

# 国立国語研究所学術情報リポジトリ

## 構文解析自動化の研究 2 : 文構造解析のプログラムからプログラム言語へ

メタデータ	言語: Japanese 出版者: 国立国語研究所 公開日: 2017-03-31 キーワード: 作成者: 木村, 繁, KIMURA, Sigeru メールアドレス: 所属:
URL	<a href="https://doi.org/10.15084/00001002">https://doi.org/10.15084/00001002</a>

## 構文解析自動化の研究 II

—文構造解析のプログラムからプログラム言語へ—

木村 繁

### 0. はじめに

本報告は自動単位切りの方法と HITAC 3010による AUTOSEG(AUTOMatic SEGmentation) システム試作・実験\*のうち、文構造の解析=検定を中心に述べている。

自動単位切りの方法としては、テーブル・ルック・アップ方式\*\*であって、辞書の語形による単位切り、接続規則による付属的な語の接続・接合のチェック、文法規則による文構造の解析=検定を行なうものである。従って、システムの処理手続きとしては大筋の処理基準だけを定め、その枠のなかで User 各自の文法\*\*\*で記述する辞書・接続規則・文法規則によって処理される。

また、語形による単位切り・文法規則による構文解析は、最長一致法などによる優先を認めていない。一意の解だけではなく、可能な限りの解を求めている。辞書・規則類は磁気テープを使用していることもあって、とりあえずはハードウェア上の制限による処理時間などの経済性は無視している。

§ 1 では自動単位切りのシステムについて、§ 2 の構文分析のアルゴリズム化に関する説明に必要な点を中心に概括的に述べている。§ 3 では、この研究の目的・応用そして今後の問題などに触れる。

---

注) \* AUTOSEG システムは、石綿敏雄のシステム・アナリシスに基づく齋藤秀紀および筆者の3人の共同研究である。システム全体にわたる詳細な報告は『計量国語学』に発表する予定。

\*\* 字種や文字の出現のしかたの確率・環境をもとに、プログラム派による自動単位分割の方法については、『計量国語学』43/44号の江川 清「漢字かなまじり文の『自動単位分割に関する一研究』を参照。

\*\*\* 本報告に用いたルールは説明のためのもので一貫した形をとっていない。

## 1. 自動単位切りのシステム

本システムは現時点において3つのプログラム・セグメントからなりたっている。〔図1 AUTOSEG のシステム・チャート参照〕

漢字テレタイプないしはフレキシソ・ライターの穿孔テープを1文（センテンス）\* ずつ入力し、辞書の語形により単位切りと情報転写を行ない（AUTOSEG 1）、次に付属的な語の接続検定と接合によってメタ言語の列を作り（AUTOSEG 2）、文法規則によって文構造の解析＝検定を行なって適格なもののみを OUTPUTする（AUTOSEG 3）。

なお、AUTOSEG 3は context free の P.S. 文法（Phrase Structure Grammar）で書かれたルールによって文を解析している。それで書き切れない部分を取り除くために、照応陳述（照応と付加）検定\*\*（AUTOSEG 4）を加える予定で現在 Computable な公式化を進めている段階である。

### 1. 1 辞書の語形による単位切りと情報転写

入力された原文を辞書に出てくる語形に合わせて切り、単語の情報を転写する。ことばの意味のつながり方に関する検定は行なわれない。単位切りされた単語の列全体がとにかく形だけすっかり原文にあえば、AUTOSEG 1では合格する。そういう切り方はいくつもありうるから、可能な限りの切り方をすべて出力する。

辞書=DIC (TIONARY) の項目

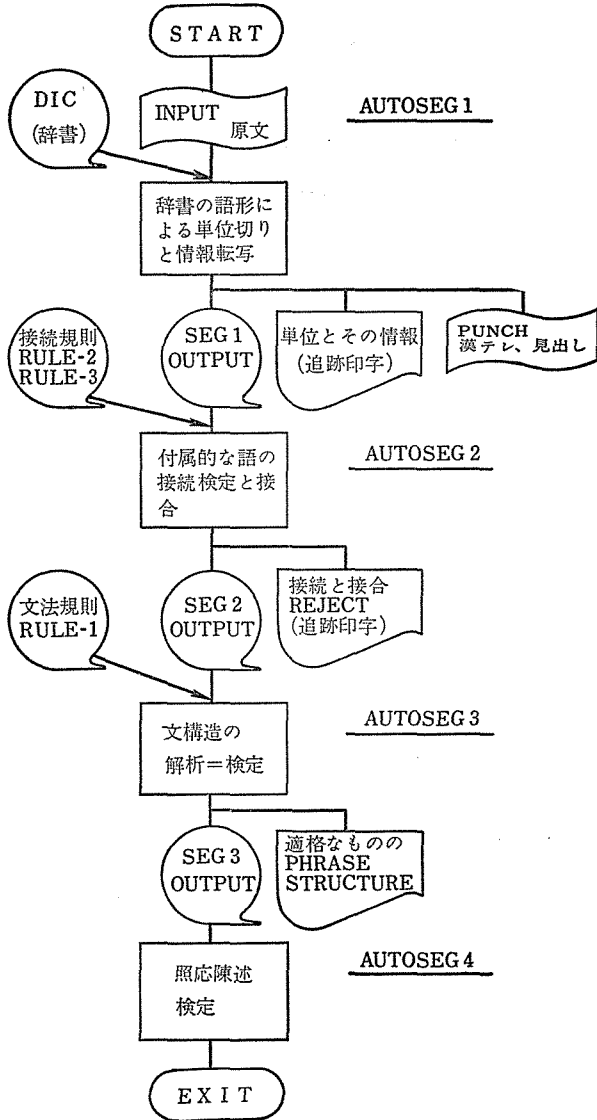
1. 見出し語の桁数……この項目以外は可変長で、項目間は I S S マーク(75)で区切っている。見出し語の比較の桁数として用いる。なお、漢テレ1字はH-3010コード2桁にあたる。
2. 見出し語……用言の活用形、同音異語、異品詞語など、転写すべき辞書の

---

注) \* § 3, 2で述べるが、このシステムは単に自然語の自動単位切りだけに使うのではなく、プログラム言語など人工語の文法研究のための実験道具としても役立つ。広義の意味での文である。

\*\* 本報告書所収石綿敏雄・「構文解析自動化I」の7.を参照。

〔図1〕 AUTOSEGシステムチャート



情報が異なるものはすべて別々に見出し語をたてる。

3. 各種表記……例えば、処理過程を追跡印字できるように、ラインプリンタで印字可能なローマ字ないしはカナ文字表記など。

[例] (桁数) (見出し) (各種表記)

1) 4 谷川 ・TANIGAWA\*タニカ\*ワ・

2) 5 タニカ\*ワ・TANIGAWA\*谷川・

4. メタ言語……説明は § 1・2

4-1) 語のグループ分け

4-2) 品詞細分

5. 照応陳述関係

6. 文法情報 } ……説明は § 1・2  
7. 接続情報 }

[辞書の内容・例]

(桁) (見出し) (ローマ字表記) (メタ言語) (陳述) (文法情報) (接続情報)

4	谷川	TANIGAWA	X	N	N	
6	小さな	TIISANA	X	KD	KD-10	
2	の	NO	B	PA	PA	1
4	まし	MASI	A	AUXG	AUX-G-7	3
2	,	,		COMMA		
2	。	。	=	PERIOD		

[AUTOSEG 1 の処理・例]

(INPUTの一部) はながさく

(辞書の見出し語の語形) はな, はながさ, ながさ, が, さく, は, く

(OUTPUT) 次の3通りの単位の切り方が出力される。

1) はな が さく

2) はながさ く

3) は ながさ く

## 1. 2 付属的な語の接続検定と接合

AUTOSEG 1 の OUTPUT について、文頭の単語から順々に付属的な語 (A, B, Cグループの語) をとりあげ、磁気テープに書かれた接続規則(RULE-2) に照合して、前の語との接続を検定し、具合の悪いものは REJECT する。

付属的な語のうちAグループに属すものについては、RULE-3とのテーブル・ルック・アップによって、前の語のメタ言語を書き換え、Aグループの語のグループ分けを SPACE (空白) にして、次の構文処理にそなえる。例えば、用言に助動詞 (=複語尾) をつけて1つにまとめるとか、あるいは、助動詞どうし連続しているものをまとめることができる。これらの処理はこのシステムを使用する人の D I C (辞書) の記述のしかた、および文法の書き方によって自由にきめられる (助動詞を複語尾のように取り扱っても扱わなくてもよい)。

OUTPUT は AUTOSEG 1 の出力番号とメタ言語の列からなる。なおその処理過程は順次印字され、接続検定に不合格のものはその時点で REJECT として処理を切りあげ、次の AUTOSEG 3 のための出力はされない。

### メタ言語のグループ分け

- 1) 自立的な語 X群……AUTOSEG 3 の自立項になりえる語。
- 2) 付属的な語 A群……接続検定を行ない、前の語に接合する語。  
B群…… // , 接合しない語。  
C群…… // , 接合しない語。\*
- 3) 情報無視の語 SPACE 群……単位切りされた語としては意味をもつが、処理されるときは NO EFFECT。接合により前後に情報をまとめ、以後の情報を失ったときや、注記などに用いる。
- 4) 文末語 =群……文末を示す文字・記号・語。

[AUTOSEG 2 の処理・例]

(辞書接続情報) 見出し語と RULE-2の適用番号を示す。

---

注) \* B, C群は § 1・3 で述べる、文法規則の適用範囲が異なる。

の 1, に 1, が 1, で 1, は 1, まし 2, た 3  
 (RULE-2) ……接続検定ルール番号および接続可能な前の語の文法情報  
 を示す。

- 1 N,
- 2 V-4,
- 3 AUX-G-7, V-4, COP-7,

(RULE-3) ……Aのグループ語に適用され, 前のメタ言語が書き換え  
 られる接合ルール。

X V + A AUXG→X V  
 X V + A AUXF→X V  
 X COP + A AUXF→X COP

(追跡印字の例)

見出し	メタ言語	文法情報	接続情報	書き換えされる前のメタ言語
1) 学校	X N	N		
が	B PA	PA	1	
あり	X V	V-4		
まし		AUXG	AUX-G-7 2	X V
た		AUXF	AUX-F-9 3	X V
2) 朝	X N			
でし	X COP	COP-7		
た		AUXF	AUX-F-9 3	X COP
3) 朝	X N	N		
で	B PA	PA	1	
し	X N	N		
た		AUXF	AUX-F-9 3	DATA REJECT

### 1. 3 文構造の解析=検定

AUTOSEG 2 の OUTPUT について, X グループの語を自立項として隣り  
 あった (交差しない) 1 ~ 3 項間において合成が可能であるかを, P. S. 文

法の inversion rules で書かれた句の文法規則 (RULE-1) で判定する。可能であればメタ言語を書き換えてまとめていき、全体として単一の自立項にまとまったとき、文の文法規則によって文として解釈できるかをチェックする。1つのまとまった文であると判定した場合に、その単位切りは妥当な切り方として適、そうでない場合は不適とする。

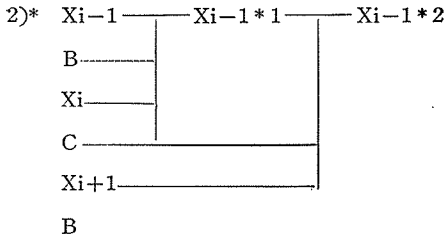
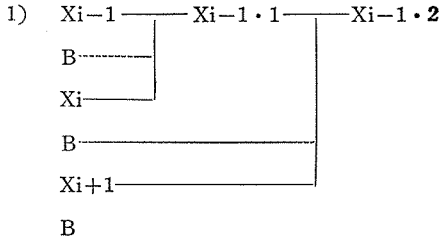
文の検定のしかたとしては、まず自立項 1 ~ 3 項間で合成可能なすべての項をとる。次に句の文法規則とテーブル・ルック・アップによって照合して、文法的に可能な句をつくる。そして項が互に交差しない組み合わせをとり、自立項を書き換える。順々に次の段階 (レベル) での合成を繰り返して、単一の自立項および文の解釈をチェックしていく。文の解釈をした後、またはあるレベルで句の合成が 1 つも存在しないで中断したとき、前の組み合わせを分解する。そして次の新しい組み合わせがあるかをチェックして、できる限りの組み合わせを作り、可能な限りの文構造を解析する。

AUTOSEG 1 の単位切りのうち、AUTOSEG 2, 3 の検定に合格したものについて、各単語および合成された句のメタ言語とそのまとまった過程 (= 文の構造) の情報をつけて OUTPUT する。

#### Syntactic Combination (ルールの適用範囲)

1. 自立的な語のグループ X の単語を自立項として、1 つから 3 つまでの自立項の結合を認め、それを 1 項則、2 項則、3 項則とする。可能な限りの組み合わせを調べることを前提としているので、最長一致などの優先を認めない。結合されて一項に書き換えられる項も自立項である。
2. 付属的な語のうち、A グループは AUTOSEG 2 で情報無視 (グループは Ⓣ) になる。B, C グループは自立項の間に何個あってもよい。しかし、最後の項は自立項または C グループの項でなければならない。
3. 下図の例のように、ルール範囲の最後の自立項  $X_i$  に続く C グループの項は  $X_i$  とともにルールの適用範囲には入るが、書き換えには影響されないで、次の  $X_{i+1}$  と結合するときに再び適用範囲内の項に入る。

〔例〕



## 2. 構文分析のアルゴリズム

§ 1. 3 で構文分析の概略を述べたが、§ 2 では処理のアルゴリズム化についてふれたい。とりあえずはスピードを無視して処理の論理手順を把握することを主眼としたため、プログラム言語としてはディテールなフローチャートが必要とせず、かつプログラムそれ自体がドキュメントとしての性格をもつ COBOL によった。この章では、主として計算機の内部表現をもとに話をすすめるので、初めに INPUT および OUTPUT における内部表現と外部表現との関係を述べる。

### 2. 0 OUTPUT の内部表現と外部表現

ここでは、なんらかの方法で AUTOSEG 1 単位切りと AUTOSEG 2 接続検定などの処理が行なわれた結果であるメタ言語の列が INPUT され、1 つ

---

注) \*①パン+を+たべ+ます

②パン+が+たべ+たい

②は(が)と(たい)が対応していて、「パンがたべます」とは言えない。

XiのあとのCもルールの適用範囲に入れるとチェックできる。

の文が解釈されたものとする。従って、ここでは自然語との対応づけは行わず、メタ言語で記述する。

分割された単語のメタ言語に対応して項をとり、単位語間の結合関係を表わす。そのパラメータとして、L, M, NとV, Hの2種をとる。

内部表現

表1. IN-FILE

HEAD		1 2 3 4
IN-ITEMS		
IN-META (1)		
01	X AD	
02	X N	
03	B P · D	
04	X N	
05	B P · B	
06	X N	
07	C P · A	
08	X V · 10	
09	X KD · 10	
10	X N	
11	X COP · 9	
12	AUXF · 9	
13	= PERIOD	
	E/I	

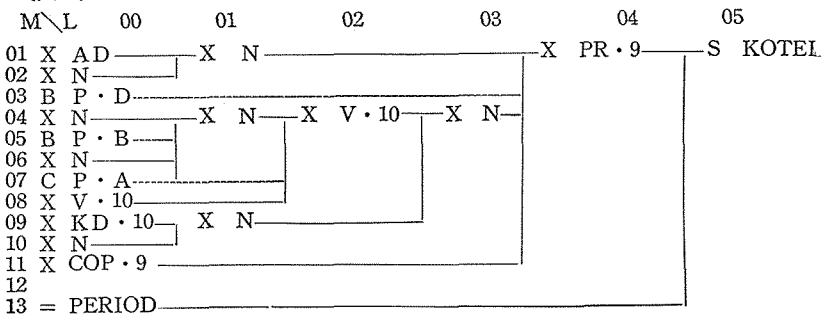
表2. OUT-RECORD

SEG 1-OUT-NO	1 2 3 4
SEG 3-OUT-NO	0 1
CONTINUE	==
REWRITTEN-PHRASE (J)	
(J)	L M N V H NEW-META
001	00 01 02 000 002 X AD
002	00 02 03 000 000 X N
003	00 03 04 000 017 B P · D
004	00 04 05 000 005 X N
005	00 05 06 000 006 B P · B
006	00 06 07 000 000 X N
007	00 07 08 000 008 C P · A
008	00 08 09 000 000 X V · 10
009	00 09 10 000 010 X KD · 10
010	00 10 11 000 000 X N
011	00 11 13 000 000 X COP · 9
012	00 13 == == == PERIOD
013	01 01 03 001 003 X N
014	01 04 07 004 007 X N
015	01 09 11 009 000 X N
016	02 04 09 014 015 X V · 10
017	03 04 11 016 011 X N
018	04 01 13 013 012 X PR · 9
019	05 01 == 018 000 S KOTEI
020	E/I

外部表現

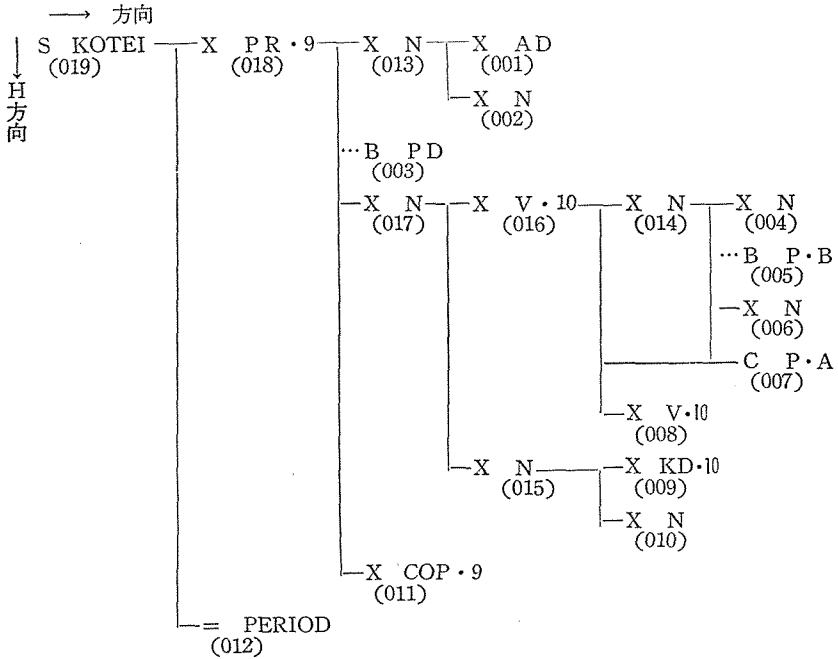
〔図2〕 L, M, N による外部表現

自立項の結合は実線で、  
付属的な項は点線で示す。



【図3】 V, H による外部表現

メタ言語の下の ( ) 内の数字は表2のITEM# Jを示す。



1) L, M, N

L…レベル；INPUT したメタ言語に対応する項のレベルを0とし，項の結合の段階を示す。

M…項ナンバー；文頭からの順序を示す番号。文頭の項を01として順に，02, 03, ……とする。

N 後方向への結合すべき項ナンバー。

2) V, H

V…書き換えされる前の句の位置を示す。表2・OUT-RECORD の REWRITTEN-PHRASE の添字番号Jで表わす。

(いわば，垂直方向ないしは親子の関係を示す)

H…書き換えられた項のルール適用範囲を示し，番号Jで表わす。

(いわば，水平方向ないしは兄弟の関係を示す)

表1の INPUT が表2のような文構造に解釈および内部表現されて，OUTPUT としては，図2，図3のように表現できる。



表3 X-ITEMS

[項の初期値]

[自立項の書き換え]

X (1) XJ XI XN

01	001	02	02	013 04 03			018 13 13
02	002	04	03				
03	003	00	04				
04	004	06	05	014 08 07	016 09 09	017 11 11	
05	005	00	06				
06	006	08	07				
07	007	9R	08				
08	008	09	09				
09	009	10	10	015 11 11			
10	010	11	11				
11	011	13	13				
12	000	00	00				
13	012	==	==				

レベル

L=00

L=01

L=02

L=03

L=04

## 2.1 メタ言語の列の読み込みと初期値設定

- 1) 表1のようなメタ言語の初 IN-FILE を読み込み,
- 2) 表2のうち, L=00の OUT-ITEM を作成し,
- 3) 表3のような自立項の表 X-ITEMS の初期値を設定する。

X-ITEMS は XJ XI XN から構成されている。

XJ……現時点における(添字Iの)項のメタ言語の値をもつ表2の ITEM  
# J で表わす。

XI……後方向への次の自立項ナンバー。ただしCグループに対応する項  
では負の数-99 (H-3010コードで9R) とする。

XN……後方向への結合すべき項ナンバー。

## 2.2 句の合成とテーブル・ルック・アップ

- 1) レベル・アップ (L=L+1) をする。
- 2) 合成可能な項をすべてとる。(表4 LEVEL-GOSEI 参照) 合成可能とは1~3項則の適用範囲の中に少なくとも1つ, 前のレベルにおいて書き換えられた自立項が存在することである。

表4 LEVEL-GOSEI と KUMIAWASE 表

L=02からL=03にレベルアップし、合成可能な項をとった時の状態

4-1)

LEVEL-GOSEI (K)		ITEM									
		LINK	NO	L	M	N	V	I	LG-META		
*	001	003	2	01	01	03	001	02	X	N	
	002	004	2	01	02	05	002	04	X	PR・N	
*	003	006	2	01	04	07	004	06	X	N	
	004	006	2	01	06	09	006	08	X	V・10	
	005	007	3	01	06	10	006	08	X	PR・9	
*	006	007	2	01	09	11	009	10	X	N	
	007	008	1	01	11	13	011	13	X	PR・9	
	008	===	=	==	==	==	==	==	==	=====	
	009	011	2	02	01	07	013	04	X	PR・N	
*	010	012	2	02	04	09	014	08	X	V・10	
	011	012	2	02	08	11	008	09	X	N	
	012	===	=	==	==	==	==	==	==	=====	
	013		2	03	01	09	013	04			
	014		3	03	01	11	013	04			
	015		1	03	04	09	016	09			
	016		2	03	04	11	016	09			
	017		3	03	04	13	016	09			
	018	===	=	==	==	==	==	==	==	=====	

4-2)

KUMIAWASE (H)      OUT-ITEM#Jとの関係

01	001	→	013
02	003	→	014
03	006	→	015
04	010	→	016
05			

なお、最後の自立項に続いた項がCのグループに対応するとき、Cの項はルールの適用範囲には入るが、項の合成では含まれない。従って項の合成演算される項の連糸の最後は自立項である。

3) 合成可能な項の適用すべきルールを内部ソートして、テーブル・ルック・アップに要する時間を短縮するのにそなえる。

4) テーブル・ルック・アップする。句の文法規則に合致したものは、その書き換えのメタ言語を転写する。(表5-1 参照)

R-Xは項則の数(表4のITEM-NOに対応)、REWRT-METAは書き換えられるメタ言語。RULE-META(M)は適用されるメタ言語の連糸の要素を表わす。

5) 文法規則のない合成項をLEVEL-GOSEI(表4)から削除して、文法

表5 文法規則 (RULE-1)

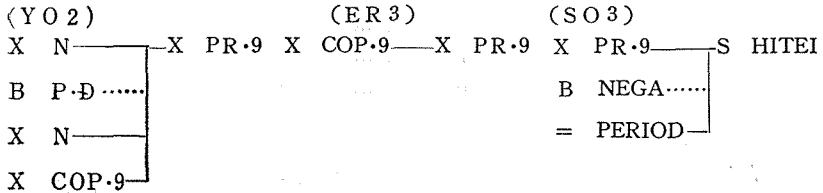
表5-1) 句の文法規則

RULE-ID	R-XREWRT-META	RULE-META	(1)	(2)	(3)	(4)
(TO1)	2	X N	X AD	X N		
(TO2)	2	X N	X N	B P · B	X N	C P · A
(TO3)	2	X N	X KD · 10	X N		
(TO4)	2	X N	X V · 10	X N		
(YO1)	2	X V · 10	X N	C P · A	X V · 10	
(YO2)	3	X PR · 9	X N	B P · D	X N	X COP · 9
(ER1)	2	X PR · N	X N	B P · D	X N	
(ER2)	3	X PR · 9	X N	C P · A	X V · 10	X KD · 10
(ER3)	1	X PR · 9	X COP · 9			

表5-2) 文の文法規則

(SO1)	0	S KOTEI	X PR · 9	= PERIOD	
(SO2)	0	S GIMON	X PR · 9	= QUESMARK	
(SO3)	0	S HITEI	X PR · 9	B NEGA	= PERIOD
(SO4)	0	S TOCHI	X V · 9	C P · C	= PERIOD

〔図5〕 文法規則の外部表現 (一部の例として)



的に合成可能な句を得る。同一レベル内に1つも文法的に合成された句が存在しないならば、1つ前のレベル ( $L=L-1$ ) にもどして、句の分解にとぶ。

### 2.3 句の書き換え

- 1) 同一レベル内で合成項が互に交差しないで次の句としてとることができるように、結合順を表4 (LEVEL-GOSEI) の LINK に作る。
- 2) LINK を使い、組み合わせ表 (表4-2: KUMIAWASE 参照) を作る。合成句としてこのレベルでとる表4-1の ITEM # K を KUMIAWASE 表にのせる。

- 3) KUMIAWASE 表\*にのせた合成句を OUT-ITEM にうつす。
- 4) 新しい自立項の表に書き換える。(表3のレベルごとの X-ITEMS の書き換えを参照)
- 5) 単一の自立項にまとまったならば、文のチェックにうつる。  
まとまらなかったなら、次のレベルの句の合成にとぶ。

## 2. 4 文の解釈

- 1) 単一の項と文末語の間のメタ言語について、(表5-2)のような文の規則のテーブル・ルック・アップを行なって、文として解釈できるかをチェックする。項則数を示す R-X が 0 であるルールが文の文法規則である。
- 2) OUT-RECORD を磁気テープに書き、ラインプリンタに文の構造を外

\*注)

ある文の自立項の数が N のとき、文法規則の句のルールがすべて 2 項則で書かれており、合成可能な項について適用するルールが全部存在したとする。その全組み合わせ数を SUMK (N) とすると、

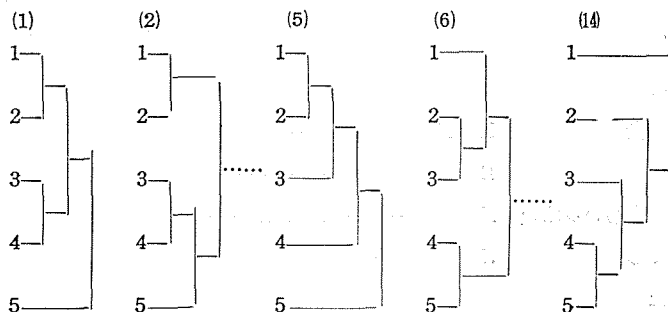
$$\text{SUMK}(1) = 1$$

$$\text{SUMK}(1) = \frac{2 * (2 * N - 3)}{N} * \text{SUMK}(N - 1) \text{ になる。}$$

すなわち、その値は

N	SUMK (N)	N	SUMK (N)
2	1	6	42
3	2		
4	5		
5	14	20	1, 886, 672, 865 となる。

上の条件の下で N=5 のときの組み合わせの順序は下図のようになる。





例 2-3)

08	日光	X	N	—	X	KD
09	は	B	PD	.....		
10	運動場いっぱい	X	KD	—		
11	でし	X	COP	—		
12	た					

## 2. 5 句の分解

- 1) KUMIAWASE 表の新しいものから (Push-downで), 合成句を分解し, 自立項をもとにもどす。
- 2) 同一レベル内で新しい組み合わせが可能であることをチェックしたり, 最後の組み合わせかをチェックしたりなどして, あらゆる可能な組み合わせをとるように変化させる。

## 3. 目的・応用と今後の問題点

### 3. 1 目的と今後の課題

この単位切りシステムは, テーブル・ルック・アップ方式を用いておるので, User のいろいろな文法で辞書・文法規則を記述し, 従来の国語学では方法として欠けていた実験・試行錯誤による検証が可能になる。その際, 単に一意の解釈が得られるのではなく, 可能な限りの解を求めることができるので, さらに実験手段としての意義は大きい。このことは日本語のLDPの基礎として機械翻訳・情報検索・自動抄録などを機械にのせようとする研究における第一歩である。

次に, CL (Computational Linguistics)の研究における問題発見のために, 語彙・文法をどのように研究すべきかという観点からこのシステムをながめてみる。このシステムを有効に利用するために, 語をどのように分類し, ルールをどのように組み立てるかということについて語彙・文法を十分に相互に関連された研究が必要であり, そのために, その観点から用語調査の結果を分析する

ことが必要である。ところで、我々が参加している語彙調査の分析において、例えば語の認定という問題がある。言語行動として人間が語の認定をしていくとき、あらかじめ経験などによって得られた知識やその体系化を行なう能力などの記憶をひきだして、前後の文脈をみて文を理解し意味・内容を限定していく。このような語の認定のための文解釈——表現～理解という言語行動——の過程を、例えばこの AUTOSEG のようなモデルによって、シミュレートし近似度を高めていくことになって、語彙調査における根本問題であるところの基本語彙・基本漢字への接近が得られるのではなからうか。

今回のシステムは処理時間などの経済性を無視し、言語行動の1つの近似モデルとしてのアルゴリズム化を主眼とした。機器構成などハードウェア上の制約もあるが、語彙調査などの実用化にたえることを目的とするならば、処理時間の短縮をはからなければならない。ソフトウェア上でも、例えば辞書に見出し語形がないとき、品詞の推測を行なう機能を含んでおらず、単位切りをその時点で中断する。また、現システムは3つのプログラム・セグメントが単に直列に並んでいる。これをデータの条件に応じてランダムにとりだして立体的に処理したり、そのほか例えば品詞の推測・ルールの推測機能を加えて、能率のよいコントロールができるよう改良することなども今後の課題である。第2プログラム・セグメントの接続検定においても、付属的な語の前に位置すべき語ばかりでなく、逆方向に次に来るべき語のチェックを加えることなども考えられる。このシステムは全体としてテーブルの照合によっているので、テーブルの配列、辞書・ルールの作成のしたか、テーブル・ルック・アップの方法なども処理時間の短縮に大きなウエイトをもつ。

しかし、このシステムにおける基本的な機能のもとで、辞書（語彙）・ルール（文法）の体系の記述を改良すること——例えば名詞の細分化とか活用情報・意味情報の付加——を実験しながら、このシステムの近似とその限界\*を知り、質的機能の拡大・改良をはかることが当面の大きな課題と思う。

---

\*注) P. S. 文法の限界に対して、第4セグメントとして照応陳述（照応と付加）検定を加える予定。

### 3. 2 プログラム言語への応用

自然語に対してプログラム言語は人工語であるが、自然語と同様に言語としての固有の問題を含んでいる。プログラム言語を大別すると、個々の計算機の固有の言語である機械語とほぼ1対1の対応をしているアセンブラーと、数式や日常語に近い形でのコンパイラーとに分けられる。しかし、その文章のシンタクス\*は根本的に違っている。アセンブラーはだいたい配列の順に処理される operational language である。これに対して、コンパイラーの1種である ALGOL は単純な羅列ではなく、かっこでくくられた複雑な構造を一貫としてっており、その文法はバックス記号で厳密に定義されている Phrase Structure Language である。

また、言語情報処理用コンパイラー\*\*は既にいくつか開発されている。しかし日本語の LDP, CL の研究のために、我々は漢字の処理を考慮したコンパイラーの必要を感じている。我々の試作した自動単位切りのシステムは、このような言語情報処理用プログラム言語の文法研究のためにも応用できるものと思われる。以下では、言語情報処理用ではないが、数式の処理においても言語に似た構造をもったルールで記述できることの例として、8を法とする演算式の処理をとりあげる。

1) の辞書によって AUTOSEG 1において、英数字・記号など各文字単位に単位切りが行なわれる。ただし、数字は0から7までしか用いられない。2) の接続規則により乗除記号と等号に関して接続検定をする。加減記号は正負の符号としても用いるので付属的な語に入れない。また、等号の左辺には英字しか認めないので、等号をCグループとしてチェックしている。3)の文法規則をもとに4)の RULE-FILE を作り、実験したところ、5)の第2行目のメタ言語に対して、唯一の構文解析の解を得られた。

---

注 \*) 参考文献(4)第3章4. ALGOL のSyntax-Phrase Structure を参照。

\*\*\*) 非数値演算を目的とするプログラム言語としては、たとえば LISP, COM-MIT, SNOBOL などがある。

8 を法とする演算式の解析

1) 辞書

Basic Symbols

〔メタ言語〕〔文法情報〕〔見出し〕

<XA> := <英字> := A | B | C | ..... | X | Y | Z

<X9> := <数字> := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

<B\*> := <乗除記号> := \* | / | ..... → [接続情報] 1

<X+> := <加減記号> := + | -

<X (> := <左かっこ> := (

<X )> := <右かっこ> := )

<CEQ> := <等号> := = ..... → [接続情報] 2

<=END> := <文末記号> := ;

<ⓈSPACE> := <空白> := Ⓢ

2) 接続規則

1 <英字> | <数字> | <右かっこ>

2 <英字>

3) 文法規則

Expression (句の文法規則)

<XN> # := <Number> := <数字>

<XV> .. := <Variable> := <英字>

<XV><CEQ> := <英字><等号>

<XTERM> := <Term> := <Number> | <Variable> |  
<左かっこ><Expression><右かっこ>

<XMUL> := <Multiplying Factor>  
:= <Term> | <Multiplying Factor><乗除記号><Term>

<XEXPR> := <Expression>  
:= <Multiplying Factor> | <加減記号><Expression> |

<Expression><加減記号><Multiplying Factor>

<XASSIGN> := <Assignment Statement>  
:= <Variable><等号><Expression>

Statement (文の文法規則)

<STATEMENT> := <Basic Statement>

:= <Assignment Statement><文末記号>

4) RULE-FILE

RULE-ID	R-X	REWRT-META	RULE-META
NUMBER	1	XN	X9
VARI 1	1	XV	XA
VARI 2	1	XV	XA CEQ
TERM 1	1	XTERM	XN
TERM 2	1	XTERM	XV
TERM 3	3	XTERM	X ( XEXPR X)
MULT 1	1	XMUL	XTERM



法規則によって operationを行ない、この読解の難易度がはかられる。そこから漢字の読解に果す役割もさぐれるだろうし、個々の漢字の基本度も測定できよう。

このプログラム自身にはまだその機能がないが、もし未登録の語の処理法を付加しえたとすれば——これはゆくゆくは例えば類推というようなことを含めて処理手順のなかに入れてゆくことが考えられる。また、そのためには、類推のアリゴリズムを確立させる方法の研究も必要である——、それによって一定範囲の語のもつ効率の測定もできるようになる。「基本的な用語」についてもたとえばこの種の接近法が考えられる。したがって、使用度数とは別な見方による、用語や用字の基本性への計量的な接近が可能になる。

AUTOSEG 1はわかち書きしていないデータについてわかち書きを行ない、辞書から語についての文法情報を入手する役目をもつ。もし原文がわかち書きしてあって、別に、単に辞書から語についての文法情報を転記するプログラムを作るならば、わかち書きをしたかな文を漢字かなまじり文に改めることに使える。もちろん、わかち書きをしていない原文を、漢字かなまじり文でもかな文でも、他の文字構成にコンバートすることが可能である。漢字かなまじり文のある標準的な表記（たとえば当用漢字現代かなづかい、あるいは読売式表記法）に改めることも可能である。ただしこのようなばあいには、終わりに別な小さいプログラムを付加する必要がある。これは、辞書の文字情報の部分だけをひろいあげて並べればよいというだけのものである。

以上述べたことを次のようにまとめることができる。それぞれの目的に応じてプログラムの方式や内容を発展させてゆくことにより種々の応用が可能である。日本語の用語・用字・表記などの研究のためにも、構文解析を伴った研究が有益であり必要である。また、自然語ばかりではなく言語として共通の問題をもつ人工語の処理などでも構文解析がその基礎的な手法として重要である。

#### <参考文献>

- 1) 石綿敏雄：「構文解析自動化の研究 I」 本報告書所収
- 2) N. Chomsky : Syntactic Structure. (1957) 勇 康雄訳『文法の構造』
- 3) 西村恕彦：「機械翻訳システムについての予報」 (1968) 電気試験所彙報

- 4) 森口繁一編：ALGOL入門（1962）
- 5) John A. N. Lee：The Anatomy of a Compiler（1967）
- 6) Hons Breuer：Dictionary for Computer Languages（1966）
- 7) A. Naur：“Revised Report on the Algorithmic Languages ALGOL 60”  
（Numerische Mathematik 4, 420（1963）and Cominication of ACM 6,  
No. 1（1963））（6）の巻末にも取められている。
- 8) 石綿敏雄，斎藤秀紀，木村繁：「単語認定プログラム」情報処理学会委員会資料  
69-4