはハイティング代数と呼ばれる．

pf	perfect
mj	major
mn	minor
ag	augmented
dm	diminished

ここで mg, mj, mn, ag, dm などは音程の数値を入れると全音と半音の数をペアにして返すような関数であると考えることもできる．音程の半音による移行は次のように表わされる．

$$dm(\{4,5\}) \xleftarrow{\frac{1}{2}} pf(\{4,5\}) \xrightarrow{\frac{1}{2}} ag(\{4,5\})$$

$$\begin{array}{ccc} dm(\{2,3,6,7\}) & \xleftarrow{\frac{1}{2}} & mn(\{2,3,6,7\}) \\ & & \updownarrow \frac{1}{2} \\ ag(\{2,3,6,7\}) & \xleftarrow{\frac{1}{2}} & mj(\{2,3,6,7\}) \end{array}$$

音程のオクターブによる転回は次の約束で決められる．

$$\begin{array}{rcl} pf(n) & \leftrightarrow & pf(9-n) \\ ag(n) & \leftrightarrow & dm(9-n) \\ mj(n) & \leftrightarrow & mn(9-n) \end{array}$$

次の節以降，'$intv$' という関数を仮定する．これは二つの音を与えられたときに，例えば

$$intv(c^1, g^1) = pf(5),$$

のようにその音程を返すものである．

3.2　和音の表現

一つの音に関する情報は，音名，音の長さ，拍の有無などを含むべきである．

$$\left[\begin{array}{ll} \tilde{}note & \\ synsem|loc||head & \left[\begin{array}{l} pitch \\ duration \\ beat \end{array}\right] \end{array}\right]$$

しかしここでは和声の理論に特化するため，音名あるいは音の高さ (pitch) の情報のみに着目する．

和音は単なる複数の音の集合ではない．協和音として認識されるには，各音間の音程に一定の規則がある．まずそのためにわれわれは根音 (root) を見つけ，根音と第三音の音程，根音と第五音の音程を決

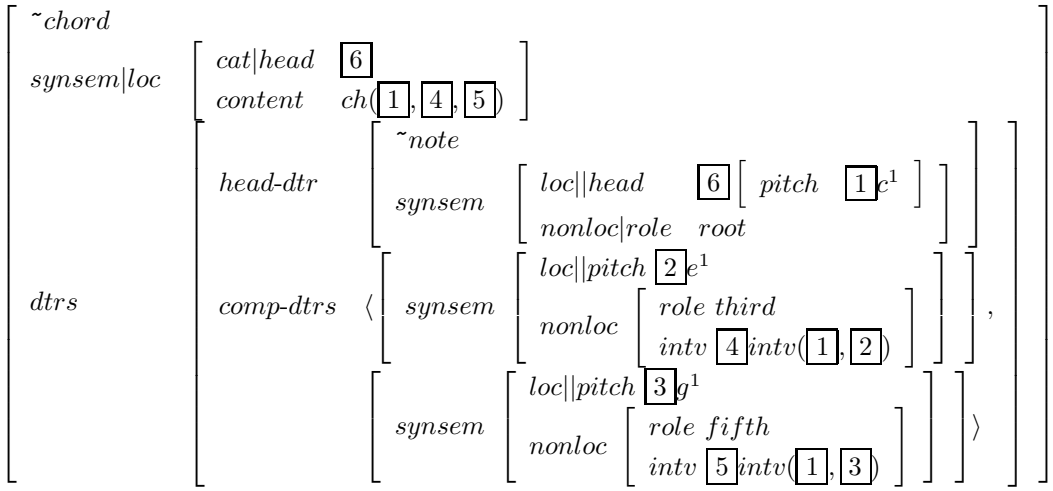


図 1: ハ長調の I の和音

定しなければならない．すなわち和音とは音の集まりの間に与えられた制約である．

まず $C{:}I$ (ハ長調の I の和音) の属性構造による記述例を示す．図 1 では，まず和音は 1, 2, 3 の三つの音からなっている．ここで根音は c^1 であり，第三音，第五音としてそれぞれ e^1 と g^1 を伴っている．これらは子構造として埋め込まれている．根音とこれらの間の音程は 4 および 5 において関数 '*intv*' によって計算されており，ここではそれぞれ長三度 ($mj(3)$) と完全五度 ($pf(5)$) となっている．和音の名前は根音とこの二つの音程によって決定されるため，1 および 4 と 5 が名前づけの関数 '*ch*' に渡される．全体のヘッドの情報は根音の 6 の情報と共有されている．

3.3 カデンツの文法

カデンツは音楽の中の句である．一般にはカデンツとは次のような三種類の和音の連なりである．

- T-D-T
- T-S-T
- T-S-D-T

ここで T (トニック), S (サブドミナント), D (ドミナント) はそれぞれ I, IV, V の和音を示す．[2] しかしながらこれらの和音は II, III, VI, VII のような副和音で代用されることもあるため，T, S, D は和音の機能の名前とし，実際の和音とは区別する．以下にこの副三和音を考慮して和音の機能と具体的な和音の間の制約関係を示す．

$$\left[\ ||function \quad T\ \right] \Rightarrow \left[\ ||content \quad \mathrm{I} \sqcup \mathrm{VI}\ \right] \tag{3}$$

$$\left[\ ||function \quad D\ \right] \Rightarrow \left[\begin{array}{ll} ||content & \mathrm{V} \sqcup \mathrm{III} \\ & \sqcup \mathrm{VII} \end{array}\right] \tag{4}$$

$$\left[\ ||function \quad S\ \right] \Rightarrow \left[\ ||content \quad \mathrm{IV} \sqcup \mathrm{II}\ \right] \tag{5}$$

ここで '$\sqcup$' は構造のジョイン (join) 操作を表わす．上記の場合に限り，この操作は排他的な 'or' となる．

この節の目的は，以下のようなカデンツを産み出す生成規則を定義することである．

$$\overline{T} \ \rightarrow \ S,\ \overline{T}. \tag{6}$$

$$\overline{T} \ \rightarrow \ D,\ T. \tag{7}$$

$$\overline{T} \ \rightarrow \ T. \tag{8}$$

T に付随する任意回数の $\overline{T}$:

$$\overline{\overline{T}} \ \rightarrow \ T,\ \overline{T}^{*}$$

は次のような和音列を生成する．

{T-D-T, T-S-T, T-S-D-T, T-D-T-D-T, T-S-T-D-T, T-D-T-S-D-T, ···}.

[2] 音楽はカデンツの連なりであるが，それを終わらせるには次の四つの特殊な形のカデンツが定義されている．(IV-)V-I (完全終止), V-VI (偽終止), I-V (半終止), and IV-I (プラガル終止).

$$\left[\begin{array}{ll} \tilde{}cadence & \\ synsem||head & \boxed{1} \end{array}\right] \rightarrow \left[\begin{array}{ll} \tilde{}chord & \\ synsem||function & S \\ ||head\text{-}dtr||pitch & \boxed{2} \end{array}\right], \left[\begin{array}{ll} \tilde{}chord & \\ synsem & \left[\begin{array}{ll} head & \boxed{1} \\ subcat & S \end{array}\right] \\ ||head\text{-}dtr||pitch & \boxed{2} - pf(4) \end{array}\right]. \quad (6)'$$

$$\left[\begin{array}{ll} \tilde{}cadence & \\ synsem||head & \boxed{1} \end{array}\right] \rightarrow \left[\begin{array}{ll} \tilde{}chord & \\ synsem||function & D \\ ||head\text{-}dtr||pitch & \boxed{2} \end{array}\right], \left[\begin{array}{ll} \tilde{}chord & \\ synsem & \left[\begin{array}{ll} head & \boxed{1} \\ subcat & D \end{array}\right] \\ ||head\text{-}dtr||pitch & \boxed{2} + pf(4) \end{array}\right]. \quad (7)'$$

$$\left[\begin{array}{ll} \tilde{}cadence & \\ synsem||head & \boxed{1} \end{array}\right] \rightarrow \left[\begin{array}{ll} \tilde{}chord & \\ synsem||head & \boxed{1} \end{array}\right]. \quad (8)'$$

カデンツを決める規則の中で共通するのは二つの和音の根音の音程が完全四度であることである．すなわち，先行する和音の根音と後続の和音の根音が四度上行するか四度下行するかであると，その連続した和音は音楽的に妥当であると解釈される．このことを鑑みて ID-スキーマを以下のように定義する．

以上すべてのルールで *head* 属性

$$\boxed{1}\left[\begin{array}{ll} function & T \end{array}\right]$$

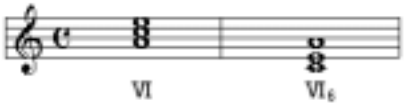
は共通である．ある和音が上記 ID-スキーマのボディ部の一つと単一化可能であるためには，その和音は機能の制約のいずれかを満さなければならない．

4 制約充足

本研究では上記 HPSG の実装を行うために LiLFeS [9] をパーサとして用いる．LiLFeS は効率的な単一化機能を持ち，メモリ管理の最適化メカニズムも持つ．パーサのエンジンは属性構造を処理し確定節を実行する抽象マシンである．本稿で提案している音楽の解析システムの枠組みでは二つのフェーズで複雑な制約を解く必要がある．その例を以下に示し，LiLFeS による実装の可能性を最後に論じる．

(1) の構造表示で見たように，和音は複雑なインデックスのネットワークである．このことからただちに和音の同定は直線的な処理では不可能であることが示唆される．

例 1 (和音の同定) 音の集合 $\{a^1, c^2, e^2\}$ は $C{:}\,VI$ という和音を構成する．しかしこれが転回形 $\{c^1, e^1, a^1\}$ で現れた場合，バス音をただちに根音だと思う方法ではうまくいかない．第 3 節で提示した音程計算に基づき，われわれは $\{c^1, a^1\}$ の長六度の代わりに $\{a^1, c^2\}$ の短三度，$\{a^1, e^1\}$ の完全四度の代わりに $\{a^1, e^2\}$ の完全五度を発見することにより $C{:}\,VI_6$ の和音を同定することができる．



機能和声の制約の下では，この和音は必然的にトニックの機能を負う．$a \sqsupseteq b$ は，a が b に向かって単一化可能 $(a \sqcap b = b)$ であるとする．この例では *dtrs* 属性の値: $\langle a^1, c^2, e^2\rangle$ が分解されて *head-dtr* と *comp-dtrs* の二つの属性の値に分離されている．

$$\left[\begin{array}{ll} dtrs & \langle a^1, c^1, e^1\rangle \end{array}\right]$$

$$\sqsupseteq \left[\begin{array}{ll} \tilde{}chord & \\ dtrs & \left[\begin{array}{ll} head\text{-}dtr & a^1 \\ comp\text{-}dtr & \langle c^2, e^2\rangle \end{array}\right] \end{array}\right]$$

$$\sqsupseteq \left[\begin{array}{ll} \tilde{}chord & \\ synsem||function & T \\ dtrs & \left[\begin{array}{ll} head\text{-}dtr & a^1 \\ comp\text{-}dtr & \langle c^2, e^2\rangle \end{array}\right] \end{array}\right]$$

同様な複雑な制約充足の例はカデンツの認識にも起こる．

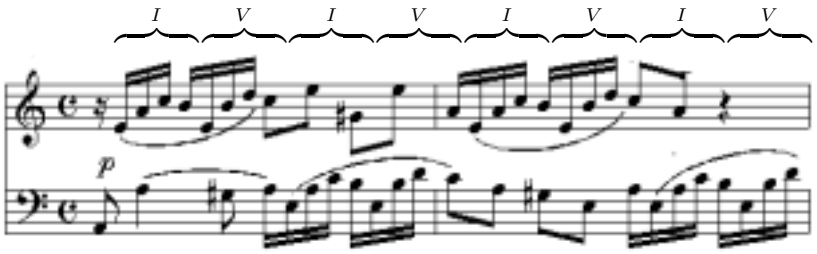


図 2: カデンツの認識

例 2 (カデンツの認識) 以下の例 (J. S. バッハ: インヴェンション第 13 番 イ短調) では各拍毎に I の和音と V の和音を見い出すことができる (図 2). しかしながら曲の途中などで調号が表示されない場合には, 'I-V' の上行と 'IV-I' の上行は区別がつかず, したがって調はこの後に続く和音が来るまで決定されないことになる. このようにカデンツの同定は決定的なプロセスではなく, ここでも制約を非直線的に解く実行パラダイムが必要になる.

5 結語

本論文では和声学を HPSG で記述する試みを行った. この文法理論により, 文法規則が階層的な表現に使えるだけでなく, それぞれのカテゴリーが属性構造であることより制約の表現にも優れ, かつヘッドの概念がより適切に音楽知識を記述できることを示した.

本論文で提案した音楽解析システムの構想は本質的に双方向的である. すなわち, パーサとして定義した文法規則を逆向きに使うことにより, 音楽の生成システムとしても利用可能である. ただし, 本システムが解析対象とするのは音の集まりの列に対してであり, それを和音と認識し, その並びの合法性を検討するところまでである. すなわち, 音の集まりを分散させた通常の音楽ではグルーピングにより, 音の集まりを生成する前処理が必要であり, これは将来の GTTM の自動グルーピングシステムに多いに期待がかかるところである.

参考文献

[1] Lerdahl, F. and Jakendoff, R. *A Generative Theory of Tonal Music.* MIT Press, 1983.

[2] 平田圭二, 青柳龍也. パーピープン: ジャズ和音を生成する創作支援ツール. 情報処理学会論文誌, vol.42, no.3, pp.633–401, 2001.

[3] Aït-Kaci, H and Nasr, R. LOGIN: A logical programming language with built-in inheritance. *Journal of Logic Programming*, vol.3, pp.187–215, 1986.

[4] Carpenter, B. *The Logic fo Typed Feature Structures.* Cambridge University Press, 1992.

[5] Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. *Generalized Phrase Structure Grammar.* Oxford: Basil Blackwell, 1985.

[6] Sells, P. *Lectures on Contemporary Syntactic Theories.* CSLI publications, 1985.

[7] Pollard, C., and Sag, I. A. *Information-based Syntax and Semantics, Vol.1.* CSLI publications, 1987.

[8] Pollard, C., and Sag, I. A. *Head-driven Phrase Structure Grammar.* University of Chicago Press, 1994.

[9] Miyao, Y., Makino, T., Torisawa, K., and Tsujii, J. The LiLFeS Abstract Machine and its evaluation with LinGO grammar. *Natural Language Engineering*, vol.6, part.1, 2000.