4F−6

語彙機能文法の横型構文解析 —否定演算を含む属性構造記述—

森永 正信, 冨浦 洋一, 中村 貞吾, 日高 達
九州大学

1 はじめに

LFG(Lexical Functional Grammar) は, CFG(Context Free Grammar) の生成規則に注釈として機能スキーマと呼ばれる式を付加し, 規則の適用に制限を与えたものである. 従来の LFG の機能スキーマで記述可能なことは肯定的関係のみであったが, より柔軟に自然言語の文法を記述するために否定的関係の記述が望まれる.

既に, 属性構造に否定を導入することが幾つかの論文で試みられている. 本論文では, 基底要素という特別な要素の集合として属性構造を明確に定義し, この基底要素の集合上で否定的関係を記述した. そして, 機能スキーマを基底要素に関する連立方程式とみなし, これを解くことによって属性構造が求められることを示した.

2 属性構造

はじめに基底要素 (Ground Instance) と, 属性構造 (Feature Structure) について, 定義を行なう.

基底要素 (GI) は, 以下の定義で再帰的に与えられるもののみである. ただし, $A = \{a_1, a_2, \cdots, a_M\}$ をアトムの集合, $L = \{l_1, l_2, \cdots, l_N\}$ を属性名であるラベルの集合とする.

$$a \qquad (a \in A) \tag{1}$$

$$l_1 : g_1 \wedge l_2 : g_2 \wedge \cdots \wedge l_N : g_N \qquad (l_i \in L,\ g_i \in GI,\ i = 1, \cdots, N) \tag{2}$$

基底要素とは, 全ての属性名の値が明確に定められたものである.

属性名のラベル $l_i \in L$ は, 基底要素 x を引数として, その属性名 l_i の値である基底要素を返す関数とみなすことができる. すなわち, 式 (2) で示される基底要素 x に対して, 以下の式が成り立つ.

$$l_i(x) = g_i \tag{3}$$

属性構造 (FS) は, 以下の定義で再帰的に与えられるもののみである.

$$a \qquad (a \in A) \tag{4}$$

$$l : \Phi \qquad (l \in L,\ \Phi \in FS) \tag{5}$$

$$\Phi \wedge \Psi \qquad (\Phi, \Psi \in FS) \tag{6}$$

属性構造とは, ある基底要素を示すために特定の属性名とその値だけを明示したものである. ここで, 関数 $S(\cdot)$ を用いて, 属性構造が示す基底要素の集合を定義する.

$$S(a) = \{a\} \qquad (a \in A) \tag{7}$$

$$S(l : \Phi) = \{x \mid l(x) \in S(\Phi)\} \qquad (l \in L,\ \Phi \in FS,\ x \in GI) \tag{8}$$

$$S(\Phi \wedge \Psi) = S(\Phi) \cap S(\Psi) \qquad (\Phi, \Psi \in FS) \tag{9}$$

3 連立方程式の解としての属性構造

機能スキーマの集合を連立方程式とみなした F-記述を解くことにより属性構造を求める方法を示す.

3.1 F-記述

機能スキーマは, 基底要素同士の満たすべき関係を記述するものである. 機能スキーマで二つの基底要素が異なることを記述するのに $\neq$を用いる. $\neq$ は $=$ の純粋な否定である.

$\neq$の導入によって拡張した機能スキーマを以下の定義で示す. 機能スキーマは以下のいずれかの形式をした式の有限集合である. ただし, Mは各節点を示すメタ記号 $\uparrow$, $\downarrow$と, 局所的に使用する変数の集合である.

A Breadth First Parsing for LFG — A Feature Description with A Negative Operation —
Masanobu MORINAGA, Yoichi TOMIURA, Teigo NAKAMURA, Toru HITAKA
Kyushu University

$$(x\ \alpha) = (y\ \beta) \qquad (\alpha, \beta \in L^*,\ x, y \in M) \qquad (10)$$
$$(x\ \alpha) = a \qquad (\alpha \in L^*,\ a \in A,\ x \in M) \qquad (11)$$
$$(x\ \alpha) \neq (y\ \beta) \qquad (\alpha, \beta \in L^*,\ x, y \in M) \qquad (12)$$
$$(x\ \alpha) \neq a \qquad (\alpha \in L^*,\ a \in A,\ x \in M) \qquad (13)$$

C-構造中の各節点の機能スキーマを全て集め，連立方程式としたものをF-記述（Feature Description）と定義する．ただし，機能スキーマ中のメタ記号を，各節点を示す基底要素の変数に置き換える．

3.2 F-記述の標準化・簡約化

F-記述は次の形式のみからなる等価な式に変換できる．これをF-記述の標準型と定義する．ただし，Vは基底要素の変数の集合である．

$$x = y \qquad (x, y \in A \cup V) \qquad (14)$$
$$(x\ l) = y \qquad (x, y \in A \cup V,\ l \in L) \qquad (15)$$
$$x \neq y \qquad (x, y \in A \cup V) \qquad (16)$$

F-記述Eにおいて基底要素xとyが等しくないことを，$E \models x \neq y$で定義する．以下の場合に$E \models x \neq y$である．

1. $x, y \in A$であり，$x \neq y$である場合．
2. $x \in A$であり，$(y\ l) = v \in E$の場合．
3. $x < y$または$y < x$の関係が導かれる場合．
（ただし，$(z\ l) = v$という式に対し，$z < v$である．）
4. $(x\ l) = v_1$，$(y\ l) = v_2$であり，$E \models v_1 \neq v_2$である場合．

標準型のF-記述Eを，次に示す簡約化I，簡約化IIの操作を行うことによって，さらに等価な式に変換できる．

F-記述Eの簡約化Iとは，E中で同値関係の変数をある一つの変数に代表させる操作である．

また，F-記述Eの簡約化IIとは，$x \neq y \in E$であり，$E - \{x \neq y\} \models x \neq y$の場合に，$E - \{x \neq y\}$を新たな$E$とする操作である．

3.3 解の存在条件

標準化・簡約化されたF-記述が，以下の条件を満たしていれば具体的な基底要素を求めることができる．これはF-記述の解の存在条件である．

1. $a, b \in A$かつ$a \neq b$であり，$a = b \notin E$である．
2. $x \in V$であり，$x \neq x \notin E$である．
3. 全ての変数に対して，$x < y$という半順序関係が付けられる．

3.4 一般解の導出

F-記述Eが解の存在条件を満たしていれば$(x\ l_{x_1}) = y_{x_1}, (x\ l_{x_2}) = y_{x_2}, \cdots, (x\ l_{x_n}) = y_{x_n}$の形式の式から，基底要素$x = l_{x_1} : y_{x_1} \wedge l_{x_2} : y_{x_2} \wedge \cdots \wedge l_{x_n} : y_{x_n}$が求まる．ただし，記述の無い属性名の値は任意定数であると考える．

また，E中に$x \neq y$の形式の式があれば，これらは全て求めた基底要素の任意定数に対する制約条件となる．

ここで，解の存在条件を満たしているF-記述に対する変数と任意定数を含む基底要素，および，それに対する制約条件の形式をF-記述の一般解と定義する．

4 横型構文解析への応用

F-記述Eにおいて，次の重要な定理が成り立つ．ただし，$\wp(E)$はF-記述Eに標準化と簡約化の操作を行なった後の式集合を示し，$\equiv$は二つのF-記述が等価であることを示す．

1. $E = E_1 \cup E_2$ならば，$E \equiv \wp(E_1) \cup \wp(E_2)$である．
2. E'（$E' \subseteq E$）が解の存在条件を満たしていなければ，Eは解を持たない．

すなわち，C-構造のある部分木に対する属性構造は独立に解くことができ，C-構造全体の属性構造は，それら部分木の一般解から求めることができる．また，ある部分木のF-記述が解の存在条件を満たしていなければ，C-構造全体の解も存在しないことが判定できる．

これらの特性により，F-記述解法の横型構文解析への適用は容易である．

5 まとめ

属性構造を基底要素の集合として定義したことで，属性構造同士の関係を明確に記述できるようになった．また，アトムの否定のみならず，構造を有する素性自体を否定した機能スキーマを用いて，LFGの構文解析が可能になった．

参考文献

[1] 溝口 英樹，否定的関係の記述を含む語彙機能文法の横型構文解析，九州大学大学院総合理工学研究科修士論文，平成4年3月

[2] Anuj Dawar, K.Vijay-Shanker, An Interpretation of Negation in Feature Structure Descriptions, Computational Linguistics, Vol.16, March 1990