

3T-8

BUP系解析システム上での
トップダウンな情報の制御について

奥村学, 岩山真, 田中穂積
(東京工業大学 工学部)

1 はじめに

近年、Prologを用いた自然言語解析に関する研究が多く見られるようになってきた。Prologを自然言語解析に用いることの1つの利点は、DCG(Definite Clause Grammar)[4]とよばれる文法記述形式で記述した文法を、Prologプログラムに変換し、そのままパーザとして動作させることが可能な点にある。我々は、作成した文法中に左再帰的な文法規則が含まれている場合のことを考慮して、DCGで記述した文法を変換して、ボトムアップにパーズングを行なうPrologプログラムを生成するシステムBUP[3]を用いている。このBUPは、現在は自然言語解析システムLangLAB[8]の中核になっている。

純粋にボトムアップなパーズングを行なうシステムでは、解析木を下から上に向けて生長させるにつれて、情報を下から上へと自然に送ることができる一方、上から下へは情報を送ることができない。そのため、解析途中での構文的、意味的チェックも、情報が下から上がってきてはじめて実行されることになり、上から下に情報を流すことが可能な場合に較べると、その実行のタイミングが遅くなり、不要な計算をすることがある。

既存のボトムアップ解析システムの中には、下から上への情報を用いて一度パーズングを終了し、その後、得られた解析木上をたどる過程で上から下に情報を流すものは存在する[1][6]。しかし、本稿で述べるように、解析途中にダイナミックに情報を双方向に流せるものは存在しなかった。

BUPは、ボトムアップ解析システムではあるが、詳細に検討すると、純粋なボトムアップ法ではなく、後述するgoal節をその解析に用いていることから、トップダウンな予測を利用することができ、従って、上から下へ情報を流すことも可能である。左外置処理のためBUPを拡張したBUP-XG[2]では、この特徴を生かし、左外置処理に必要なスタックを実現している。

本研究では、BUP-XGにおけるスタックがトップダウンな情報として用いられていることに着目し、同様の手法を用いて、BUP系の解析システムでは困難であると思われていた、上から下へ情報を流す機構BUP-UTI(Using Top-down Information BUP)を開発した。このBUP-UTIにより、ボトムアップ解析システム上で、解析途中に上下双方向へダイナミックに情報を流すことが可能になることを示す。

本稿では、まず2章で、BUP-UTIにおけるトップダウンな情報の流れを実現する基礎となった、BUP-XGにおけるスタックの実現法について述べるとともに、BUP-UTIの基本原則を説明する。3章では、トップダウンな情報の流れを生かした解析の例を示し、BUP-UTIが従来のボトムアップ解析システムに比べ、早期に非文法的箇所を検出できるなどの有効性をもつことを示す。

2 BUP-XGの概要とBUP-UTIの基本原則

2.1 BUP-XGにおけるスタックの実現法

左外置処理をトップダウンに実行するのに、XG(extra position grammar)[5]を用いる方法が考えられている。この方法では、左外置処理実現のため、スタックを用いる。このスタックを実現するため、BUP-XGでは、後述するようにBUPの基本動作を利用している。

BUP-XGでは、スタックの状態を構造体で表わし、それを後続する述語に次々受け渡していくことで、スタックを実現している。例えば、下の文法規則(1)に対して、(2)のように2変数を付加したものを考える。(2)の各文法カテゴリに付加された2変数(下線部)のうち、左の変数は、その文法カテゴリの解析を始める時点でのスタックを表わし、右の変数は、解析が終了した時点でのスタックを表わす。

$$s \rightarrow np, vp. \quad (1)$$

$$s(\underline{X0}, \underline{X2}) \rightarrow np(\underline{X0}, \underline{X1}), vp(\underline{X1}, \underline{X2}). \quad (2)$$

これを用いて解析を行なうと(実際の解析は(3)のBUP節で行なわれるから)、

$$\begin{aligned} np(G, [], 1, \underline{X0}, \underline{X1}, XR) &\rightarrow \{s(G)\}, \\ goal_x(vp, [], \underline{X1}, \underline{X2}), \\ s(G, [], 1, \underline{X0}, \underline{X2}, XR). \end{aligned} \quad (3)$$

npの解析が成功すると、X0, X1にスタックが返される。npの出力スタックX1が、vpの入力スタックに渡され、vpの解析の結果、X2に新しいスタックが返される。npの入力スタックX0, vpの出力スタックX2をそれぞれsの入出力スタックとして解析が続けられる。

ここで注意したいのは、npの出力スタックX1をvpの入力スタックに伝える点である。このvpに対する入力スタックはトップダウンな情報であり、npからgoal節のボディで辞書引きされる文法カテゴリまで情報を伝えることができる(図1参照)。このスタックの流れを実現するメカニズムが、BUP-UTIの基本原則となっている。次節では、この基本原則について述べる。

2.2 BUP-UTI (BUP-XG)の基本原則

前節では、BUP-XGにおけるスタックの実現法について述べた。BUP-XGでは、変数を介して次々とスタックの状態を受け渡し、スタックを実現している。その際、各文法カテゴリに対して入力、出力スタックが存在し、ある文法カテゴリの出力スタックは、文法規則中のその文法カテゴリのすぐ右にある文法カテゴリの入力スタックになっている。そして、この入力スタックは、goal節を用いて解析されることにより、トップダウンな情報として、次に辞書引きされる文法カテゴリまで伝えられる。

以上のようなスタックの流れに関する考察から、スタックの実現に用いられた2変数をトップダウンに情報を流すための変数(以後、TI用変数と呼ぶ)として利用し、この変数で情報を左から右に(上から下に)流すことで、より多様な解析システム実現を目指したのがBUP-UTIである。

BUP-UTIでは、上に示した(1) → (3)のように、

ユーザが記述した文法は、BUP節等に変換される。そして、それらが図1のgoal節とともにボトムアップ構文解析システムとして直接動作する。

解析は、次のゴールを実行することから始まる。

?-goal(s,A,[she,smiled],[]).

このゴールの実行により、(4)のボディが実行される。(4)のボディでは、入力文の先頭の単語の辞書引きを行ない(dict述語)、その単語の文法カテゴリをヘッドとするBUP節を選択する。例の場合、先頭の単語"she"の辞書引きの結果、その文法カテゴリであるnpをヘッドとするBUP節(3)が選択される。そして、そのボディの実行において、文法カテゴリvpを次のゴールとして解析を行なう(goal_x(vp,[],X1,X2))。これは、入力文の残りの部分に対して、文法カテゴリvpの部分が存在するはずだという予測をし、その予測(文法カテゴリ名vp)をトップダウンに流していることに相当する。ここで、さらにgoal節のボディで次の単語を辞書引きし、その文法カテゴリをヘッドとするBUP節を選択することで解析が進むが、このようにBUPは、純粹にボトムアップに解析を行なっているわけではなく、レフトコーナーから解析を始め、そこから得た情報を用いて、次に来るべき文法カテゴリについて予測をし、その予測を(トップダウンに)用いることによって解析を進めているわけである。

goal節を用いることによって得られる、このBUPの動作特性を生かし、従来のgoal節を修正し、TI用変数を扱えるようにする(図1参照)ことで、情報をトップダウンに流すことが可能になる。2.1で述べたように、ある文法カテゴリを解析して得られた情報が、次に辞書引きされる文法カテゴリに(上から下へ)伝えられるのは、図1のgoal_x節(5)において、前の文法カテゴリから伝わった入力TI用変数X1が、次の単語を辞書引きした結果として呼び出される述語Cの入力TI用変数に受け渡されることによる(下線部参照)。

```
goal(G,A,X,Z) :-
  dict(C,A1,X,Y),
  Link =.. [C,G],call(Link),
  P =.. [C,G,A1,AA,[ ],[ ],[ ],Y,Z],call(P),
  A = AA. (4)
```

```
goal_x(G,A,X1,X0,X,Z) :-
  dict(C,A1,X,Y),
  Link =.. [C,G],call(Link),
  P =.. [C,G,A1,AA,X1,X1,X00,Y,Z],call(P),
  A = AA,X0 = X00. (5)
```

図1 BUP-UTIにおけるgoal節

3 BUP-UTIを用いた構文的、意味的チェック

本章では、従来のBUPおよびBUP-UTIにおける構文的チェックの実行法を比較することで、その実行のタイミングの違いを明らかにし、BUP-UTIの有効性を示す。例としては、主語と動詞の呼応のチェックを用いる。

(i) 従来のBUPの場合

従来のBUPでは、下に示すような文法規則を用いてチェックが行なわれる。

```
s(S_A) --> np(NP_A), vp(VP_A),
  {concord(NP_A,VP_A)}, (6)
```

```
vp(V_A) --> v(V_A), np(NP_A). (7)
```

具体的には、v(動詞)の各単語の辞書に記述されている人称・数に関する情報が、文法規則(7)を用いて述語vpの引数まで伝わり、その結果(6)で、npの人称・数に関する情報との間で呼応のチェックが行なわれる(concord)。これは、呼応のチェックが、(6)のvpの解析終了まで遅延されることを意味する。

(ii) BUP-UTIの場合

BUP-UTIを用いると、次に示すような文法規則でチェックを行なうことができる。

```
s(S_A)[X0,X1] ==> np(NP_A),
  vp(VP_A)[[NP_A1X0],X1]. (8)
vp(V_A)[[NP_A1X0],X0] ==> v(V_A),
  {concord(NP_A,V_A)},
  np(NP1_A). (9)
```

BUP-UTIでは、(8)の文法規則で、npの人称・数の情報をTI用変数に追加してvpに渡し、それを用いて(9)の規則中で、vの人称・数の情報との間の呼応のチェックを直ちに行なうことができる(concord)。

この例からわかるように、従来のBUPでは、vpの解析が終わるまで呼応のチェックが遅延されるのに対し、BUP-UTIの場合には、vpの解析中、vが現われた直後に呼応のチェックが行なえる。この違いは、vpにあたる部分が単純な場合には、さほど問題にならないが、vpが複雑で大きな構造をもつ文を解析する場合(動詞が目的語として不定詞句、that節など、文に近いような構造のものをとる場合)には、この呼応のチェックの実行遅延による計算時間の損失は大きくなる。

以上述べてきたように、BUP-UTIを用いれば、構文的、意味的チェックを、従来のBUPに較べ早期に実行できることから、不要な計算を早期に回避することが可能である。

4 おわりに

以上、BUP-UTIの基本原則、そのインプリメントなどについて述べた。

BUP-UTIを用いることにより、ボトムアップ解析システム上で、解析途中に上下双方向へダイナミックに情報を流すことが可能になった。また、それにより、従来のボトムアップ解析システムに較べ、早期に非文法的箇所を検出できるなどの有効性が得られた。

今後の課題としては、3章の例からわかるとおり、BUP-UTIにおける文法記述は書きづらい。従って、高水準な文法記述言語[6]を設計する必要がある。この問題の解決の方向が、[7]に部分的ながら示されている。

参考文献

- [1] 電総研(編): 拡張LINGOL, 1978.
- [2] 今野聡, 田中穂積: 左外置を考慮したボトムアップ構文解析システム, コンピュータソフトウェア, Vol.3, No.2, 1986, pp.19-29.
- [3] Matsumoto, Y., Tanaka, H., Hirakawa, H., Miyoshi, H. and Yasukawa, H.: BUP: A Bottom-Up Parser Embedded in Prolog, New Generation Computing, 1,2, 1983.
- [4] Pereira, F. and Warren, D.: Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artif. Intell., Vol.13, 1980.
- [5] Pereira, F.: Extraposition Grammar, AJCL, Vol.7, No.4, 1981.
- [6] 田村直良, 高倉伸, 片山卓也: 自然言語処理を目的とした属性文法評価システム, コンピュータソフトウェア, Vol.3, No.3, 1986, pp.266-279.
- [7] 田村直良, 田中穂積: 並列名詞句の構造解析について, 情報処理学会NLI研, 59-2, 1987.
- [8] 田中穂積, 上脇正, 奥村学, 沼崎浩明: 自然言語処理のためのソフトウェアシステムLANG LAB, Proc. of the LPC'86, 3.1, 1986.