係り受け制約を表現する文脈自由文法への文節文法の組み込み

松井裕二　田辺利文　冨浦洋一　日高 達
(九州大学大学院 システム情報科学研究科)

1 はじめに

自然言語処理における構文解析では，一般に入力文に対応する構文構造が複数存在し，それらからどのようにして構文構造を選択するかが問題点の一つである．構文構造の中には意味的に不適格なものも含まれるため，意味処理による意味的に適格な構文構造の絞りこみが重要になる．実用的な意味処理の構文解析への導入として係り受け制約を用いる方法がある．

意味的に適格な係り受け関係においては、係る句のhead word (その句の主要な意味を担う語) と係られる句の head word と係りの種類の3者の間に一定の制約がある。係り受け制約とはこの制約のことで、これを表現するために統語範疇をそれから導出される句のhead word と function(係りの種類を規定する語) で細分した係り受け文脈自由文法を提案している．

これを日本語に適用する場合,function になり得る語は付属語列中の1つの語 (助詞や活用語尾など) であるため、どれがfunction になるかを CFG の生成規則として記述する必要がある．付属語列の語の並び方については隣接語間の統語制約を基にした文節文法があり，付属語列中のどの付属語がfunction になるかには一定の規則性があることが分かっている，これらを利用した，function の決定機構を含む日本語文法の構成法を提案し，文節に対する構文解析の実験結果を報告する．

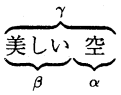
2 係り受け制約と文脈自由文法への組み込み

2.1 係り受け制約

句γは，句αと，αを修飾するいくつかの句$\beta_1 \cdots \beta_l$から構成されるものとすると，句αは句γ全体の意味を代表する句である．ここで句αを句γの head phrase と定義する．

句が他の句を修飾する時には一般に修飾する方の句の中にその修飾の種類を規定する情報があり，これを句の function と定義する． function としては日本語文では助詞が示す係りの種類や文節末の活用語の活用形が挙げられる．

例えば、句「美しい空」は，



のように表現されるので，句「空」は句「美しい空」の head phrase であり，句「美しい」が句「空」を連体形で修飾しているので，句「美しい」の function は「連体形」であることが分かる．

Xを root node に持つ部分木において，Xの head phrase がαである場合

- αが終端記号の時，α
- αが非終端記号の時，αを root node とする部分木の head word

を X の head word と定義する．head word は，その句の意味を代表する語になる．ここで，

$$X \longrightarrow Y_1 \cdots Y_{i-1} \quad Z \quad Y_i \cdots Y_l \qquad (1)$$

を頂点からの書き換えに適用した Xを頂点とする構文木において，Y_i の head word が w_i, Y_iの function が f_i，Zの head word が wであるとき，w_iは f_iを介して wに構造的に係っている（構造的な係り受け関係にある）と定義する．

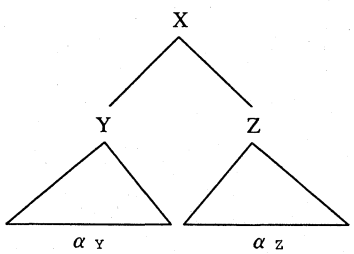


図 1: 文脈自由文法による構文木

図1の文脈自由文法の構文木において，句α_Y と句α_Zの間に意味的に適格な修飾関係が成立しているも

のとすると，Yの head word と Yの function 及び Z の head word にはある一定の意味的な制約（係り受け制約）が成立している．構造的な係り受け関係のうち，修飾句の head word，function，被修飾句の head word が係り受け制約を満足しているものを，意味的に適格な係り受け関係と呼ぶ．

2.2 係り受け制約の文脈自由文法への組み込み

文脈自由文法においては，構文木の任意の生成規則の右辺の異なった非終端記号から導出される単語どうしはお互いに独立である．そのため係り受け制約を生成規則の形で表現できず，意味的に不適格なものを導出する可能性がある．これを防ぐためには，それぞれの非終端記号からの導出が他方の非終端記号からの導出を束縛できるような機構を生成規則に設けるとよい．これは係り受け制約を生成規則の形で表現することを意味する．

そのために，従来用いられてきた統語範疇をその句の head word と function で細分化したものを非終端記号とする．

$$X(h) \longrightarrow Y_1(-h)\cdots Z(h)\cdots Y_l(-h) \qquad (2)$$

$$Y_j(-h) \longrightarrow Y_j(h_j, f_j) \qquad (3)$$

また、修飾句となり得る句を構成する生成規則

$$X \longrightarrow Y \quad F$$

を

$$X(h,f) \longrightarrow Y(h) \quad F(f) \qquad (4)$$

と改める．ここで各非終端記号は以下のような意味を持つ．

$X(h,f)$ head word が h であり，function が fである統語範疇 Xの句を導出する非終端記号

$X(-h)$ head word が h である句に係り得る統語範疇 Xの句を導出する非終端記号

$X(h)$ head word が h である統語範疇 Xの句を導出する非終端記号

$X(f)$ function が fである統語範疇 Xの句を導出する非終端記号

生成規則（3）は，h_jが f_jを介して h に係り得る，つまり係り受け制約を表している．

従来の非終端記号の統語範疇の取り方では意味的に適格でないものも導出されていたが，head と function で非終端記号をさらに細分化することで意味的に適格でないものを導出しなくなる．

3 日本語文法への適用

係り受け制約文脈自由文法を日本語に適用する方法について述べる．2章で係り受け文脈自由文法ついて述べたが，それを実現するためには生成規則 (4) における function が適切に選定される必要がある．文節内の単語間の接続可能性は前の単語の品詞・活用とそれに続く単語の品詞情報によって決定されるため，文節を導出する非終端記号を B, 品詞 Fの付属語[1] が先頭である付属語列を導出する非終端記号を F, 自立語 (用言の場合はその語幹) を b, 付属語を wとして文節内における文法を正規文法で表現すると次のようになる。

$$B \longrightarrow b \quad F \qquad (5)$$
$$B \longrightarrow b \qquad (6)$$
$$F \longrightarrow w \quad F' \qquad (7)$$
$$F \longrightarrow w \qquad (8)$$

(5) は, 自立語 (あるいは自立語の語幹)b と品詞 Fの語は接続可能であり,(6) は,b が文節末尾になりえ, (7) は, 付属語 wの品詞が Fで wと品詞 F'の語が接続可能であり,(8) は, 品詞 Fの語 wが文節末尾になりえることを意味する．

日本語においては head phrase は最後尾に位置するので,(4) の生成規則の $Y(h)$ の末尾の語は h である.$F(f)$ はそれに続く付属語列であるから, 語 h と品詞 Fの語が接続可能でなければならない．これは (5) に対応する．

例：PP(人, が) $\longrightarrow$ NP(人)　副助詞【だけ】(が)

従って生成規則 (4) の右辺以降を導出する生成規則は，生成規則 (6)(7) 及び (8) をそれぞれ細分化することで与えられることになり，文節文法を組み込んだ係り受け文脈自由文法を実現することが出来る．これは係り受け文脈自由文法を日本語に適用したものである．

細分化は基本的には以下のようにして行なう．

(6) は,

$$X(h) \quad \longrightarrow \quad h$$

と改める．但し X は h の統語範疇である．

例：名詞句 (人) $\longrightarrow$ 人

[1]活用語尾も付属語として扱う．またここで品詞とは, 各付属語毎に設定された品詞 (一単語一品詞) である．ただし活用語尾に関しては品詞に活用型の情報も含める．例えば「を」の品詞は単に助詞ではなく “格助詞【を】” であり, 助動詞「だ」やその連用形「で」の品詞は “助動詞【だ】” であり,「歩く」の活用語尾の「き」や「く」の品詞は “カ行五段動詞語尾” である．

(7) は

$$F(f) \longrightarrow w \quad F'(f) \tag{9}$$

あるいは

$$F(\tilde{w}) \longrightarrow w \quad F'(f) \tag{10}$$

と細分化する. 但し,(10) において$\tilde{w}$は

$$\tilde{w} = \begin{cases} w & ;w\text{が助詞} \\ w\text{の活用形} & ;w\text{が活用語} \end{cases}$$

(8) は,

$$F(\tilde{w}) \longrightarrow w \tag{11}$$

と改める.(7) を細分化したとき,(9) とするか (10) とするかは,function としての f と$\tilde{w}$の強さ及び前後関係により決まる. これを以下のように考察して決定した.

(a) 文節末尾が活用語ならば, その活用形が function.

(b) (a) 以外の場合, 助詞が function となる. 格助詞単独の場合, 接続助詞単独の場合, それらが function になる. 副助詞は本来, 係りの種類を規定しないが, 副助詞単独ならば格助詞の代用の機能を持つ.

- 「彼にだけは負けたくない」…「彼」は「に」を介して「負け (る)」に係る.
- 「彼だけ負けた」…「だけ」は「が」あるいは「に」の代用.

(c) 格助詞が複数, 格助詞と接続助詞が付属語列中に存在するとき, 後方のものが function になる.

- 「東京でが良い」…「東京」は「が」を介して「良い」に係る.
- 「A と B とを結ぶ直線」…「B」は「を」を介して「結ぶ」に係る.

fが活用形, あるいは f, wが助詞で $f \geq \tilde{w}$のときは (9) の生成規則, f, wが助詞で,$\tilde{w} > f$のとき (10) の生成規則となる。但し, 接続助詞＝格助詞＞副助詞 である.

4 実験

3 で述べた文法を確率化し、文節の構文解析を行なった。文節によっては、function, 語の区切れ、品詞 (活用型) により複数の構文木が得られる場合がある。構文解析の結果から、それらの選択が正しく行なわれているかを確認する。

200 個の文節を構文解析し、構文木の生起確率の高いものから何番目までに正しい構文木が得られたかの累積正解率は以下の通りとなった。入力文節は、あいまいになるように自立語を平仮名にした. なお、200 個の文節のうち 166 個で複数の構文木が得られ、1 文につき平均 3.4 個の構文木が得られた。

	正解数	正解率 (%)
1	180	90
2	191	95.5
3	198	99
4	200	100

:入力文節数 200

5 おわりに

今回は、日本語係り受け文脈自由文法を定義し、単文節の構文解析を行ない、高い正解率を得た。今後の課題として、日本語係り受け文脈自由文法全体の評価実験が望まれる。

参考文献

[1] 日本電子化辞書研究所: EDR 電子化辞書仕様説明書, 1995

[2] 田辺利文:係り受け制約の文脈自由文法への組み込み法, 九州大学システム情報科学研究科報告,Sep.1996

[3] 渡辺健一郎:付属語列文法に対する一考察, 九州大学工学部学士論文,1996

[4] 日高達: 確率文法, 情報処理学会学会誌, 1995,3