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

田中 穂積，高倉 伸，今野 聡
（東京工業大学 工学部）

1. はじめに

prologによる英語の構文解析に関する研究は、各地で進められてきたが、prologで記述された大きな英語の文法体系は、まだ開発されていない。そこで、本研究では、文法規則数約400の比較的大きな英語の文法体系を作成し、ＥＴＬとＩＣＯＴで開発されたＢＵＰシステム［松本 他 83a］上で構文解析を行なった。その結果、１単語当り100～300 msecで構文解析結果が得られ、ＢＵＰシステムのツールとしての有効性を確認した。特に、本研究では、文法規則の増加に伴う構文レベルの曖昧さを、意味的な情報をなるべく用いずに除去する方策に重点を置き、文法規則を開発した。

一般に、自然言語の文法規則を文脈自由文法で表現しようとした時、左再帰規則が現れる。（我々の開発した文法体系では、現在21個の左再帰規則がある。）このため、ボトムアップに構文解析を進めるＢＵＰシステムを採用することにした。

次章では開発した英文法の概要を述べる。3章では補強項について述べ、4章では、これらを用いた構文解析例をもとに考察を加える。

2. 英語の文法体系の作成

英文法を作成するに当り、次のことを検討した。

(i) 解析させる英語の範囲

書きことばに限定し、倒置法、挿入などの一部の特殊構文を除いた英文を対象とする。

(ii) 文法の記述形式

文法規則の増加に伴う構文レベルの曖昧さを除去するために、文脈自由文法の文法規則に、その適用可能条件をprologプログラムで記述し、補強項として、埋込んだ。

以下に我々の作成した英語文法の詳細を述べる。

2.1. 文の種類

一般に、文は命令法と直接法とに分けられ、命令法の中に命令文があり、直接法の中には平叙文と疑問文とがある。更に、疑問文は、文全体の正否を尋ねているYes-no型疑問文と文の一部を疑問詞で尋ねているWh-型疑問文との2つがある。

これらは、それぞれ文頭の構文に特徴がある。命令文は主語が省略され、主にdo, be、一般動詞の原形で始まる。平叙文は、主語となりうる名詞句から始まる。Yes-no型疑問文は助動詞、もしくは、be動詞の直後に名詞句が続く形で始まる。Wh-型疑問文は、疑問詞、又は前置詞＋疑問詞で始まる。

例を示すと、

命令文 : Do it quickly.
Put the apples in the basket.

平叙文 : He put the apples in the basket.
There are several of them.

Yes-no型疑問文 : Are there more than two?
Did he put them in the basket?

Wh-型疑問文 : What is in the basket?
To whom did he give them?

```
sentence(SDEC_A,SENT_S,[sentence,SDEC_T])
 --> sdec(SDEC_A,SDEC_S,SDEC_T).

sentence(SIMP_A,SENT_S,[sentence,SIMP_T])
 --> simp(SIMP_A,SIMP_S,SIMP_T).

sentence(SQ_A,SENT_S,[sentence,SQ_T]) -->
 sq(SQ_A,SQ_S,SQ_T).

sentence(SWHQ_A,SENT_S,[sentence,SWHQ_T])
 --> swhq(SWHQ_A,SWHQ_S,SWHQ_T).
```

図1

以上のことから、我々は文を図1のような文法規則で、次の4つのカテゴリーに分けて文法規則を記述した。

sdec：平叙文(Declarative Sentences)
simp：命令文(Imperative Sentences)
sq　：Yes-no型疑問文(Propositional Interrogatives)
swhq：Wh-型疑問文(Argument Interrogatives)

2.2. 平叙文

単に平叙文といっても英語には特殊構文がいくつかあり、そのすべてに対応できる文法体系を作成することは困難である。我々の作成した文法では、基本五文型の他に、受動態、there-is構文、分詞構文、従属節、重文が扱える。

(a) 基本五文型

基本五文型を表す文法規則の記述例を示すと次のようになる。*1

```
sdec(VP_A,SDEC_S,[sdec,SUBJ_T,ADV_T,VP_T])
--> subj(SUBJ_A,SUBJ_S,SUBJ_T),
  ( adv(ADV_A,ADV_S,ADV_T) ; [] ),
    vp(VP_A,VP_S,VP_T),
  ( subj_v(SUBJ_A,VP_A) ).
```

図2

ここで文法規則の第1引数は補強項で用いられる情報が貯えられる部分で以下補強部と呼ぶ。第2引数は空のスロットで、本研究では使用していない。第3引数は構文木の生成に使用している。{}で囲まれている部分が補強項になり、subj_vという2引数の述語がprologで定義されている。この述語は、主語と動詞の呼応の現象を確認するもので、主語が三人称単数の場合、動詞は過去形又は原形+sでなければならないので、これを確認している。

否定文、各時制を表すために、進行形を作るbe、完了形を作るhave、叙述の助動詞、否定に用いるdoを合せたauxdという助動詞句のカテゴリーを作り、vpの前に置いた。

```
sdec(VP_A,SDEC_S,[sdec,SUBJ_T,ADV_T,
      AUXD_T,VP_T]) -->
    subj(SUBJ_A,SUBJ_S,SUBJ_T),
  ( adv(ADV_A,ADV_S,ADV_T) ; [] ),
    auxd(AUXD_A,AUXD_S,AUXD_T),
    vp(VP_A,VP_S,VP_T),
  ( subj_auxd(SUBJ_A,AUXD_A),
    auxd_v(AUXD_A,VP_A) ).
```

図3

この規則中の補強項も呼応の現象を確認するものである。

今まで述べた基本五文型の文法規則は一般動詞に対するものだが、be動詞の場合、呼応の現象が一般動詞と異なり、否定にする時、doを用いずbe動詞のあとにnotを付けるので一般動詞とは別に記述した。(Appendix参照。)

(b) there-is構文

there-is構文を記述する文法規則を1つ示すと次のようになる。

```
sdec(SDEC_A,SDEC_S,[sdec,THERE_T,AUX_T,
     BEP_T,SUBJ_T]) -->
    there(THERE_A,THERE_S,THERE_T),
    aux(AUX_A,AUX_S,AUX_T),
    bep(BEP_A,BEP_S,BEP_T),
    subj(SUBJ_A,SUBJ_S,SUBJ_T),
  ( subj_auxd(SUBJ_A,AUX_A),
    aux_bep(AUX_A,BEP_A) ).
```

図4

この構文におけるthereの扱い方には、2通りの考え方がある。1つはthereを形式主語と考え、基本五文型の一種とする考え方である。しかし、呼応の現象が形式主語としたthereではなく、be動詞の後の名詞句との間で起こる。このことから、この考え方を用いる事には無理がある。もう1つの考え方は、副詞のthereが文頭にきたために、主語と動詞が倒置を起こしたとする考え方である。この場合、呼応の現象もうまく説明でき、自然であるので、この考え方を採用した。しかし、すべての副詞句に関してこの倒置が起きるわけではないこと、そしてthere-is構文中のthereには副詞的な意味がほとんどなくなっていることから、thereという新たなカテゴリーを用いて記述した。

2.3. 命令文

命令文は、平叙文の主語が欠け、時制もないので、その文法規則は簡単である。ここでは、そのうちの1つだけを示しておく。(他はAppendix参照)

```
simp(VP_A,SIMP_S,[simp,DOP_T,VP_T]) -->
    dop(DOP_A,DOP_S,DOP_T),
    vp(VP_A,VP_S,VP_T),
  ( simp_vp(DOP_A),dop_vp(VP_A) ).
```

図5

この規則中の補強項、simp-vpは命令文の文頭にくる助動詞do、一般動詞、be動詞が原形であることを確認するもので、dop-vpは助動詞doの後の動詞が原形であることを確認する述語である。

脚注*1 文法開発者は、DCG[Pereira 80]で普通に文法規則を記述し、BUPトランスレータ[松本 他83b]を通してボトムアップに構文解析するprologプログラムに変換する。

2.4. Yes-no型疑問文

疑問文の文法規則を記述する時、平叙文の時のように、助動詞句を一括して一箇所に置くことができないため、規則数がどうしても多くなる。たとえば、"He has been doing it."と"He is doing it."という2つの平叙文はともに図3に示す規則で扱える。しかし、それぞれの疑問文"Has he been doing it?"と"Is he doing it?"とは次のような2つの規則を必要とし、1つにまとめることができない。

```
sq(SQ_A,SQ_S,[sq,HAVEP_T,SUBJ_T,ADV_T,
    BEP_T,ADV1_T,VP_T]) -->
     havep(HAVEP_A,HAVEP_S,HAVEP_T),
     subj(SUBJ_A,SUBJ_S,SUBJ_T),
   ( adv(ADV_A,ADV_S,ADV_T) ; [] ),
     bep(BEP_A,BEP_S,BEP_T),
   ( adv(ADV1_A,ADV1_S,ADV1_T) ; [] ),
     vp(VP_A,VP_S,VP_T),
   ( subj_havep(SUBJ_A,HAVEP_A),
     havep_bep(BEP_A),be_ving(VP_A) ).

sq(SQ_A,SQ_S,[sq,BEP_T,SUBJ_T,ADV_T,
    VP_T]) -->
     bep(BEP_A,BEP_S,BEP_T),
     subj(SUBJ_A,SUBJ_S,SUBJ_T),
   ( adv(ADV_A,ADV_S,ADV_T) ; [] ),
     vp(VP_A,VP_S,VP_T),
   ( subj_bep(SUBJ_A,BEP_A),
     be_ving(VP_A) ).
```

図6

ここで用いられている補強項は、subj_havepとsubj_bepがともに主語と助動詞a形との呼応の現象を確認するもので、havep_bepは、haveの後のbe動詞が過去分詞"been"であることを確認し、be_vingはbe動詞+ingで進行形を作っていることを確認するものである。

2.5 Wh-型疑問文

疑問詞を用いた疑問文の文法規則は次の4つの規則に代表される。

```
swhq(SWHQ_A,SWHQ_S,[swhq,WHPP_T,SQ_T])
--> whpp(WHPP_A,WHPP_S,WHPP_T),
    sq(SQ_A,SQ_S,SQ_T).

swhq(SWHQ_A,SWHQ_S,[swhq,WHNP_T,SQ1_T])
--> whnp(WHNP_A,WHNP_S,WHNP_T),
    sq1(SQ1_A,SQ1_S,SQ1_T).

swhq(SWHQ_A,SWHQ_S,[swhq,WHNP_T,SQ2_T])
--> whnp(WHNP_A,WHNP_S,WHNP_T),
    sq2(SQ2_A,SQ2_S,SQ2_T).

swhq(SWHQ_A,SWHQ_S,[swhq,WHNP_T,VP_T])
--> whnp(WHNP_A,WHNP_S,WHNP_T),
    vp(VP_A,VP_S,VP_T),
  ( whnp_vp(WHNP_A,VP_A) ).
```

図7

whppは疑問副詞又は前置詞+疑問詞で、その後は完全なYes-no型疑問文がくる。whnpは疑問代名詞であるか、この後には、sq1, sq2, vpと3通りある。sq1とは、Yes-no型疑問文における動詞の目的語あるいは補語が欠けたもので、疑問詞がそれを指している疑問文に対して適用される。sq2はthere-is構文のYes-no型疑問文の主語にあたる名詞句が欠けたものである。後にvpがくる場合は疑問代名詞が主語として働く場合である。それぞれの例を示す。

whpp[When], sg[did you go there]?
whnp[What], sg1[do you have]?
whnp[Who], sg2[is there]?
whnp[Who], vp[went to school]?

sq1, sq2は、sqの1部が欠けているため、単独では非文法文となる。しかし、その文法規則はsqのものとほとんど変わらない。そのため、欠けている主語や目的語等の位置にtraceを置くことで、その欠けた品詞を補なって解析を進めていくという考え方、Extrapositional Grammar [Pereira 81]がある。現在、この考えの実現には、すべての品詞の切れ目で何かが欠けていると仮定して解析を進めるなど、効率面で問題があると考えられている。しかし、先読みをするなど、効率の改善を図り、この考え方を導入する方向で検討すべきであろう。

2.6. 動詞句

動詞句の文法規則(Appendix参照)は、基本五文型におけるV, VC, VO, VOO, VOCという単純なものではなく、ホーンビーの動詞型[Hornby]の分類に基づいて細分化している。

動詞句を記述する文法規則に左再帰規則が多く現れる。これは、前置詞句、副詞などの修飾語句がうしろからかかることが多いためである。

```
vp([advp|VP1_A],VP_S,[vp,VP1_T,COM_T,
    PP_T]) -->
     vp(VP1_A,VP1_S,VP1_T),
   ( comma(COM_A,COM_S,COM_T) ; [] ),
     pp(PP_A,PP_S,PP_T),
   ( vp_pp(VP1_A) ).

vp([advp|VP1_A],VP_S,[vp,VP1_T,ADVP_T])
--> vp(VP1_A,VP1_S,VP1_T),
    advp(ADVP_A,ADVP_S,ADVP_T),
  ( vp_advp(VP1_A) ).
```

図8

この左再帰規則を用いた解析例を以下に示す。

```
He went there with her at five yesterday.

1565  msec.
No. 1
 |-sentence
    |-sdec
       |-subj
       |   |-np
       |       |-pron -- he
       |-vp
          |-vp
          |   |-vp
          |   |   |-vp
          |   |   |   |-v -- went
          |   |   |   |-advp
          |   |   |       |-adv -- there
          |   |   |-pp
          |   |       |-p -- with
          |   |       |-obj
          |   |           |-np
          |   |               |-pron -- her
          |   |-pp
          |       |-p -- at
          |       |-obj
          |           |-np
          |               |-detq
          |               |   |-number -- five
          |-advp
              |-adv -- yesterday

Total Time = 2488  msec.

number of wf_goal was :       13.
number of wf_dict was :       10.
number of wf_fail_goal was : 73.
```

図9

3. 補強項

構文上の制約条件を加えるために文法規則中に埋込まれたprologプログラムを補強項と呼ぶことにする。我々の作成した英文法では約100種類の補強項が用いられている。このうち10個は単語を限定するもので、補強項とは別の意味で用いられている。7個が主語と動詞の呼応の現象を確認するもので、20個は動詞の文型パターンを確認するものである。構文レベルの曖昧さの除去において、この文型パターンの確認は非常に効果がある。又、副詞句、前置詞句、不定詞句などの副詞的な修飾語句のかかり方を制限するものが10個以上あり、これらが曖昧さの除去に大きく関係がある。他のものは語形の確認（haveの後は過去分詞でなければならない等）のためのものが多く、非文法文の除去には役立つが、曖昧さの除去とはあまりならないものである。

4. 実験結果とその検討

我々の作成した文法において、補強項がどの程度影響しているか、又、BUPシステムが、文法規則数約400のものをどの程度の速度で稼動するかを次のような実験を行なって調べた。

<実験1>

"This is hard for me to do."を例にとり、補強項を全く用いない状態から補強項を順次加えていき、その効果を調べてみる。

<実験2>

補強項が全くない場合とある場合とで、構文解析木の数と解析時間とを比較検討する。

4.1. 実験1の結果と考察。

"This is hard for me to do"を補強項がない状態で構文解析を行なうと図10に示す5つの構文木が表れる。そして、補強項を加えるとNo.5の構文木のみが残る。この5つの構文木を見た時、No.1, No.2, No.4はいずれも"me"という代名詞を不定詞が修飾しており不自然である。そこで、np → pron, ncomp.という規則に対してncompによる修飾を受けられる代名詞は"it", "something", "everything", "nothing"のいずれかの場合だけであると制限することで、No.3とNo.5の2つだけに減らすことができる。更に、No.3とNo.5を比較した時、意味的には、ほとんど差がないと考えられ、No.3を「predが形容詞句ならば不定詞がかからない」という条件を加えることで除去することができる。この結果、5つあった構文解析結果を1つにすることができた。構文解析時間を見ると、補強項を用いない場合、Totalで4179msec.かかっていたが、補強項を用いると、Totalで3190msecと約1秒短縮されるという結果が得られた。

つまり、補強項の導入は、構文レベルの曖昧さを除去するだけではなく、解析の高速化にもつながると考えられる。

```
This is hard for me to do.

2173  msec.
No. 1
 ¦-sentence
    ¦-sdec
       ¦-subj
       ¦  ¦-np
       ¦     ¦-ddet
       ¦     ¦  ¦-det -- this
       ¦-bep
       ¦  ¦-be -- is
       ¦-pred
          ¦-pred
          ¦  ¦-adjp
          ¦     ¦-adj -- hard
          ¦-pp
             ¦-p -- for
             ¦-obj
                ¦-np
                   ¦-pron -- me
                   ¦-ncomp
                      ¦-infinitrel
                         ¦-infinitive
                            ¦-to -- to
                            ¦-vp
                               ¦-v -- do

369  msec.
No. 2
 ¦-sentence
    ¦-sdec
       ¦-subj
       ¦  ¦-np
       ¦     ¦-ddet
       ¦     ¦  ¦-det -- this
       ¦-bep
       ¦  ¦-be -- is
       ¦-pred
          ¦-adjp
             ¦-adj -- hard
             ¦-adjcomp
                ¦-p -- for
                ¦-np
                   ¦-pron -- me
                   ¦-ncomp
                      ¦-infinitrel
                         ¦-infinitive
                            ¦-to -- to
                            ¦-vp
                               ¦-v -- do

304  msec.
No. 3
 ¦-sentence
    ¦-sdec
       ¦-subj
       ¦  ¦-np
       ¦     ¦-ddet
       ¦     ¦  ¦-det -- this
       ¦-bep
       ¦  ¦-be -- is
       ¦-pred
          ¦-pred
          ¦  ¦-adjp
          ¦     ¦-adj -- hard
          ¦     ¦-adjcomp
          ¦        ¦-p -- for
          ¦        ¦-np
          ¦           ¦-pron -- me
          ¦-infinitive
             ¦-to -- to
             ¦-vp
                ¦-v -- do

150  msec.
No. 4
 ¦-sentence
    ¦-sdec
       ¦-subj
       ¦  ¦-np
       ¦     ¦-ddet
       ¦     ¦  ¦-det -- this
       ¦-bep
       ¦  ¦-be -- is
       ¦-pred
          ¦-adjp
             ¦-adj -- hard
             ¦-adjcomp
                ¦-p -- for
                ¦-np
                   ¦-pron -- me
                   ¦-ncomp
                      ¦-infinitrel
                         ¦-infinitive
                            ¦-to -- to
                            ¦-vp
                               ¦-v -- do

124  msec.
No. 5
 ¦-sentence
    ¦-sdec
       ¦-subj
       ¦  ¦-np
       ¦     ¦-ddet
       ¦     ¦  ¦-det -- this
       ¦-bep
       ¦  ¦-be -- is
       ¦-pred
          ¦-pred
             ¦-adjp
                ¦-adj -- hard
                ¦-adjcomp
                   ¦-p -- for
                   ¦-np
                   ¦  ¦-pron -- me
                   ¦-infinitive
                      ¦-to -- to
                      ¦-vp
                         ¦-v -- do

Total Time = 4179  msec.

number of uf_goal was :      37.
number of uf_dict was :      9.
number of uf_fail_goal was : 80.
```

図10

4.2. 実験2の結果と考察

次の例文を用いてこの実験を行なった。

1. I saw many more books than she did.
2. I take my dictionary to her.
3. This is a film developed in the research.
4. There are some men from the city.
5. Be careful not to break the vase when you put it down.
6. Tell me when he came.
7. Did he give them to her?
8. Could she have been given many of them?
9. Which ones did he give her?

10. Who was made the boy?

補強項を用いた場合と用いない場合とで、構文解析木の数、解析時間の違いを表1にまとめた。

例文	補強項なし		補強項あり		単語数
	構文木数	解析時間(msec)	構文木数	解析時間(msec)	
1	18	* (178084)	1	5158 (16851)	8
2	6	7003 (12370)	1	1935 (3415)	6
3	10	6644 (6290)	1	2485 (3441)	8
4	3	1834 (1889)	2	1355 (1566)	7
5	396	* (256554)	2	4279 (20854)	12
6	6	2396 (3854)	2	1456 (2107)	5
7	22	* (87967)	1	1927 (2651)	6
8	7	4380 (29266)	1	2453 (4213)	8
9	24	* (194189)	1	2457 (3559)	6
10	2	4483 (12333)	1	2321 (8716)	5

表 1

補強項を用いない場合、BUPの高速化[松本他83c]を図るためにassertしているwf_goalが大きくなりすぎ、workspaceがなくなってしまう。そこで、解析木の数を把握するためにwf_goalのassertを制限することで最後まで解析させた。表1における解析時間は、高速化されたBUPによる構文解析のTotal Timeを示したもので、*はworkspaceがなくなり、途中で解析が終ってしまったものである。又、( )内の数字はwf_goalのassertを制限した時の時間で、平均すると2倍近く遅くなっている。

補強項を用いなかった時、workspaceがなくなってしまった例文に対して、補強項を用いた場合、解析結果を得ることができる。これは、補強項によって誤った解析を早めに打ち切ることができ、wf_goalのassertが少なくなるためである。得られる構文木の数を比較すると、補強項がない場合、単語数が7個前後の文に対して約10の曖昧さがあり、そのほとんどが、構文レベルの曖昧さである。そのため、補強項を用いることで、1つないし2つの構文木にしぼることが可能であった。補強項を用いて2つの構文木が得られたNo.4, No.5, No.6の文について見てみると、No.4の場合、there-is構文の主語の後の前置詞句の処理によって異なった構文木を得る。「何人かの男達が町から来ている。」と「町からきた数人の男達がいる。」という2種類の解釈が可能である。No.5の場合putという動詞が現在形と過去形とが同形であるために起こり、時制の一致の制限がゆるいために2つの構文木が得られる。No.6の場合、「彼が来た時に教えて下さい。」と「彼がいつきたのか教えて下さい。」という2種類の解釈が存在する。

解析時間を比較してみると、高速化された場合で2倍以上、wf_goalのassertが制限された場合では10倍以上短縮されている。これによって1単語あたり200~400msecで構文解析を終えることができている。この時間はすべての可能性を調べ終るまでの時間を単語数で割った値であるが、1つ目の構文木が得られるまでの時間を表2に示す。

例文	解析時間(msec)	単語数	単語当りの時間(msec/word)
1	2556	8	319.5
2	915	6	152.5
3	1663	8	207.9
4	931, 373	7	133, 53.3
5	3929, 195	12	327.4, 16.3
6	425, 850	5	85, 170
7	933	6	155.5
8	1764	8	220.5
9	2149	6	358.2
10	2168	5	433.6

表2. 構文木を得るまでの時間.

これより平均して1単語あたり240msecという時間で1つ目の構文木を得られ、BUPシステムが、文法規則が大きくなっても、高速に解析できることが実証された。

4. 結論

本研究ではDCG形式で約400からなる比較的大きな英語文法を開発した。文法規則中にprologプログラムで記述

した補強項を埋込むことで、構文レベルの曖昧さの除去を図り、意味情報をほとんど用いずに、構文レベルの曖昧さを除去できることを確認した。

又、補強項を用いることで、メモリーや解析時間をも少なくすることができることを確認した。そして、BUPシステムで約400の文法規則を用いて構文解析を行なうと1単語あたり平均240 msec という満足のいく値が得られ、BUPシステムのツールとしての有用性が実証された。

現在の英語文法において、省略語の処理に関する規則が冗長であると考えられる。そこで、Extrapositional Grammarの導入が望まれる。又、省略語と同様に挿入句の対策が大きな問題となる。更に、辞書項目の増加に伴って検索時間が増加してきており、辞書構造、辞書引きの方法を検討していく必要がある。

謝辞

本研究にあたって、計算機使用の機会を与えて下さるなど、多大な御援助をいただきました渕一博ICOT研究所長に感謝します。また、日頃、討論いただきました東工大情報工学科田中研究室の諸氏に感謝します。

[参考文献]

[Pereira 80] 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, pp 231-278, May 1980

[Pereira 81] Pereira, F.
"Extrapositional Grammar"
American Journal of Computational Linguistics Vol 7 No 4
Oct-Dec. 1981

[松本 他83a] 松本裕治, 田中穂積, 平川秀樹, 三宅秀夫, 安川秀樹, 向井国昭, 横井俊夫
"Prologに埋め込まれたボトムアップパーサ: BUP"

[松本 他83b] 松本裕治, 清野正樹, 田中穂積
"BUPトランスレータ — 文法規則から構文解析プログラムの自動生成"
電子技術総合研究所彙報 Vol 47, 第5号
Apr. 1983.

[松本 他83c] 松本裕治, 田中穂積, 清野正樹
"BUPの高速化"
情報処理学会 自然言語処理研究会 39-7 Spt. 1983

[Hornby] A.S. Hornby
"オックスフォード 現代英英辞典"
開拓社.

[Robinson 80] Jane. J. Robinson
"DIAGRAM: A Grammar for Dialogues."
Technical Note 205 (SRI) Feb. 1980.

Appendix 文法規則

```
ADJCOMP  :  Adjective Complements
  for him, to go there (as in :
     difficult for him to go there)
  that he went (as in :
     possible that he went)

adjcomp --> { [enough] / infinitive }.
adjcomp --> [for],np,(infinitive).
adjcomp --> enough,infinitive.
adjcomp --> enough,[for],np,
            (infinitive).
adjcomp --> [that],sdec.
-----------------------------------------

ADJP  :  Adjective Phrases
  too big, more difficult than that,
  easy for him to do, as ready to go
  as he is, necessary that it is done

adjp --> adj,(adjcomp).
adjp --> qdet,adj,(adjcomp).
adjp --> qpp,qdet,adj,(adjcomp).
adjp --> ddet,adj,(adjcomp).
adjp --> [more],adjp.
adjp --> a,([more]),adjp.
adjp --> detq,([more]),adjp.
adjp --> detq,a,([more]),adjp.
adjp --> adjp,thancomp,(adjcomp).
adjp --> as,adjp,(ascomp).
adjp --> not,as,adjp,ascomp.
adjp --> not,so,adjp,ascomp.
adjp --> [most],adj,(adjcomp).
adjp --> adjp,(comma),paraconj,adjp.
-----------------------------------------

ASCOMP  :  Adjective Complements for
           Equality Comparisons
  as that, as he appears to be,
  as it is

ascomp --> as,{ np / sdec }.
ascomp --> as,subj,(auxd),vp2.
ascomp --> as,subj,(aux),bep.
-----------------------------------------

ADVP  :  Adverb Phrases
  as fast as that, too often, more
  carefully than John did it, when he
  is here, if they come

advp --> adv.
advp --> qdet,adv.
advp --> qpp,qdet,adv.
advp --> [more],adv.
advp --> detq,([more]),adv.
advp --> advp,thancomp.
advp --> as,advp,(ascomp).
advp --> q,adv.
advp --> p,sdec.
advp --> subconj,sdec.
advp --> subconj,vp.
-----------------------------------------

AUX  :  Auxiliary Phrases
  could not have been -ing, can

aux --> havep,bep.
aux --> modalp,(adv),havep,bep.
aux --> havep.
aux --> modalp,(adv),havep.
aux --> bep.
aux --> modalp,(adv),bep.
aux --> modalp,(adv).

AUXD  :  Do-type Auxiliary Phrases
  do not, could have -ed

auxd --> aux.
auxd --> dop.
-----------------------------------------

BEP  :  BE Auxiliary Phrases
  is not, am, being

bep --> be,(not).
-----------------------------------------

DDET  :  Definite Determiner Phrases
  the many, all five, these two,
  not all these five

ddet --> { det / all }.
ddet --> not,all.
ddet --> all,det.
ddet --> not,all,det.
ddet --> det,number.
ddet --> det,qpp.
ddet --> all,number.
ddet --> all,qpp.
ddet --> not,all,number.
ddet --> not,all,qpp.
ddet --> all,det,number.
ddet --> all,det,qpp.
ddet --> not,all,det,number.
ddet --> not,all,det,qpp.
-----------------------------------------

DETQ  :  Determiner / Quantifier
         Phrases
  many, much more, too many more,
  two, no two

detq --> qpp.
detq --> a,q,qpp.
detq --> ddet,q,(qpp).
detq --> number.
detq --> [the],q.
-----------------------------------------

DOP  :  DO-Auxiliary Phrases
  do, does not

dop --> do,(not).
-----------------------------------------

GERUND  :  Gerund Phrases
  being informed, not having realized
  arriving

gerund --> vp.
gerund --> not,vp.
gerund --> have,vp.
gerund --> not,have,vp.
gerund --> be,vp.
gerund --> not,be,vp.
gerund --> have,be,vp.
gerund --> not,have,be,vp.
gerund --> be,pred.
gerund --> not,be,pred.
gerund --> have,be,pred.
gerund --> not,have,be,pred.
-----------------------------------------

HAVEP  :  HAVE-Auxiliary Phrases
  has, had not

havep --> have,(not).
```

```
SDEC  :  Declarative Sentences
  they went there, it can be difficult
  they might have been going there then
  there are some apples in the basket

sdec --> subj,(adv),bep,(adv),(pred).
sdec --> subj,aux,(adv),bep,(adv),(pred)
.
sdec --> subj,(adv),(auxd),vp.
sdec --> there,(aux),bep,subj.
sdec --> there,(aux),bep,subj,pred.
sdec --> vp3,comma,sdec.
sdec --> sdec,comma,vp3.
sdec --> sdec,comma,infinitive.
sdec --> sdec,(comma),advp.
-----------------------------------------

SENTENCE  :  Sentences

sentence --> sdec.
sentence --> sdec,comma,scmp.
sentence --> sdec,(comma),conj,sdec.
sentence --> sdec,(comma),paraconj,
             sdec.
sentence --> sdec,(comma),paraconj,vp1.
sentence --> simp.
sentence --> simp,comma,scmp.
sentence --> simp,(comma),paraconj,
             sdec.
sentence --> simp,(comma),paraconj,
             simp.
sentence --> simp,(comma),paraconj,vp1.
sentence --> simp,(comma),conj,simp.
sentence --> simp,(comma),conj,sdec.
sentence --> sq.
sentence --> swhq.
sentence --> pp,(comma),sentence.
sentence --> conj,sentence.
sentence --> advp,comma,sentence.
-----------------------------------------

SIMP  :  Imperative Sentences
  put them on the table, do be careful
  don't do that

simp --> be,(adv),pred.
simp --> dop,be,(adv),pred.
simp --> vp.
simp --> dop,vp.
simp --> simp,(comma),advp.
-----------------------------------------

SQ  :  Yes/No Interrogatives
  Is he going? Are there any more?
  Did he go? Can there be a man?
  Have you ever been to America?

sq --> bep,subj,(adv),{ pred /
       be,(adv),pred }.
sq --> modalp,subj,(adv),bep,(adv),
       (be),pred.
sq --> modalp,subj,(adv),havep,bep,
       (adv),(be),pred.
sq --> havep,subj,(adv),be,(adv),(be),
       pred.
sq --> dop,subj,(adv),vp.
sq --> modalp,(adv),subj,(adv),vp.
sq --> modalp,subj,(adv),bep,(adv),vp.
sq --> modalp,subj,(adv),havep,(adv),
       vp.
sq --> modalp,subj,(adv),havep,(adv),
       bep,vp.
sq --> havep,subj,(adv),vp.
sq --> havep,subj,(adv),bep,(adv),vp.
sq --> bep,subj,(not),(adv),vp.
sq --> bep,there,subj,(pred).
sq --> modalp,there,(not),bep,subj,
       (pred).
sq --> modalp,there,(not),havep,bep,
       subj,(pred).
sq --> havep,there,bep,subj,(pred).
sq --> sq,(comma),advp.
-----------------------------------------

SQ1  :  Yes/No interrogatives that
        lack object or predicate
  What did you see? What can you do?
  Who are you? Where have you been?

sq1 --> dop,subj,(adv),vp2.
sq1 --> modalp,(adv),subj,(adv),vp2.
sq1 --> modalp,subj,(adv),bep,(adv),
        vp2.
sq1 --> modalp,subj,(adv),havep,(adv),
        vp2.
sq1 --> modalp,subj,(adv),havep,(adv),
        bep,vp2.
sq1 --> havep,subj,(adv),vp2.
sq1 --> havep,subj,(adv),bep,(adv),vp2.
sq1 --> bep,subj,(not),(adv),vp2.
sq1 --> bep,subj.
sq1 --> modalp,subj,(adv),bep.
sq1 --> modalp,subj,(adv),havep,bep.
sq1 --> havep,subj,(adv),be.
sq1 --> sq1,(comma),advp.
-----------------------------------------

SQ2  :  Yes/No Interrogatives that
        lack subject
  Who is there? What is there hanging
  on the wall?

sq2 --> bep,there,
        ({ vp / be,pred / pred / srel }).
sq2 --> modalp,there,(not),bep,
        ({ vp / be,pred / pred / srel }).
sq2 --> modalp,there,(not),havep,bep,
        ({ vp / be,pred / pred / srel }).
sq2 --> havep,there,bep,
        ({ vp / be,pred / pred / srel }).
sq2 --> sq2,(comma),advp.
-----------------------------------------

SREL  :  Relative Clauses
  who went there yesterday, to whom he
  gave it, that was there, that you saw
  yesterday, the watch he gave

srel --> relpro,(auxd),vp.
srel --> relpro,(aux),bep,pred.
srel --> subj,(auxd),vp2.
srel --> subj,(aux),bep.
srel --> p,relpro,sdec.
srel --> relpro,subj,(auxd),vp2.
srel --> relpro,subj,(aux),bep.
srel --> comma,srel.
srel --> srel,(comma),pp.
-----------------------------------------

SUBJ  :  Subject Phrases

subj --> np.
```

```
INFINITIVE  :  To-Infinitive Phrases
  to have gone, to be informed,
  not to go, to have been very care-
  ful about that

infinitive --> to,vp.
infinitive --> not,to,vp.
infinitive --> to,be,
               { pred / vp / be,pred }.
infinitive --> not,to,be,
               { pred / vp / be,pred }.
infinitive --> to,have,vp.
infinitive --> not,to,have,vp.
infinitive --> to,have,be,
               { pred / vp / be,pred }.
infinitive --> not,to,have,be,
               { pred / vp / be,pred }.
-----------------------------------------

INFINITIVE1  :  Root-Infinitive Phrases
  go (as in : made him go)

infinitive1 --> vp.
infinitive1 --> not,vp.
infinitive1 --> be,
                { pred / vp / be,pred }.
infinitive1 --> not,be,
                { pred / vp / be,pred }.
infinitive1 --> have,vp.
infinitive1 --> not,have,vp.
infinitive1 --> have,be,
                { pred / vp / be,pred }.
infinitive1 --> not,have,be,
                { pred / vp / be,pred }.
-----------------------------------------

INFINITREL  :  Nonfinite Relative
       Clauses
  the man to whom to give it
  the thing for you to do
  the part being attached

infinitrel --> { infinitive / vp /
                 be,vp }.
infinitrel --> [for],np,{ infinitive /
                 vp / be,vp }.
infinitrel --> p,relpro,{ infinitive /
                 vp / be,vp }.
-----------------------------------------

MODALP  :  Modal Auxiliary Phrases
  can, may, must, will, shall, cannot

modalp --> modal,(not).
-----------------------------------------

NCOMP  :  Noun-Phrase Complements
  of tea, on the corner, that I saw
  good cook than he

ncomp --> { pp / of,np }.
ncomp --> { vp2 / srel / infinitrel /
            adjp / thancomp }.
ncomp --> ncomp,pp.
-----------------------------------------

NOMHD  :  Nominal Heads
  very big task, broken vases, running
 steams, more difficult task

nomhd --> n.
nomhd --> { adjp / v },nomhd.
nomhd --> qpp,adjp,nomhd.
```

```
NP  :  Noun Phrases
  water, cats, the length of that board
  as big a box as you could carry, two
  of them, fish and meat

np --> nomhd,(ncomp).
np --> a,nomhd,(ncomp).
np --> ddet,nomhd,(ncomp).
np --> detq,nomhd,(ncomp).
np --> detq,q,nomhd,(ncomp).
np --> a,ncomp.
np --> ddet,(ncomp).
np --> detq,(ncomp).
np --> detq,q,(ncomp).
np --> pron,(refl).
np --> pron,ncomp.
np --> ddet,([most]),adjp,nomhd,
       (ncomp).
np --> ddet,([most]),adjp,ncomp.
np --> gerund.
np --> det,gerund.
np --> as,qpp,(ascomp).
np --> as,{ adj / qpp },(of),np,
       (ascomp).
np --> as,qpp,nomhd,(ascomp).
np --> as,(qpp),adj,(of),np,thancomp,
       (ascomp).
np --> [that],sdec.
np --> np,comma,np.
np --> np,(comma),paraconj,np.
-----------------------------------------

OBJ  :  Object Phrases

obj --> np.
obj1 --> np.
obj2 --> np.
-----------------------------------------

PP  :  Prepositional Phrases
  on it, after that, within A of B,
  between A and B

pp --> p,obj.
pp --> [within],np,of,np.
pp --> [between],np,[and],np.
-----------------------------------------

PRED  :  Predicate Phrases
  that is very heavy, he is a student,
  that wasn't in the box, he was
  attached in it

pred --> { adjp / np / pp / vp2 }.
pred --> pred,(comma),pp.
pred --> pred,(comma),infinitive.
-----------------------------------------

QPP  :  Indefinit Quantifiers Phrases
  much, too much, very much

qpp --> q.
qpp --> qdet,q.
qpp --> q,qpp.
qpp --> qdet,q,qpp.
-----------------------------------------

SCMP  :  Sentence Complements
  he ran, swam, and rode a bicycle

scmp --> sdec,comma,scmp.
scmp --> vpl,comma,scmp.
scmp --> sdec,(comma),paraconj,sdec.
scmp --> sdec,(comma),paraconj,vpl.
scmp --> vpl,(comma),paraconj,sdec.
scmp --> vpl,(comma),paraconj,vpl.
```

```
SWHQ  :  Wh-Questions
  Who is it? Where do you go?
  With whom are you going?

swhq --> whnp,sq1.
swhq --> whpp,sq.
swhq --> whadjp,sq1.
swhq --> whadjp,sq.
swhq --> whnp,sq2.
swhq --> whnp,(auxd),vp.
swhq --> whnp,(aux),bep,(pred).
----------------------------------------

THANCOMP  :  Than Complements
  I am prettier than he is.
  He can run faster than I.

thancomp --> than,np.
thancomp --> than,sdec.
thancomp --> than,subj,(auxd),vp2.
thancomp --> than,subj,(aux),bep.
----------------------------------------

VP  :  Verb Phrases
  go there in September, make him go
  move it from here to there,
  seem to be difficult, give her a pen,
  break it easily, break easily

vp --> v.
vp --> v,(p),advp.
vp --> v,np.
vp --> v,adjp.
vp --> v,infinitive.
vp --> v,vp.
vp --> v,be,pred.
vp --> v,obj.
vp --> v,refl.
vp --> v,obj1,obj2.
vp --> v,obj2,of,obj1.
vp --> v,obj,infinitive.
vp --> v,obj,infinitive1.
vp --> v,infinitive.
vp --> v,sdec.
vp --> v,(np),relpro,sdec.
vp --> v,(np),whpp,
       { sdec / infinitive }.
vp --> v,(np),whnp,
       { sdec / infinitive }.
vp --> v,(np),whadjp,
       { sdec / infinitive }.
vp --> v,obj,adjp.
vp --> v,obj,vp.
vp --> v,obj,be,pred.
vp --> v,advp,obj.
vp --> v,obj,advp.
vp --> vp,(comma),pp.
vp --> vp,(comma),infinitive.
vp --> vp,(comma),paraconj,infinitive.
vp --> vp,(comma),paraconj,adjp.
vp --> vp,advp.
----------------------------------------

VP1  :  Be-type Verb Phrases

vp1 --> vp.
vp1 --> bep,(adv),pred.
```

```
VP2  :  Verb Phrases that lacks object
  What did you see? He was attached to
  it. What did you take her?

vp2 --> v.
vp2 --> v,obj1.
vp2 --> v,p,obj1.
vp2 --> v,of,obj1.
vp2 --> v,infinitive.
vp2 --> v,infinitive1.
vp2 --> v,adjp.
vp2 --> v,vp.
vp2 --> v,be,pred.
vp2 --> vp2,pp.
vp2 --> vp2,advp.
----------------------------------------

VP3  :  Participal Constructions
  feeling tired, being ill, not being
  diligent, having taken a bath

vp3 --> vp.
vp3 --> have,(adv),vp.
vp3 --> be,(adv),pred.
vp3 --> have,be,(adv),pred.
vp3 --> not,vp3.
vp3 --> [never],vp3.
----------------------------------------

WHADJP  :  Interrogative Adjective
           Phrases
  how big, how much more easy to do

whadjp --> how,(adjp).
----------------------------------------

WHDET  :  Intrrrogative Determiners
  how much, how many more, whose, what

whdet --> how,q.
whdet --> how,q,(qpp),q.
whdet --> whn.
----------------------------------------

WHNP  :  Interrogative NPs
  how many more women, how much water,
  whose book, which two

whnp --> whdet,(ncomp).
whnp --> whdet,number,(ncomp).
whnp --> whdet,nomhd,(ncomp).
whnp --> whdet,number,nomhd,(ncomp).
----------------------------------------

WHPP  :  Interrogative PPs
  where, when, at what time, from where

whpp --> whp.
whpp --> p,whnp.
whpp --> p,whp.
```

注) ここで書かれている文法規則の記法は実際のものと異なる。実際の記法は本文中に示したように3引数を持つ。又、{A／B}という記法は「A又はB」を表したもので実際は2つの文法規則に分けて記述してある。(comma)は省略可を表し(comma;[])の略である。又、補強項はすべて省いてある。