

ボトムアップ構文解析システムBUPでの左外置変形の処理

田中 穂積、 今野 聡、 高倉 伸 (東工大 工)

1. まえがき

Prologによる自然言語処理の研究として、筆者らはETLとICOTで開発されたボトムアップ構文解析システムBUP [1]を用いて比較的大きな英語の文法を作成した [2]。文法の記述はDCG [7]でおこなったが、Wh疑問文や関係節に関する規則は名詞句の欠けた動詞句や疑問文といったカテゴリをもうけて記述していた。英語などの言語では、疑問文や関係節などを左外置変形 (Left extra position) をもちいて説明することができる。左外置変形とは、あるカテゴリの構成要素の一つがそのカテゴリの左方に移動する際、移動する前の場所に痕跡 (trace) を残すという変形である。

I know a man that [_s she loves t_i].
t_i: trace

図1-1

これまで、この左外置変形を扱えるGrammar formalismとして開発されたものに、PereiraによるXG (Extrapolation Grammar)がある [3]。このXGは、DCGを拡張したもので、実行はトップダウン、デプスファーストで行なわれるため、左再帰規則を扱うことができないといったトップダウンパーザーの欠点を持っていた。

本報告では、この左外置変形をBUP上で取り扱うために行なったBUPの拡張に関する基本的な考え方や問題点について述べる。以後、このBUPの拡張版をBUP-XGと呼ぶ。

2. DCGによる文法記述

関係節に関する文法規則をDCGで記述した例を図2-1に示す。この例で分かるように、DCGで記述する場合には、主語の欠けた規則 (1), (2) や、目的語の欠けた規則 (3) を一つ一つ書かなくてはならない。特に、目的語が欠けている場合には、vp2のように、新しい文法カテゴリを新たに設け、本来の他動詞句の規則 (4), (5) から目的語が欠けている規則 (6), (7) を記述する必要がある。また、主語が欠けている場合、動詞と先行詞間

の文法的な一致を調べるには、動詞の文法情報をsrelを通して上に上げてやり、先行詞の文法情報と比較しなくてはならない。

- srel --> relpro, (aux), vp. (1)
- srel --> relpro, (aux), bep, pred. (2)
- srel --> relpro, subj, (aux), vp2. (3)
- vp --> v, (vt_obj), obj. (4)
- vp --> v, (vt_obj2), obj1, obj2. (5)
- vp2 --> v, (vt_obj). (6)
- vp2 --> v, (vt_obj2), obj1. (7)

図2-1 DCGによる文法記述例

3. 左外置変形処理の基本的考え方

BUP-XGにおける左外置変形の処理は、基本的に、ATNで関係節の処理に使用されているHoldレジスタを用いたHold-Unholdメカニズム [4] による処理と同等であるとみなすことができる。BUP-XGでは、このHoldレジスタに対応するものとして、Xlistと呼ばれるスタックを使用している。本章では、まずトップダウンで解析する場合について述べ、次にボトムアップで (BUP-XGで) 解析する場合について述べる。

3.1 トップダウン法による左外置変形の処理

まえがきで述べたように、左外置変形を扱えるトップダウン解析システムとしてPereiraのXGがあるが、本節ではBUP-XGの説明との整合性からBUP-XGで使用されているsyntax sugarを用いて説明する。

図1-1の例文を解析するためのDCGによる文法規則及び辞書項目を次ページ図3-1に示す。

文法カテゴリでsrel, relproはそれぞれ関係節、関係代名詞を表わすカテゴリである。

npの規則 (11) で". . . /" は新たに導入したsyntax sugarで、トランスレータは2つの文法カテゴリを引数として持つslash(srel,np)というシステムカテゴリ (例ではslash(srel,np)) であると解釈する。srel../np (slash(srel,np)) が実行されると、スタックにカテゴリnpがプッシュされ、そのあと

- s --> np, vp. (8)
- vp --> v, np. (9)
- np --> pron. (10)
- np --> a, noun, srel../np. (11)
- srel --> relpro, s. (12)

- pron --> (i), pron --> (she).
- a --> (a), noun --> (man).
- v --> (know), v --> (love).
- relpro --> (that).

図 3-1

で s r e l の解析が行なわれる。又、トランスレータは、(11)の規則で n p がスタックにプッシュされることから、自動的に次の(13)のような n p に関する規則を生成する。

- np --> virtual(np). (13)

この規則の右辺に現われる virtual は、スタックトップが n p である場合に、スタックをポップして n p の解析を成功させるシステムカテゴリである。このカテゴリが見つかることは、言語学的に見ると、前方に移動したカテゴリが残した痕跡を見つけたことに相当している。

s をゴールとしてトップダウンに解析を始め、今、"a man that she loves" を n p をゴールとして解析しようとしているとする。図 3-2 は解析が進む様子を示したものである。まず、(10)が選択されるがすぐに失敗し、次の(11)が選択される。すると、a, noun が、それぞれ、"a", "man" で成功し、次の s r e l../np を新しいゴールとして実行する。s r e l../np の実行は、まず、n p をスタックにプッシュし、次に改めて s r e l をゴールとして実行する(図に*で示す)。以後、再び通常のトップダウンの解析を続けるが、"loves" の目的語が見つからないため、n p の規則(10)、(11)は失敗し、次に(13)を選択する。このときスタックトップに n p があるため、それをポップして(13)は成功する(図に**で示す)。このように、トップダウン法では、(6)のような規則を追加し、これを実行することでスタックのポップをおこなう。

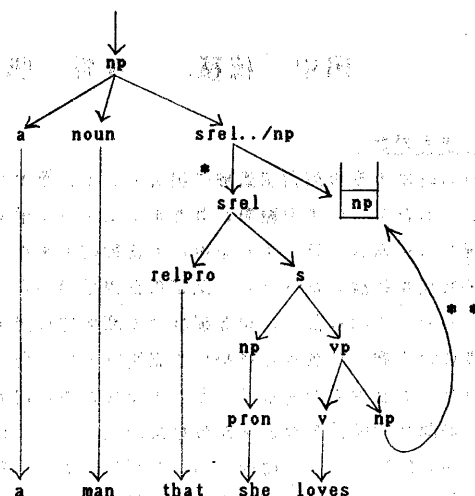


図 3-2 トップダウンによる実行例

以上、左外置変形の処理を行なうトップダウン解析の簡単な例を示した。

3.2 ボトムアップ法 (BUP-XG) による左外置変形の処理

ボトムアップ法で左外置変形の処理を行なう場合も、トップダウン法と同様にスタックを使用する。BUP-XGでは、図 3-1 の規則は、次章で述べるトランスレータにより、Xlist のついた BUP 節に変換される。その際トップダウン法の場合のように、スタックをポップするための規則(13)が付加されることはない。

トップダウン法と BUP-XG との違いは、スタックをポップするタイミングにある。これは、BUP の基本的な動作メカニズムによるものである。通常、BUP システムでは、図 3-3 に示すように、DCG で記述された規則(14)をトランスレータ [5] によって BUP 節(15)に変換する。実行に際しては、始めに入力文の最初の単語の辞書引きを行ない、その単語の文法カテゴリが左辺の述語名となっている BUP 節を選択する。これにより、変換前の規則において右辺の最初の文法カテゴリとなっている規則を選択し、第 2 カテゴリ以降を、次々にサブゴールとして立てて解析を行なうことになる。尚、(15)の goal 節は BUP システムに組み込まれているものを用いる。

a(ARG) --> b(ARG1),c(ARG2),d(ARG3). (14)

↓変換

b(G, (ARG1), I) --> {link(a,G)}, (15)

goal(c, (ARG2)),

goal(d, (ARG3)),

a(G, (ARG), I).

図3-3 DCGからBUP節への変換例

BUP-XGでは、スタックのポップが行なわれるタイミングに二つの場合がある。第一は、スタックトップにあるカテゴリがゴールとして呼ばれた時で、この場合には、スタックをポップして単に成功させればよく、トップダウンの場合と基本的に同じである。第二は、スタックトップにあるカテゴリが文法規則の右辺の第1カテゴリの場合である。BUPシステムの基本メカニズムより分かるとおり、BUPでは、DCGによる文法規則の右辺の第1カテゴリ(図3-3のb)は、辞書引き又はボトムアップ解析の結果できるカテゴリであり、これらがゴールとして呼ばれることはない。したがって、右辺の第1カテゴリに相当する単語列が欠けている場合には、スタックトップのカテゴリを辞書引きによって見つけたカテゴリとして扱う必要がある。この例として主語の欠けた関係節を解析する場合がある。これについては次章で詳しく説明する。

4. インプリメンテーション

本章では、左外置変形を扱うためにBUPシステムに施した改良点について述べる。

4.1 Xlist

Xlistは以下に示すような3つの要素を持つ関数xの構造体である。

x(category, argument, xlist)

ここで、argumentは、DCGで記述された文法規則で各categoryに付加されていた引数のリストである。又、xlistは、プッシュする前のXlistで、これによりスタックが実現されている。空スタックは、[]で表わされる。例として、初めてnp(NP_A, NP_S, NP_T)をプッシュしたスタックを表わすXlistを示すと次のようになる。

x(np, [NP_A, NP_S, NP_T], [])

1.2 BUP-XGトランスレータ

従来のBUPトランスレータは、図3-3に示したようにDCGで記述された文法規則を、BUP節に変換するものであった。それに対し、BUP-XGトランスレータは、変換後の全ての述語({})で囲まれたprologプログラムを除く)に対しXlist用の変数を2つ付け加える。図3-3の規則(14)を本トランスレータで変換した例を図4-1(1)に示す。変換されてきたBUP節においてXで始まる変数がXlistである。2つのXlistのうち、最初(1番目)のものは各述語が呼ばれる時点でのスタック、2番目はゴールとしてたてたカテゴリの解析が終了した時点でのスタックを表わしており、2番目のXlistは、後続する述語の1番目のXlistとなる。

図4-1(2)は、スラッシュ(. /)のついたカテゴリを含む規則が変換された例である。

a(ARG) --> b(ARG1),c(ARG2),d(ARG3).

↓変換

b(G, (ARG1), I, X0, X) --> {link(a,G)},

goal(c, (ARG2), X0, X1),

goal(d, (ARG3), X1, X2),

a(G, (ARG), I, X2, X).

(1)

np --> a, noun, srel./np.

↓変換

a(G, O, I, X0, X) --> {link(np,G)},

goal(noun, O, X0, X1),

goal(slash, (srel,np), X1, X2),

np(G, O, I, X2, X).

(2)

図4-1 BUP-XGトランスレータによる変換例

4.3 goal節の拡張

次ページ図4-2にBUP-XGのgoal節の定義を示す。goal節において、第3及び第4引数がXlistである。従来のBUPシステムのgoal節は、(G1), (G5), (G8)だけで(但し、Xlistは無し)、それ以外はこの度追加したものである。

```

/* goal clause */
(G1) goal(G, A, X0, X, S0, S) :-
    ( wf_goal(G, _, X0, _, S0, _);
      fail_goal(G, _, S0), !,
        fail ), !,
      wf_goal(G, A, X0, X, S0, S).

(G2) goal(open, [X], X, [], S, S).

(G3) goal(close, [X], [], X, S, S).

(G4) goal(slash, [G, T], X0, X, S0, S) :-
    G =.. [Goal; Garg],
    T =.. [Trace; Targ],
    goal(Goal, Garg, x(Trace, Targ, X0), X, S0, S),
    deeper(X0, X).

(G5) goal(G, A, X0, X, S0, S) :-
    dictionary(C, A1, S0, S1),
    link(C, G),
    P =.. [C, G, A1, A, X0, X, S1, S],
    call(P),
    assertz(wf_goal(G, A, X0, X, S0, S)).

(G6) goal(G, GARG, x(G, GARG, X0), X0, S, S) :-
    assertz(wf_goal(G, GARG, x(G, GARG, X0), X0, S, S)).

(G7) goal(G, A, X0, X, S0, S) :-
    X0 = x(T, TARG, X1),
    T \= G,
    link(T, G),
    P =.. [T, G, TARG, A, X1, X, S0, S],
    call(P),
    assertz(wf_goal(G, A, X0, X, S0, S)).

(G8) goal(G, A, X0, X, S0, S) :-
    ( wf_goal(G, _, X0, _, S0, _);
      G = open;
      G = close;
      G = slash;
      assertz(fail_goal(G, _, S0)) ), !,
      fail.

```

図4-2 述語goalの定義

スタックにカテゴリをプッシュするのは (G4) である。実行時、ユニフィケーションの結果、Gにはゴールとしてたてたカテゴリ (Goal)、Tにはスタックにプッシュするカテゴリ (Trace) が、それぞれ引数付きの形でユニファイされる。そこでそれぞれをカテゴリと引数の形に分け、スタックにカテゴリをプッシュするとともに、現在のゴールとなっているカテゴリをリテラルgoalの第一引数として改めてgoalを実行する。そして、goalが成功した時点で、与えられたスタック (X0) と現時点でのスタック (X) の深さを比較しプッシュしたカテゴリが解析中に使用されたかどうかを述語deeperでチェックしている。

スタックのポップを行なうgoal節は (G6)、(G7) である。(G6) は、ゴールとしてたてられたカテゴリ (G) が、スタックのトップにある場合に適用される。前に述べたトップダウン法での述語virtualに対応する。(G7) は、スタックトップのカテゴリ (T) からゴールカテゴリ (G) へ到達可能である場合、スタックをポップし、そのカテゴリが、辞書引きによって見つかったカテゴリとして解釈し解析を進める為のものである。これは、前章で述べたように、BUPシステムではDCGによる文法規則の右辺の第1カテゴリが、ゴールカテゴリとして呼ばれることがないためである。このgoal節は、次の例文のような主語の欠けている関係節を含む文を解析するのに使われる。

I know a man that loves her.

上の文を図3-1の文法規則で解析すると、途中で“that loves her”をsrel./npとして解析する状況が生じる。srel./npをゴールとして実行すると、先に述べたようにnpがスタックにプッシュされ、改めてsrel (関係節) がゴールとして呼ばれる。次に辞書引きでrelpro (関係代名詞) の“that”が見つかり(12)が選択され、“loves her”をsをゴールとして解析することになる。ところがこれは主語が欠けたsであるため、通常の辞書引きで始まる解析では成功の可能性がない。そこでスタックのnpをポップし、このnpがボトムアップ解析の結果できあがったカテゴリとして考え(8)を選択することで、改めて“loves her”に対しvpがゴールとしてたてられ解析が成功する。

次に(G2), (G3)について説明する。図3-1の文法規則では、次のようなcomplex-NP制約[6]に違反した非文法文を受理してしまう。

I know a man that the girl that loves.

complex-NP制約とは、上の例文の“a man”のように、名詞句が二重に埋め込まれた文から左位置されることを禁止する制約条件である。図3-1の規則による解析では、“man”と“girl”のところで、スタックにプッシュされたnpがそれぞれ“loves”の目的語と主語として使用されるため、この制約条件が破られる。したがってこの制約条件を守るためには、srel./npを解析する前にスタックを一時的に空にし、srel./npの解析終了後にスタックを復元するメカニズムが必要である。BUP-XGでは、このようなスタックの退避及び復元のためのシステムカテゴリとしてopen, closeを用意した。文法記述の際にはopen, closeはそれぞれ“<”, “>”と略記し、トランスレータにより変換される。図4-3はcomplex-NP制約を満足するように図3-1(11)を修正したDCGによる規則及びその変換例である。

尚、PereiraのXGでもopen, closeというカテゴリを用いており、それらは、他の文法規則と同じようにユーザが定義するかたちになっているが、BUP-XGでは予めシステムが用意している。

```

np --> a, noun, < srel./np >.
      ↓ 変換
a(G, 0, I, X0, X) --> (link(a, G),
                        goal(noun, 0, X0, X1),
                        goal(open, (Y), X1, X2),
                        goal(slashb, (srel, np), X2, X3),
                        goal(close, (Y), X3, X4),
                        np(G, 0, I, X4, X)).

```

図4-3 open, closeのある規則及びその変換例

5. 実験結果

本章では、BUP-XG上で作成した英語の構文解析システムについて、従来のBUP上で作成されたシステムと比較しながら説明する。

5.1 文法の記述

これまでの例から、左位置変形を扱えるようにすることで、関係節に関する規則を減らすとともに、文法を見通し良く記述することができる。Wh疑問文も、これまで紹介して来たsyntax sugarを用いて次のように簡単に記述できる。

```

whq --> whnp, sq./np.

```

ここで、whnpは、疑問代名詞のカテゴリである。又、be動詞で始まるYes-no疑問文を、宣言文の中のbe動詞が文頭に移動したものと考えると、これも同様にして次のように記述できる。

```

sq --> bep, s./bep.

```

ここで、BUP-XG上で作成した文法とBUPでの文法の規則数を比較した結果を次ページ表5-1に示す。(注:各文法でカバーしている範囲が異なるので厳密な意味では正しい比較にはなっていない。)

表より分かるように、規則数の減少の要因となっているのは、ほとんどが疑問文に関するものであることがわかる。(注:不定詞句等に関する規則数の減少は左位置変形とは関係がない。)

| カテゴリ | BUP-XG | BUP | 差(BUP-BUPXC) |
|-----------|--------|-----|--------------|
| 文法総数 | 275 | 437 | 162 |
| 動詞句に関する規則 | 45 | 55 | 10 |
| 関係節 | 4 | 15 | 11 |
| Yes-no疑問文 | 4 | 92 | 88 |
| Wh疑問文 | 7 | 11 | 4 |
| 不定詞句等 | 30 | 53 | 23 |

表5-1 文法数の比較

5.2 構文解析例

図5-1(1)は、本システムによる構文解析結果の一例である。左外置されたカテゴリは、構文解析木の中で“t”で表わされている。(2)は、同じ文を従来のBUPシステムで解析した場合の解析結果の中で、全解析時間及び登録された成功、失敗ゴールの数だけを取り出したものである。図より明らかのように、BUP-XGでは、BUPの2倍以上の解析時間がかかっていることが分かる。

それに対し、以下のような左外置変形のない平叙文や疑問文を解析した場合の解析時間を表5-2に示す。

1. I saw many more books than she did.
2. I take my dictionary to her.
3. There are some men from the city.
4. Tell me when he came.
5. Did he give them to her?

| 例文 | 全解析時間 (msec) | |
|----|--------------|------|
| | BUP-XG | BUP |
| 1 | 5222 | 5158 |
| 2 | 1766 | 1935 |
| 3 | 1435 | 1355 |
| 4 | 1342 | 1456 |
| 5 | 1975 | 1927 |

表5-2 解析時間の比較

表5-2より左外置変形のない場合の解析時間は従来のBUPシステムでの場合とほぼ等しいことが分かる。以上のことから、本システムでの解析時間が左外置変形を伴った文を解析する場合に従来のシステムと比べて時間がかかる主な原因として次の2つが考えられる。

- 1) 本システムのBUP節は、Xlistの為に引数が全て2つずつ増えているためユニフィケーションに時間がかかる。
- 2) BUPシステムは、成功ゴール、失敗ゴールを登録して無駄な再計算を避けるメカニズムを持っているが、BUP-XGでは、それらを登録する際にXlistもパラメータとしている。そのため、左外置変形に関与しないカテゴリをゴールとして実行する際、その実行結果はXlistの状態には依存しないにも拘らず、Xlistの状態が異なると再計算をしてしまう。

1) による解析時間の増加にも拘らず、平叙文などでの解析時間がほぼ等しいのは、文法規則数の減少による解析時間の減少と1)による増加の分が相殺された為と考えられる。

2) については、成功ゴール、失敗ゴールを登録する際、左外置変形に関与しないカテゴリの場合には、Xlistの部分を変数の形で登録することで、上で指摘した再計算を避けることができる。簡単な実験を行なったところでは、図5-1の例文で解析木が出るまでの時間を8.4秒、全解析時間を9.3秒程度に短縮でき、このアルゴリズムの有効性を確認している。

```

The man that the girl that I like likes liked her.
12318 msec.
No. 1
|-sentence
|-sdec
|-sbas
|-subj
|-np
|-ddet
|-det -- the
|-nomhd
|-n -- man
|-ncomp
|-srel
|-relpro -- that
|-sbas
|-subj
|-np
|-ddet
|-det -- the
|-nomhd
|-n -- girl
|-ncomp
|-srel
|-relpro -- that
|-sbas
|-subj
|-np
|-pron -- I
|-vp
|-v -- like
|-obj
|-np -- t
|-vp
|-v
|-v -- like
|-suffix -- s
|-obj
|-np -- t
|-vp
|-v
|-v -- like
|-suffix -- ed
|-obj
|-np
|-pron -- her

```

Total Time = 13464 msec.

number of wf_goal was : 29.
number of wf_dict was : 12.
number of fail_goal was : 215.

(1) BUP-XGによる解析例

Total Time = 5257 msec.

number of wf_goal was : 19.
number of wf_dict was : 12.
number of wf_fail_goal was : 129.

(2) BUPによる解析結果 (statisticsのみ)

図5-1 構文解析例

6. 問題点

6.1 文法記述上の制限

本システムで文法を記述する際、文法規則の右辺の第1カテゴリに slash のついたカテゴリや、open, close で囲まれたカテゴリを記述することができない。これは、BUPシステムにおける第1カテゴリが常に辞書引き又は、解析の結果できるカテゴリであるため、それらのカテゴリがゴールとして呼ばれることがないためである。どうしても必要な場合には、第1カテゴリをダミーのカテゴリとし、辞書引きの際そのダミーのカテゴリが常に見つかる形で実現できるが、その部分が完全にトップダウンになるため、左再帰規則には使えない。

6.2 等位構造に関する問題

下のように、関係節の中の埋め込み文が and でつながっている場合、現在のシステムでは、スタックにプッシュされたカテゴリ np が "love" の目的語として一度ポップされるとスタックが空となるため、"dislike" の目的語が見つからず解析は失敗する。

She is a girl that I love and you dislike.

したがって、文法規則 $s \rightarrow s, \text{and}, s$ において、 s の部分木が見つかり、右辺の2つめの s をゴールとして呼ぶ時になんらかの方法でスタックを復元するメカニズムが必要である。

7. おわりに

本研究では、英語の関係節や疑問文における語の移動を説明する左外置変形に注目し、これをボトムアップ構文解析システムBUP上で扱えるようにBUPを拡張した。文法の記述は拡張したDCG(新機能の為に加えたsyntax sugarを含む)で行なう。又、実際に英語の文法を記述して、その有効性を確認した。しかし、その反面、解析時間がかかることや、規則の右辺の第1カテゴリに関して制約があること、関係節内での文の等位構造に関する問題などが残っている。又、本システムを改良することで右外置変形(主語の関係節が文末に移動するもの)を扱えるかどうか検討していきたい。

謝辞

本研究を行なうにあたり、計算機を使用させてくださいました、瀬 一博ICOT研究所長に深く感

謝致します。又、日頃、御討論いただいた東京工業大学田中研究室の雷氏に感謝致します。

参考文献

- [1] Y. Matsumoto and et al. "BUP: A bottom-Up Parser Embedded in Prolog", New Generation Computing, 1,2,1983.
- [2] 田中穂積、他、"ボトムアップ構文解析システムBUP上での英語文法開発とBUPの評価", Proceedings of THE LOGIC PROGRAMMING CONFERENCE '84, 12-2, 1984.
- [3] Pereira, F. "Extrapolation Grammar" AJCL Vol. 7, No. 4, 1981.
- [4] Winograd, T. "Language as a Cognitive Process, Vol 1: Syntax", Addison-Wesley, 1983.
- [5] 松本裕治、他、"BUPトランスレータ", 電総研報, Vol47, 第8号, 1983.
- [6] 大塚高信、他、"新英語学辞典", 研究社
- [7] Pereira, F and Warren, D. "Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks", Artificial Intelligence, 13, pp231-278, May 1980.