

卒業論文

天気予報における知識表現と
言語表現との相互変換

昭和57年 3月1日提出

指導教官 藤崎 博也 教授
広瀬 啓吉 助教授

電子工学科 0437

亀田 弘之

内容梗概

本研究では、日本語文に対してコンピュータにより、文理解及び文生成の処理を行なった。記述の対象を天気予報に絞り、ATN文法により解析し、知識表現と言語表現との相互変換を行なうという形で試みた。

知識表現を得る際、推論が重要であること、文生成の際、機械的方法により自然な文を作るのがかなり困難であることや二種類の文簡約化が存在すること、更には、文解析・文生成において曖昧性の処理が極めて重要であることが明らかとなった。

目次

第1章 序論	1
第2章 自然言語処理に関する従来の研究とその問題点	2
§ 2-1 従来の自然言語処理(文解析・文生成)	
§ 2-2 従来の知識表現	
§ 2-3 問題点の指摘	
第3章 天気予報文の構造と知識表現	4
§ 3-1 天気予報文の構造	
§ 3-2 天気予報文の文法	
§ 3-2-1 文法作成手順の概略	
§ 3-2-2 天気予報文の用語の分類	
§ 3-2-3 幾つかの構文素の説明	
§ 3-3 天気予報の知識表現	
§ 3-3-1 知識表現	
§ 3-3-2 マトリックス形式とリスト形式との比較	
§ 3-4 実際に用いた知識表現の形式	
§ 3-4-1 実際に用いた知識表現形式の概略	
§ 3-4-2 天気文における“所により”の処理について	
第4章 天気予報文の解析	16
§ 4-1 ATN文法	
§ 4-2 天気予報文解析のためのアルゴリズム	
§ 4-2-1 アルゴリズムの概要	
§ 4-2-2 アルゴリズムの各部の概説	
§ 4-2-3 簡単な動作例	
第5章 天気予報文の生成	21
§ 5-1 知識表現をそのまま文変化する方方法	
§ 5-2 詠点をともなう文生成	
§ 5-2-1 詠点, “ひ”及び“ですか”の比較	
§ 5-2-2 具体的考察	
§ 5-2-3 詠点と“ひ”を挿入するアルゴリズム	

§ 5-3	簡約化をもとなう文生成	
§ 5-3-1	文生成の為の知識表現簡約化の仕方に関する考察	
§ 5-3-1-(i)	一つの例	
§ 5-3-1-(ii)	知識表現簡約化の方法	
§ 5-3-2	文生成の為の文表現簡約化	
§ 5-4	実行結果	
第6章	考察・検討	27
第7章	結 論	30
	謝 辞	30
	参考文献	31
	付 録	32
付録1	予報用語としての“のち”, “一時”, “時々”の意味内容	
付録2	実際に用いた知識表現の書式	
付録3-1 ~ 3-14	総合的動作例	
付録4	文生成関数BSFN1とBSFN2の出力例	
付録5	GFN1とREDUCEの動作例	
付録6-1 ~ 6-14	天気予報文解析及び生成用プログラムのリスト	
付録7	天気予報文解析関数PFの動作の流れ図	
付録8-1 ~ 8-23	風節解析関数KKKの動作の流れ図	

第1章 序論

自然言語は、(1)知識・概念・思想の表現・伝達・蓄積 (2)明確な推論・思考 (3)人間相互間の意思疎通等を人間が行う為の有用な手段である。情報化がますます進んでいる現代社会においては、人間とコンピュータとの自然言語によるコミュニケーションや機械翻訳システム等の自然言語処理の実現が切実に望まれるようになってきている。特に最近はVLSIやマイクロプロセッサ等のICの発達や素子の価格低下と相まって、このような自然言語処理システムを(特にOA関係で)作るようとする社会的気運が高まりつつある。今後、自然言語処理は更にその重要性をますますもたえられよう。

自然言語処理が行なわれ始めたのはコンピュータの黎明期からであるが、初期の頃は、自然言語処理における統語規則や意味を十分とり扱うだけの学問的基盤が未確立であったために、また、コンピュータ言語が未発達であったために、加えて自動翻訳の研究などは一時挫折をしてしまうこととなった。しかし、一方では、取り扱う対象の世界を限るなどの工夫を凝らすなどして、自然言語処理は、著実に成果を上げて行き、1977年には、架空のロボットが人の命令に従い積木を動かすという積木の世界のシステム¹⁾が作られた。今日では、自然言語処理は、QAシステムやデータベースシステムの分野で更に研究され続けており、パーズンカの際の文法としては、文脈自由文法、ATN文法²⁾、格文法³⁾が盛んに用いられている。また、内部情報を記述する方式としては、データベース分野ではCoddの関係理論⁴⁾が近年注目されており、人工知能分野では、フレーム理論等が研究されている。

このように、自然言語処理は盛んに研究されているが、従来の研究では、文解析・文生成の一方のみを扱うか、もしくは、入力文の解析は綿密に行なうか、出力文は入力文に対する単なる応答文にすぎない、というやや文解析寄りのものが多い。

そこで、本論文では、対象とする世界を天気予報の世界に限るか、文解析・文生成の両者をもとに重視し、ともに綿密に行なう総合的な自然言語(日本語)処理システムの実現を目標とし、その試作を行なう。

第2章 自然言語処理に関する従来の研究とその問題点

言語表現と知識表現との相互変換を行なうためには、自然言語処理(文解析・文生成)と知識表現が必要である。この章では、これらの分野でのようなことが既に行なわれてきたのかを概観し、問題点を指摘する。

§2-1 従来の自然言語処理(文解析・文生成)

初期の自然言語処理は、単語と単語を置き換える方式の翻訳であったが、これは統語規則を無視していたため、行きつまた。1956年頃にN. Chomskyにより句構造文法概念が確立され、文脈自由型文法の考えが広まると、人々は、文脈自由文法を用いて、コンピュータによる文の構文解析や翻訳を試みるようになった。この方法は、人工言語の構文解析や人工言語間の翻訳には有効であったが、統語のみに着目し意味の面を取り扱わなかったために、曖昧性等の言語現象を処理しきれず自然言語処理に対しては、なおも、不十分であった。この間、人工知能のQAシステム開発では、対象となる世界をうまく制限するなどの工夫を凝らし、意味上の曖昧性が生じぬようにしていたので、着々と進歩をとめていた。

自然言語処理のバックボーンである文法も着実に進歩し、種々提案されている。

その一つは、述語中心に考える格文法がある。この文法では、述語の動作主となるもの、述語の動作の及ぶ対象などを考える。このように、構文的情報以外に意味的な要素が非常に入っている文法であり、ロシア語や日本語のような比較的語順の緩やかな言語に対して有効である。

一方、英語のような語順の厳しい言語に対しては、ATN文法が有効である。ATNは、特に意味を考慮した文法ではないが、弧通過の際の操作として特徴量や性・数・格等をチェックすることにより或る程度まで意味的なチェックは可能である。ATNは、文法自体が効率の良いパーサーであること、内部動作を利用すると内部表現が作り易いことから、いくつかのQAシステムで利用された。

また、LINGOLもATNと同じようにChomsky標準形の文法を基礎とした文解析システムである。LINGOLは、書き換え規則という理解しやすい表現形式を保ちながら、それでは表現できない各種の制約条件はプログラムによってきめ細かく実行しようとする所に特徴がある。ATNは文法記述の方法が手続き的であるのに対し、LINGOLでは、宣言的である。

このような種々の文法により、今日、自然言語処理を行なおうとしている。

§2-2 従来の知識表現

ここで言う知識表現は、自然言語入出力の際に使われるもののみの意味とし、ロボットなどで用いられている知識表現は考えないものとする。

自然言語入出力用の知識表現は、まず、QAシステム BASEBALL (1960年) で用いられた。このシステムでの内部表現は書式化されており、外部からの質問に対し、テーブルサーチやパターンマッチングにより返答文を見い出していたにすぎなかった。

1960年代には、推論機能も用いられはじめ、1965年には、Semantic Memory という連想検索方式により次々と索引をたどれるシステムが作られ、一般的な知識を用いるQAシステムが可能になりはじめた。

1970年前後には、いくつかの対話システムが作られた。ロケットにより、月から持ち帰られた各種資料標本の解析データに関する問い合わせシステムがその一つで、自然言語によってデータベースへアクセスするという方法の最初の試みであった。その他のシステムとしては、Winogradの楯本の世界のシステムがある。このシステムは、知識表現として意味ネットワークを採用していた。

一方、QAとは別に、近年データベースの分野で知識構造が盛んに研究されており、Coddの関係モデルが注目を集めている。関係モデル理論では、内部データを表形式でしまっておくのだが、その際、できるだけ基本的な表の形に変形分解しておこうとするのである。

また、人工知能の分野では、フレーム理論、意味ネットワークが研究されており、より汎用的な知識表現を得ることを目標にしている。

§2-3 問題点の指摘

§2-1, §2-2 で概観したように、文解析・文生成の分野では、強力なパーサや豊富な表現力を持つ文法を見い出すことに重点がおかれ、知識表現の分野では、汎用的かつメモリを浪費しない知識表現を得ることに重点が置かれ、両者は分離的である。一方QAシステムでも、文解析・文生成ともにバランスよく深く掘りさか研究しているものは少ない。このように、知識表現をその仲立ちとした文解析・文生成は、未だ十分にお互いが密接につながっていないのが現状である。そこで、知識表現を仲立ちとした文解析・文生成を本論文では取り上げる。

第3章 天気予報文の構造と知識表現

113
130

天気予報文とは、通常、気象状況の予測を述べた文のことを言う。高・低気圧の位置や台風の進路等の概況も述べられることがあるが、本卒業論文では、各地の天気を述べた文に限ることにし、その構造とそこに述べられている情報を表記する為の知識表現を以下に述べる。

§3-1 天気予報文の構造

天気予報文(各地の天気)では、(i)地域名(例東京都、千葉県)(ii)時間帯(例今日は、あすは)(iii)風況(風向と強度)(iv)天候状況(例晴れ、雨)がこの順に述べられているのが普通である。~~この他~~、東京都心部等の(v)降雨確率(もしくは、降雪確率)や(vi)気温(vii)湿度も述べられることもある。このうち(i)~(iv)は、ほとんど必ず述べられていなければならない主要な情報である。このような天気文としては、例えば、『東京都、今夜は北東の風で晴れでしょう。あすは北東の風のち南の風はれのすくもり、おそくなって所により俄雨が降るでしょう。今日午後3時から9時までの間に都心で雨の降る確率は、5%未満の予想です。』~~がある~~の類である。

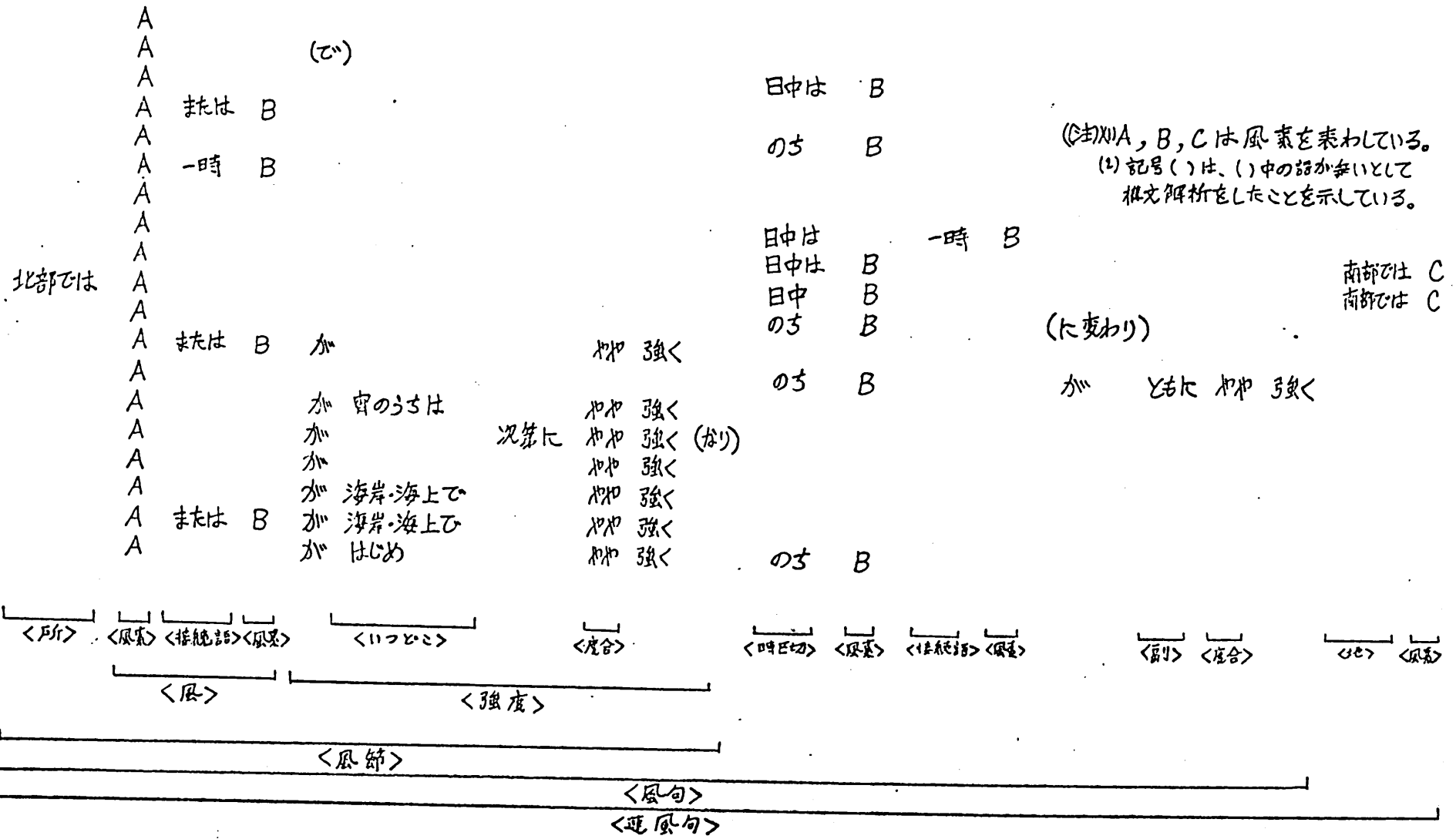
このような天気予報文においては、(iii)と(iv)の構造がやや複雑である。次頁に(iii)と(iv)の構文解析例が載せてある。これらの図は、ほとんどの場合を含んでいるが、『東京都、明日は、晴れのち曇り、おそくなって雨の降る所があるでしょう。』の類(これは、『東京都、明日は、晴れのち曇り、おそくなって所により雨が降るでしょう。』とすれば取り扱える。)や、『千葉県、今夜は、曇りで、明け方冷え込むでしょう。』とい、た天候状況+気温のようなものは扱えない。しかし、この種の文は、少数例なので、今回は、除いて考えることにする。また、各地の天気予報といっても、テレビ、ラジオ、新聞、電話等で、幾分異なっているのので、テレビ(NHK第一放送)のものを取り扱うことにする。

220 30
33
600

§3-2 天気予報文の文法

§3-2-1 文法作成手順の概略

天気文の解析図(図3-1)、風文の解析図(図3-2)から、また、表3-1のような文脈自由型の文法を作ってみた。次いで、この文法を遷移ネットワークの形式に書き直した(図3-3)。プログラムを作るには、遷移ネットワークが便利であり、かつ、遷移ネットワークは文脈自由文法の書き換え規



(注)A, B, C は風象を表わしている。
 (2) 記号 () は、() 中の語が無いとして
 構文解析をしたことを示している。

南部では C
 南部では C

(に変わり)

か とおた やや 強く

図3-1. 風文の構文解析例

	A								
	A	のち	B						
	A	のち	B	で	秋分地方では				
朝のうち	A	のち	B						
	A	ですか	日中は	B					
	A	のち	B	ですか	南部では				
朝のうち	A	ですか	昼前から	B					
朝のうち	A	ですか	まもなく	B					
	A	で	昼前から	B					
	A	で	昼頃から	B					
	A			で	北部山谷1)では	初めのうち			C
	A	で	日中は	B					
	A	で			北部山谷1)では				C
	A	のち	B	で		おそくには	所により		C
	A	のち	B			おそくなつて	所により		C
9	A	のち	B	で		おそくなつて	所により		C
	A	のち	B	で			所により		C
	A	のち	B	で		おそくなつて			
	A	のち	B			夜は			
	A	のち	B	ですか		夜は	所により		C
	A	のち	B	ですか		おそくなつて	所により		C
	A	で			北部山谷1)は		所により		B°
9	A	ですか			北部山谷1)は		所により		B° (B)
	A	で			北部山谷1)では				B°
	A	で	午後	B					
9	A	で	午後は	B					
	A	で					所により		B°
	A	ですか				宵のうち			B
	A	ですか				宵のうち			B

(注) ① A, B, C は天候素もしくは天候塊を表わしている
 ② A, B, C は第二天候素を表わしている

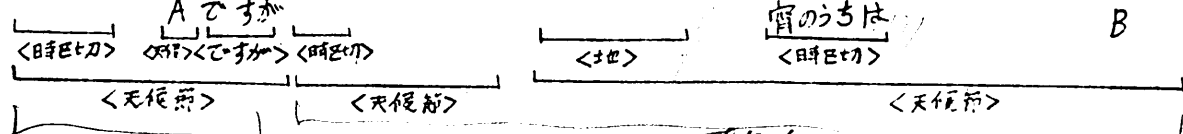


図3-2. 天候文の構文解析例

表3-1. 天気予報文の文法

(1) 天候文全体の枠組

<天気予報文> —————> <どこで> <いつ> <どうた>

<どこで> —————> <都県名語>

<いつ> —————> <時間帯語>

<どうた> —————> <風文> <天候文> <終了語>

<都府県名語> —————> 東京都 | 千葉県 | 埼玉県 | 神奈川県 | 栃木県 |
道 茨城県 | 群馬県 ...

<時間帯語> —————> きょうは | あすは | あさっ ては | こんやは
(以上 4 語)

<終了語> —————> でしょう | となるでしょう | の見込みです...

((注)) i) <***> は非終端記号を、“|”は“又は”を表わしている。

ii) A|B|C... の“...”は、“等々”と読み、終端記号が未だ書き及ばれていないことを示している。この記号の付いていないものは、全ての終端記号が書き出されている。

表3-1. 天気予報文の文法

(2) 風文の文法

<風文> → <連風句>

<連風句> → <風句> | <風句> <風句>

<風句> → <風節> | <風節> <風節>

<風節> → <所> <時区切・風> <風> <強度>

<所> → 北部では | 南部では | NULL

<時区切・風> → 日中は | のち | NULL

<風> → <風素> | <風塊>

<風素> → <方角> の風

<方角> → 東 | 西 | 南 | 北 | 東寄り | 西寄り | 南寄り | 北寄り | 北東 |
南東 | 南西 | 北西

<風塊> → <風素> <接続> <風素> | <接続> <風素>

<接続> → または | 一時

<強度> → 加 | いくほど | <副> <度合> 強 | NULL

<いくほど> → 宵のうちには | 海岸・海上では | じめ | NULL

<副> → とともに | 次第に | NULL

<度合> → やや | NULL

表3-1. 天気予報文の文法

(3) 天候文の文法

<天候文> → <連天候句>

<連天候句> → <天候句> | <天候句> <天候句>

<天候句> → <天候節*天候節> <天候節>

<天候節> → <地> <時区切・天> <天候> <ひすか>

<地> → 北部山沿いでは | 秩父地方では |
南部では | NULL

<時区切・天> → 朝のうち | 昼前か | 昼頃か | 午後 | 午後は | 夜は | 日中出
おそくなって | 宵のうち | 宵のうちは | (のち) | NULL

<天候> → <天候素> | <天候塊>

<ひすか> → ひひすか | NULL

<天候塊> → <天候素1> <接・天> <天候素1> |
<接・天> <天候素1>

<天候素> → <天候素1> | <天候素2>

<天候素1> → 晴れ | <もり雨雪>

<天候素2> → 候雨 | 候雪

<接・天> → 時々 | 一時 | か

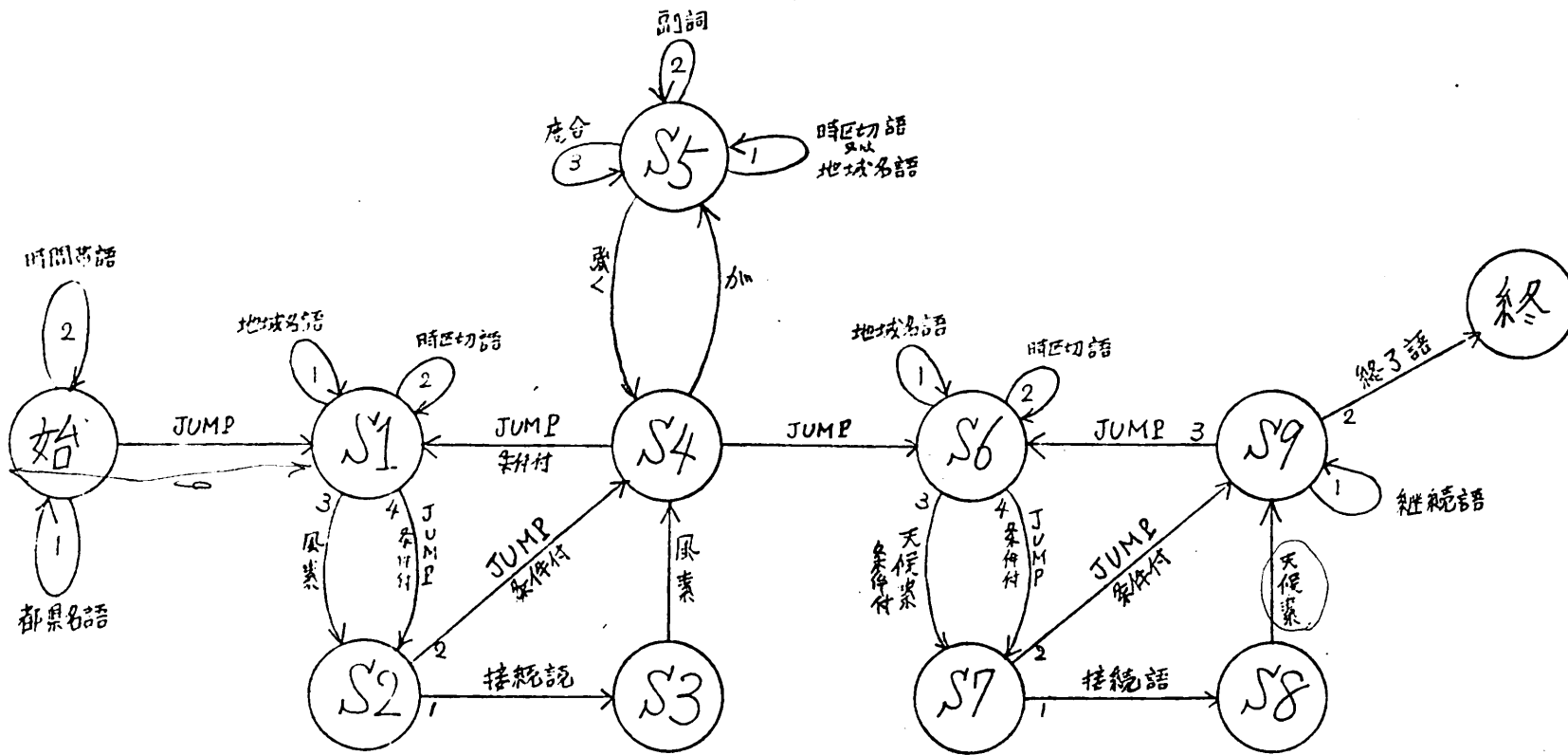


図3-3. 天気予報文の文法のネットワーク表現

則かゝ容易に作れるのでこのように書き直した。

この際、文脈自由型文法は、多種の天気予報文をその中に含み込むものを作り、これを書き直したネットワークで多少の書き直しを行ない、より強力化した。(注)ネットワークをプログラム化する際、拡張のための付加条件を加えより強化させ、所謂A T N文法に仕上げた。)

§ 3-2-2 天気予報文の用語の分類

天気予報の各地の天気文に現われる語は、以下のように分類される。

- (a) 時間帯語(JT) ----- 今日は、あすは、あさっては、今夜は
- (b) 都府県名語(TM) ----- 東京都、千葉県、群馬県 など
- (c) 地域名語(CM) ----- 北部では、南部では、北部山沿いでは、北部県境では、所により、海岸・海上で、(全域で)、(一般地では) など
- (d) 時区切語(JG) ----- のち、朝のうち、昼前から、午後は、日中は おそく、なつて、宵のうち など
- (e) 風素語(KS) ----- 北の風、南の風、東の風、西の風、北寄りの風、南寄りの風、東寄りの風、西寄りの風、北東の風、南東の風、南西の風、北西の風
(以上12語のみ)
- (f) 天候素語(TS) ----- 晴れ、雨、曇り、雪、曇雪、曇雨
- (g) 継続語(KZ) ----- で、ですか
- (h) 副詞(FS) ----- ときに、次第に
- (i) 度合語(DA) ----- やや
- (j) 接続語(SZ) ----- 時々、一時、または、か
- (k) 終了語(SR) ----- でしょう、となるでしょう、の見込みです など
- (l) 区切り記号(KG) ----- 、,、。

(注) 上記のJT, TM, CM等は、それぞれ時間帯語、都府県名語、地域名語等の省略記号である。

§3-2-3 幾つかの構文素と記号の説明

§3-2-3-(i) 風素, 風塊と天候素, 天候塊について

風素, 天候素, 風塊, 天候塊とは、それぞれの例を下記に示すと

風素	-----	北の風, 南寄りの風	など
天候素	----	晴れ, 雨, 曇り	など
風塊	-----	北の風-時南の風	など
天候塊	----	晴れ時々曇り	など

であり、“素”とは一つの基本的な天気状況を表わしており、“塊”とはそれら幾つかの“素”から成る構文構造的には複合的な天気状況を指している。

ここで注意すべき点は、天候塊も風塊も、意味的には単一の天気状況を表わしていることである。例えば、“曇り-時雨”も一つの天気状況と考えるのである。確かに、瞬時瞬時の天気は、“曇り”か“雨”のどちらかである筈だが、予報であることを考えると、“その時間帯勢は大体は曇っているが、一時雨が降る天気”と解釈できる。付録の図A-1, 表A-1からも、この解釈が、気象庁の定義解釈と合致していることから、妥当であると言える。

§3-2-3-(ii) 非終端記号<天候素1>と<天候素2>について

基本的な天気状況を表わす語としての天候素は、雨, 晴れ, 曇り, 雪etc. があるが、曇りや曇りも天候素として扱うと便利である。しかし、曇りや曇りは、天気状態のみならず、その降り方(時間的な)までをも意味内容に含んでいる。その為[□]あるは、はれですか、山浴いではくもりで曇りか曇りがあるでしょう。□という言い回しが可能になってしまう。そこで、雨, 雪等を第一天候素(記号<天候素1>), 曇り, 曇り等を第二天候素(記号<天候素2>)として一応区別した。

§3-2-3(III) “NULL”と“JUMP”について

記号NULLは、表3-1中に用いられているが、これはフランクを意味しており、『<P> → α|NULL』から、非終端記号<P>をαで置き換えるか、もしくは、フランクで置き換えてもよいことを表わしている。

また、文法を遷移ネットワークの形式で表わした時、NULLはJUMPに相当する。従って、JUMPの記された^{遷移}上は、無条件に通過できることを意味している。但し、この際、解析対象となっている語(文字列)は、弧通過後も同一の語(文字列)とし、次の語に制御を進めないとする。

(註)フランクとは、空列のことを言う。

§3-3 天気予報の知識表現

§3-3-1 知識表現

本論文では、天気予報の情報が時間と場所の関数であることに着目して、マトリックス表記法を知識表現として用いた。この表記法を用いれば、例えば、『晴れのち一時曇り、山浴いではおそくなって雨が雪が降るでしょう。』は、表3-2のようになる。

時間 地域	はじめのうち	のち	おそくなって
一般地	晴れ	晴れ一時曇り	晴れ一時曇り
山浴い	晴れ	晴れ一時曇り	雨か雪

図3-2. マトリックス表記法による知識表現

実際のコンピュータ内では、このマトリックスそのものではなく、図3-5のようなリスト形式を用いた。それは、プログラム言語としてLISPを用いたので、リスト形式の方が自然だからである。

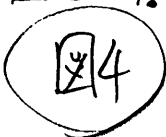
([(はじめのうち) { (一般地 晴れ) (山浴い 晴れ) }]
 [(のち) { (一般地 晴れ一時曇り) (山浴い 晴れ一時曇り) }]
 [(おそくなって) { (一般地 晴れ一時曇り) (山浴い 雨か雪) }])

(a) リスト形式 (地域に着目して纏めたもの)

([(一般地) { (はじめのうち 晴れ) (のち 晴れ一時曇り) (おそくなって 晴れ一時曇り) }]
 [(山浴い) { (はじめのうち 晴れ) (のち 晴れ一時曇り) (おそくなって 雨か雪) }])

(b) リスト形式 (時間に着目して纏めたもの)

図3-5. リスト形式の知識表現 (内容は表3-2と同一)



§3-3-2 マトリックス形式とリスト形式との比較

リスト形式は、マトリックス形式を一次元化させたものと着せられるが、両者には、以下のような長所・短所がある。

- (a) マトリックス形式では、時間と地域とを同等に扱えるが、リスト形式では、同等さを欠く。(cf. 表3-2, 図3-4)
- (b) マトリックス形式は見て分かりやすい。
- (c) リスト形式は、マトリックス形式よりも拡張性に富んでいる。

(a) についてももう少し説明を加える。例えば「晴れのち曇り、山浴いではおそくなって雨が強く降るでしょう」をリスト形式とマトリックス形式で表わすと下図3-5のようになる。

時間 地域	はじめ	のち	おそくなって
一般地	晴れ	曇り	曇り
山浴い	晴れ	曇り	雨 強い

(a) マトリックス形式

- ([(一般地) { (はじめ晴れ) (のち曇り) (おそくなって 曇り) }])
 - ([(山浴い) { (はじめ晴れ) (のち曇り) (おそくなって 雨 強い) }])
- (b) 但し、[(地域名) { (時 天気 強度) ... }] とする)

(b) リスト形式

5 図3-5 マトリックス形式とリスト形式の比較

この図から、この例文に対しては、マトリックス形式では、欄(次元)を新たに一つ追加せねばならず、かつ、この追加欄はほとんどが空白となるが、一方、リスト形式では、リストの一部を変更するだけで無駄な空白部を作らず、~~マトリックス形式のデメリット~~に済み実用的である、と言える。

一般に、時間と地域とを指定すると、天気情報が一表に定まるとはいっても、天気情報には天気状況、風向、風の強度、気温、湿度等色々ある。従って、一つの時間一つの場所に対して一つの情報を表記するには有効だが、多数の情報を表わすには不適切なマトリックス形式よりは、どの位置に何の情報を記かさえ決めておけば伸縮自在のリスト形式の方が、天気情報の知識表現形式としてはすぐれているといえよう。

§3-4 実際に用いた知識表現の形式

§3-4-1 実際に用いた知識表現形式の概略

天気予報は、一般的傾向として、“ひとび、いつ、どうだ”の順に情報か述べられ、更に、情報は時間の流れに沿って述べられることが多い。本論文ではこの点に着目し、風・天候ともに、下図のような形式で知識を表現することにする。

[((ひとび) ((いつ どうだ) (いつ どうだ) ---- (いつ どうだ)))
((ひとび) ((いつ どうだ) (いつ どうだ) ---- (いつ どうだ)))

((ひとび) ((いつ どうだ) (いつ どうだ) ---- (いつ どうだ)))]

6 図 ~~3-4-1~~ 実際に用いた知識表現形式

上図の“どうだ”の部分には、風の知識表現では、風向と強度の度合の情報を、天候の知識表現では、天候状況のみを記すこととする。実際に用いた知識表現の詳細は、付録の図A-2に述べてある。

§3-4-2 天候文における“所により”の処理について

上記の知識表現では、“ひとび”に着目して情報を整理しているが、その際、“所により”を一つの地域として考えるのか否かか問題となる。『おそくなつて所により雨』と言った時、“所により”という特殊地域で雨が降ると考えるよりは、寧ろ、“所により雨”という天候状況であると看做した方が妥当であると考え、本論文では後者の考えを採用した。なせなら、“一時雨”を或る有限時間帯内での一つの天候状況と考えたのだから、“所により雨”も或る有限地域内での一つの天候状況と考える方が、統一的であるからである。

しかし、“所により”はやはり場所を表わす語であり、かつ、知識表現は“ひとび”をインデックス風に考えていることから、下の例のように表わすこととする。

例1: ([北部では] {おそくなつて (所により (雨))})]]

例2: ([北部では] {おそくなつて (所により (曇り-時雨))})]]

上記の例で、(所により (雨)) という風に天候状況部分をリストにしているのは、“所により”と“天候状況”とが情報として異質であるという情報を保つ為である。

第4章 天気予報文の解析

文から知識表現を得るには、構文解析が必要である。本章では、前章の文法を用いて知識表現を得るアルゴリズムの概要を述べる。また、最終的な文法として用いたATN(拡大遷移ネットワーク)についても簡単に述べる

§4-1 ATN文法

ATNは、遷移ネットワークを強化したもので、入力文字列を頭から順に解析をする。見かけは遷移ネットワークと同じであるが、弧通過の際に終わりの所定の操作を行なう点が異なる。図4-1はATNの例であり、このATNは、“A pretty girl was chasing a butterfly.”のような基本的な英文を受理・解析する。

ATN文法の長所としては、

- (a) 入力文字列を頭から順次解析する。
- (b) 構文解析の動作と内部表現作りの動作がある程度独立している。
- (c) 心理学の実験結果をうまく説明する。
- (d) 見て分かりやすい
- (e) プログラム化しやすい

が、短所としては

- (a) 広い分野の文を受理しようとする、弧の本数が増えすぎて、かえって直観的理解が妨げられる。
- (b) 英語のように語順に強い制限のある自然言語には有用だが、語順の規則の緩やかな言語に対してはあまり有効ではない。

などがあげられる。

本論をとり扱う天気予報は、時間の流れに沿って、天気の変化に沿って情報が述べられるという、傾向があるので、ATN文法で記述するのが、妥当と考えられる。

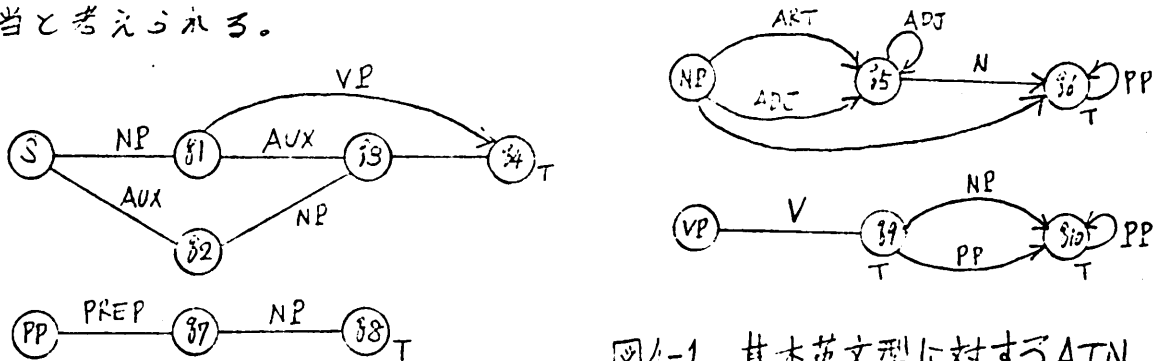


図4-1. 基本英文型に対するATN

(T: その状態の位置で終ってよいことを示している)

§4-2 天気予報文解析の為のアルゴリズム

§4-2-1 アルゴリズムの概要

天気予報文から情報を抽出し、その知識表現を作る過程は、図4-2のようになっている。このアルゴリズムの特徴は、(1)各節ごとに順次解析を行なう、(2)各節ごとに順次知識表現を作成する。その際、(3)前後の節の情報も利用してその節の知識表現を作る、である。この最後の(3)が可能なため、この文解析アルゴリズムは極めて強力なものとなっている。

§4-2-2 アルゴリズムの各部の概説

図4-2のアルゴリズムは、~~構文解析~~、構文解析(パーズンク)部分、一次記憶(短期記憶)作成部分、二次記憶(長期記憶、知識表現)作成部分の三つの部分から成っている。各部の役割り・動作は以下のようになっている。

ア (a)構文解析部分 ----- この部分のみが文法にかかわっているだけである。このパーズンクの動作例を後に述べる。

イ (b)一次記憶の作成 ----- 或る節のパーズンクが終了した後、その節から得た情報・知識をできる範囲で纏める。ここに、得られた情報とは、例えば、この節には天気状況が述べられていない、という情報も含めている。このように、情報が欠落しているという情報をも利用することにより、知識表現作成が楽になった。

ウ (c)知識表現の作成 ----- 前後一つ一つの節の情報を利用し、簡単な(常識的な)推論により未知の部分をつまみ、完全な知識表現を作る。

エ (d)知識表現の統合 ----- 知識構造を構成する各部分を一つにまとめる。

(注) 本論文においては、知識構造を構成する各部分を作成する際に、簡単な推論を行なっているが、それらの構成部分素を統合して最終的な知識表現を作る際には、何の推論も行なっていない。また、(c)と(d)の部分で、曖昧性が問題となるが、今は一定の解釈しか行なわれないようになっている。

この図をのせよ 頭

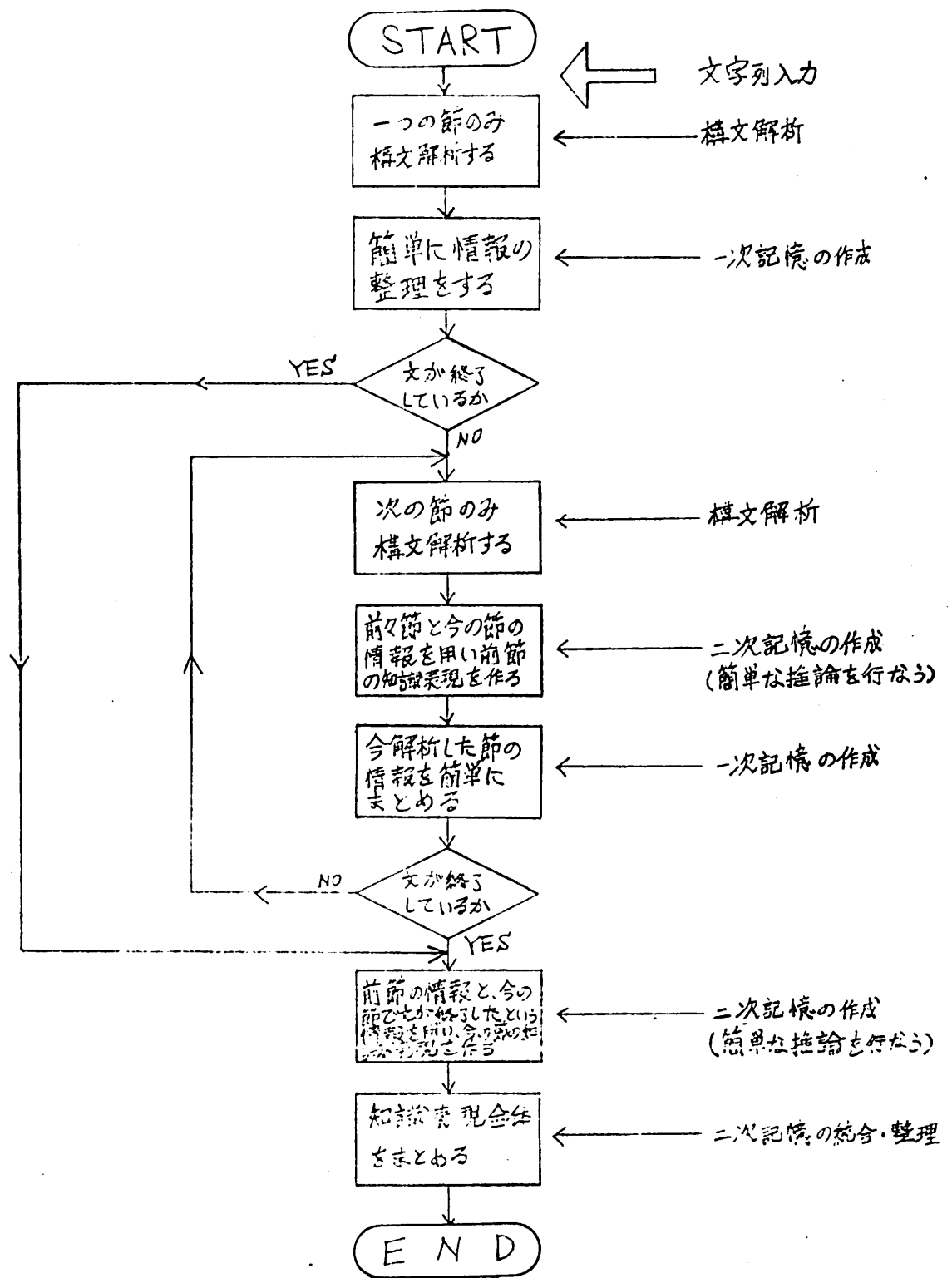


図4-2. 言語表現から知識表現を得る手順

§ 4-2-3 簡単な動作例

前節、前々節においてアルゴリズムの概略を述べたが、アルゴリズムの動作の概要を知るには、具体例にあたってみるのが最もわかり良いので、ここでは一つの簡単な例文に対するアルゴリズムの動作をざっと追ってみることにする。ここに、記号*は、解析対象となっている語が代入してある変数、天気素-1stは、その節中に現われた第一番目の天気素、天気素-2NDは、二番目に現われた天気素をその値として持つ変数名である。

以下、動作を追ってゆく。

入力例文『はれ のち 一時 くもり でしょう。』

- 1) * := はれ (変数*に“はれ”を代入する)
- 2) *は地域語ではない (地域語の処理の通過に失敗)
- 3) *は時区切語ではない
- 4) *は天気素である
- 5) 天気素-1st := *
- 6) * := のち (次の語の解析を始める)
- 7) *は接続語ではない
(ここで、天気状況は“はれ”なることがわかった)
- 8) *は継続語ではない
- 9) *は終了語でもない
(ここで、一つの節が終了したこと、及び、また続きのあることがわかった)

以上が、第一番目の節のパージンの部分である。続いて、この節を簡単に纏めて、

{ 地域名語 : なし
 時区切語 : なし
 天気状況 : はれ

なる一次記憶を得る。

ここで、更に前節の知識表現を作るのだが、今の場合、この節が第一番目の節であり、前節は存在しないので、何もせず次の節のパージンクに移る。

- 10) *は地域名語ではない (* = のちであることに注意)
- 11) *は時区切語である
- 12) 時区切語 := * (変数“時区切語”に“のち”を代入)

- 13) * := 一時 (解析を進める)
- 14) * は天候素ではない (天候素が見つけ出せなかつたか、第二節以降の節であるから、“時々何々”という可能性が未だ残されているので、また解析をやめずに続ける。但し、天候素-1stは前節のものと同じであることにする。)
- 15) * は接続語である (この節の天候状況は、天候塊で表わされていることが予想される。)
- 16) 接続語 := *
- 17) * := くもり
- 18) * は天候素である
- 19) 天候素-2ND := *
- 20) * := でしょう
- 21) * は継続語ではない
- 22) * は終了語である (この節で文が終了することが判明)

以上が第二番目の節のパーシングである。

ここで前節の知識表現を作る。まず、前節の時区切語は、前節が第一番目の節であったこと(つまり、前々節が存在しないこと)、今の節の時区切語が“のち”であったことから、“はじめ”と推定できる。また、前節の地域名語は、前節が第一番目の節であったことから、“一般地”と推定できる。以上から、前節の知識表現は、

((一般地) ((はじめ (はれ))))

であることがわかった。

次に今の節の情報を簡単に纏めるのだが、今の場合この節で文が終了していることがわかっているので、知識表現を直ちに作ることもできる。

さて、今この節で不明なのは、天候素-1stと地域名であるか、両者とも前節のものと同じと考えられるので、

{ 地域名語: 一般地
 { 時区切語: のち
 { 天候状況: はれ一時くもり

となる。これからこの節の知識表現は

((一般地) ((のち ((はれ一時くもり))))))

となる。以上全体を纏めて、

((一般地) ((はじめ (はれ)) (のち ((はれ一時くもり))))))

なる知識表現を得る。これで文解析が終了する。

第5章 天気予報文の生成

QAシステムなどにおいては、内部の知識表現から、その知識表現の表わしている情報を正確に、かつ、わかりやすく伝達する文の生成が肝要である。そこで、この章では、天気予報の知識表現から、正確でわかりやすい天気予報をいかにして作つたよいかを論じる。

§5-1 知識表現をそのまま文章化する方法

本論文で採用した知識表現は、文の表層表現に大変近い。この点に着目するならば (1) 構文木の適当な所に“で”や“ですか”を挿入する。(風文では、“か”や“強く”を適時挿入する。) (2) 知識表現内部の不要なカッコを除去する、ことにより文らしきものが簡単に生成できる。

例えば、『はれ のち 一時 くもり おそくなって 所により 儼雨 でしょう』の知識表現

((初めのうち (はれ)) (のち ((はれ-一時くもり))) (おそくなって (所により (儼雨))))
に対して

(a) 単に内部のカッコを取り除くだけ

(初めのうち はれ のち はれ-一時くもり おそくなって 所により 儼雨)

(b) 節と節の合間に“で”を入れ、カッコを取り除き、“でしょう”を付ける

(初めのうち はれ で のち はれ-一時くもり で おそくなって 所により 儼雨 でしょう)

のようにして文が生成できる。

§5-2 読点をともなう文生成

知識表現の時点では何の曖昧性がなくても、生成された文に曖昧性が生じることがある。しかし、人間が読点を用いたりして幾分曖昧性を解消させているように、コンピュータによる文生成の際にも読点をうまく打たせれば曖昧性は幾分なりとも解消できるであろう。

§5-2-1 読点，“で”及び“ですか”の比較

次のような4つの文を考えてみる。

- | | | | | | | | | |
|---|---|-----|-----|----|--------|--------|------|---|
| { | ① | 東京都 | 今日は | 曇り | おそくなって | 所により | 雨 | |
| | ② | 東京都 | 今日は | 曇り | で | おそくなって | 所により | 雨 |
| | ③ | 東京都 | 今日は | 曇り | ですか | おそくなって | 所により | 雨 |
| | ④ | 東京都 | 今日は | 曇り | 、 | おそくなって | 所により | 雨 |

これらの文から、次のことがわかる。

- | | | |
|---|-----|--|
| { | (a) | “で”，“ですか”，“、”の順に文の切れ具合がきつくなる。 |
| | (b) | “で”，“、”は単純に文の切れを表わすが、“ですか”は、接続助詞“か”の効果により逆接の意味が加わっている。 |

以上の考察から、文生成の際、“で”と“、”とは、構文構造からだけでも挿入位置をある程度決定できるが、“ですか”は、前後の文脈的なつながりも知らねば挿入位置が決められぬことがわかる。そこで、本論文では、上記の考察と相田⁽⁶⁾氏、半田⁽⁷⁾氏の研究結果を踏えて文生成を行なうことにする。

§5-2-2 具体的な考察

まず、意味の区切れには必ず読点を付けることにすると、知識表現：

((初めのうち (はれ)) (のち (はれ-時々))) (おそくなって (所により) (俵雨)))
は、(初めのうち はれ、のち はれ-時々、おそくなって 所により 俵雨 でしょう)
となり、文がフツフツと切れていて寧ろ読みづらくなってしまっている。
これは、読点を不用意にたくさん付けてはならぬことを示している。

そこで、今度は、意味の切れ目には読点を付けるか、最後の意味の切れる部分には読点ではなく“で”を入れることにすると、先程の例文は、

(初めのうち はれ、のち はれ-時々) で おそくなって 所により 俵雨 でしょう)
となり、かなり自然な表現とな⁽⁸⁾前の方針よりますますよくなっている。この後者の方針を、読点，“で”の挿入方法の最終的なものとして採用することにする。

§ 5-2-3 読点と“で”を挿入するアルゴリズム

前節で述べた後者の方式、即ち、意味の切れ目に読点を必ず打つか、最後の意味の切れ目には“で”を入れる方式の動作を行なうアルゴリズムを簡単に説明する。

このようなアルゴリズムは次のようにして実現できる。

アルゴリズムの例:

```
if Length[Q] = 0 → ERROR MESSAGE
   = 1 → R1 でしよう
   = 2 → R1 で R2 でしよう
   = 3 → R1、R2 で R3 でしよう
   = 4 → R1、R2、R3 で R4 でしよう
```

(以下同様)

(ここに、 $Q \triangleq (R_1 R_2 R_3 \dots R_n)$ 、 R_i は例えば(のち(もり))といふもので、Length[X]は、リストXのリスト長を与える関数である。)

このアルゴリズムを再帰的関数の形で実現させたのが、本論文のプログラム中のBSFN2, BSF2である。(付録 図A-4. 参照)

§ 5-3 簡約化を伴う文生成

§ 5-1 や § 5-2 の方式での文生成では『はれ のち はれ時々くもり』といった冗長な文が生成されてしまう。単に文を生成するのではなく、天気予報文を生成することを目的としているのであるから『はれ のち時々くもり』の形の文を生成すべきである。本節ではその方法について述べる。

§ 5-3-1 文生成の為の知識表現簡約化の仕方に関する考察

§ 5-3-1-(i) 一つの例

風文『北の風のち南の風がともにやや強いでしょう』に対する知識表現は、図5-1のように書ける。この表もしくは、リストだけから文を単に生成

	はじめ	のち
風向	北	南
強度	やや	やや

すると、『はじめ 北の風が やや強く、 のち 南の風が やや強い でしょう。』となる。この知識表現から“ともに強く”の表現を作り出すには、この表現形式をながめているだけではためなのである。

(a) マトリックス形式

((はじめ北の風 やや)(のち南の風 やや))

(b) リスト形式

図5-1. 風文の知識表現

§ 5-3-1-(ii) 知識表現簡約化の方法

図5-1 のリストは、形式的に

$$((A \ B \ C) \ (A' \ B' \ C'))$$

と書ける。(ここに、A, A': 時区切語, B, B': 風向情報, C, C': 風の強度情報である。) ^併、同一時間帯の天気に関する命題(即ち、A=はじめのうち、B=北の風 C=やや強く の三つ)は、同時に成り立っていないことはなることから、(A B C)をA*B*Cと書くことに~~する~~。また、(A B C) と (A' B' C') は、A ≠ A' なる互いに異なる時間帯の命題と考えられるので ((A B C) (A' B' C')) = (A B C) + (A' B' C')

と書くことにすると、

$$((A \ B \ C) \ (A' \ B' \ C')) = A * B * C + A * B' * C'$$

と書き直せる。このように書くことにすれば、例えば、 $C = C'$ の時は

$$\begin{aligned} L &\triangleq ((A \ B \ C) \ (A' \ B' \ C)) \\ &= A * B * C + A * B' * C \\ &= (A * B + A * B') * C \quad (\because C = C') \\ &= (((A \ B) \ (A' \ B')) \ C) \end{aligned}$$

と変形できる。これを図5-1の例に当てはめると、

$$\begin{aligned} L &= ((\text{はじめ} \ \text{北の風} \ \text{やや}) \ (\text{のち} \ \text{南の風} \ \text{やや})) \\ &= (((\text{はじめ} \ \text{北の風}) \ (\text{のち} \ \text{南の風})) \ \text{やや}) \end{aligned}$$

となり、これは『はじめ北の風のち南の風がともにやや強く』を表わしていると考えられる。また、

$$L = ((\text{はじめ} \ \text{北の風}) \ (\text{のち} \ \text{北の風}))$$

となっている時は、

$$\begin{aligned} L &= \text{はじめ} * \text{北の風} + \text{のち} * \text{北の風} \\ &= (\text{はじめ} + \text{のち}) * \text{北の風} \\ &= \text{一日じゅう} * \text{北の風} \quad (\because \text{はじめ} + \text{のち} = \text{一日じゅう}) \\ &= (\text{一日じゅう} \ \text{北の風}) \end{aligned}$$

となる。

このように情報は形式的に簡約化できることがわかった。以上の考察から、天気予報では、

$$\begin{cases} \text{(I)} \ A * B + A * B' = A * (B + B') \\ \text{(II)} \ A * B + A' * B = (A + A') * B \\ \text{(III)} \ D + D' = \begin{cases} D & (\text{if } D > D') \\ D' & (\text{if } D < D') \\ E & (\text{if } D + D' = E) \end{cases} \end{cases}$$

$$\begin{aligned} &A_1 * A_2 * \dots * A_i * C * A_j * \dots * + \\ &B_1 * B_2 * \dots * B_i * C * B_j * \dots * \\ &= / \end{aligned}$$

なる変形が考えられる。ここに、(I),(II)は形式的変形、(III)は算術の知識を用いた変形である。

§ 5-3-2 文生成の爲の文表現簡約化

簡約化された文表現を得る爲には、前節の方法は強力であった。が、前節の方法では、「はじめ はれ のち はれ時々くもり」を「はじめ はれ のち 時々くもり」とは簡約できぬ。この簡約は、知識表現レベルでの簡約ではなく、文表現¹²レベルでの簡約化と看做すことができる。

従って、簡約化された文表現を得るには、

(1) 知識表現を論理的に簡約化する

(例) 「はじめ北の風がやや強く、のち南の風がやや強いでしょう」
→「はじめ北の風、のち南の風がともにやや強いでしょう」

言い換え

(2) 文表現レベルで、わかっているものは省略する

(例) 「はじめ晴れのち晴れ時々曇りでしょう」
→「はじめ晴れのち時々曇りでしょう」

の二つを行なわねばならぬ。

本論文では、簡約化された天気予報文を生成することを目標にしていたが、上記のうち(1)のものしか作れなかった。この(2)の種類¹³の簡約化を行なう関数は、REDUCE2である。ただし、REDUCE2は、生成された文に作用させるのではなく、知識表現に作用させた。

§ 5-4 実行結果

以上、説点、“で”の挿入、簡約化について色々述べたが、これらを実際に行なわせた例は、付録の図A-3, 図A-4, 図A-5に載せてある。

これらを見ると、既に予想されたように、簡約化しなおかつ、説点、“で”等を挿入した文が最も自然でわかりよいことがわかる。

第6章 考察・検討

本論文では、天気予報の世界における知識表現と言語表現との相互変換を行なうシステムを実際に作成した。ここでは、このシステムの拡張性や曖昧文処理について考察・検討を加える。

(1) 本システムの対象範囲と拡張性について

本システムは、テレビ(NHK第一放送)の朝・昼・晩に放送されている天気予報の『各地の天気予報』文を取り扱うことを目標として作成されているが現段階のプロケラムでは、『北の風が次第に強くなり』を受容出来ず、不適格文として排除してしまう。文生成の際も、このような文を生成する機構はなく、また、知識表現にも“次第に”の情報をしまっておく位置が決められていない。これは、本システムが開発中であるからである。この文を受理できるためには、①図3-3のATNにおいて、状態55の弧2で“次第に”を認識した際のみ、“強く”の次に“なり”が来ているか否かチェックし、“なり”が来ていなければ不適格文、来ていれば適格文として処理するようにプロケラムの一部を変更すればよい。

また、本システムに『北部山沿いでは曇り曇りか曇り曇りでしょう。』なる文を入力すると、『北部山沿いでは、初めのうちは曇り、のち曇りか曇り』と解釈してしまう。これは、本システム作成の際、晴れ曇り、雨等の第一天気素と曇り、曇りか曇りという第二天気素とを特に区別せずに処理するようにしたからである。この文は、図3-3の状態57, 58間の弧として、接続語の弧の他にもう一本継続語の弧を設ければよい。但し、この継続語の弧を通過するには、状態56, 57間の天気素の弧を第二天気素により通過していなければならない、という付加条件が必要である。このように手直しすれば、上述の文は受理できる。しかし、この文の知識表現は通常のものと違ふが、たゞ、変則的なものになるので、この類の文生成の為の手続きが特に必要になる。

その他、本システムが現段階で処理できぬ文型としては『おそくなって雨が強く降るでしょう』や『おそくなって雨の降る所があるでしょう』がある。これらの文を受理するには、図3-3の状態59に、状態55(風文の強度に関する表現を解析する部分)のようなものを接続させ、うまく付加条件を定めれば解析受理は可能と考えられる。

更に、本システムでは、天気情報は時系列的に述べられている、という暗黙の仮定を認めた。従って、『東京都 今日 曇り ですが 朝のうちは 晴れ でしょう』なる入力文をうまく処理しきれない。それは、

いつ曇りなのか、を推論しようとして、“今日は”と後の“朝のうさは”を参照する時に起こる。“今日は”という時間帯語の枠組においては、“朝のうさは”、“のぞ”もしくは“日中”、“おえくなて”という順に時区切語が現われる筈なのに、ここでは、“朝のうさは”の一つ前の時区切語を求められている。しかし“朝のうさは”より以前の時区切語は“今日は”という時間帯語の枠組内ではあり得ないのである。従って、プログラムはこの処理をうまくできなくなるのである。

この事態を処理するには、例えば、この文では時間関係の語として“今日は”と“朝のうさは”しかないことから、“朝のうさは以降は曇り”と察知するようにプログラムを組めばよいであろう。しかし、『今日は、曇りですか 朝のうさは 雨 夜は雪でしょう』という文をも受理解させるのは大変困難であろう。最後のよな文は、NHKの天気予報には未だ出てきたことはないの、例外扱いが可能なが前者の例は実際にもあるので、是非受理解可能にする必要がある。

以上のように、本システムでは、正しく時系列的に情報が述べられた『各地の天気予報』文をその対象の範囲内に含んでいるが、非時系列的に情報を述べたものは適切に処理できぬ。この処理を行なうには、もと注意深い推論の機能があれば良いと考えられる。

また、テレビ以外の電話や新聞の天気予報は本システム内の登録語彙を多少変更すれば、大部分のものが扱えるようになるが、電話の場合、『今日、あす、あさってとまたはれ』という特有な表現があるので、このような文をも含めて処理するためには、本システムを幾分変更する必要がある。しかし、電話新聞の天気予報文も、テレビのものと同基本的には同じ構文構造であると考えられるので、本システムの延長上にあると考えられる。

(2) 文解析・文生成の際の文の曖昧性、多様性について

自然言語を機械処理する際には、文表現の多様性(一つの伝達情報に対し種々の表現形態が対応すること)や曖昧性(一つの文表現に対し複数の解釈が対応すること)の処理が重要である。このうち、多様性は取り扱い可能な文型の種類や辞書語彙を予め豊富に登録しておくことにより一応処理可能である。そこで、ここでは取り扱いの極めて難しい曖昧性について考察を加えることにする。

曖昧性の問題は、文解析・文生成のいずれの場合にも生ずる。例えば、例文「あすは晴れのち曇り 山浴いは雨」では、「一般地は晴れのち曇り、山浴いは一日中雨」、「初めのうちは全域晴れ、のち一般地は曇りで山浴いは雨」の2種類の解釈が存在する。その為このような文を解析する際には何らかの方法で一方を選択せねばならず、文生成時にはこのような文生成は避けるべきである。このように自然言語処理の際には、これら曖昧性の処理を行なう手法の確立が不可欠である。

文の曖昧性、特に、天気予報文における文の曖昧性は、相田仁氏によ

り既に研究されていて、⁶⁾人間の天気予報文の解釈に関して、『①「山ぞい」のように地域を明確に示す語が文中に存在する場合には、天気予報文をその地域に関する節と、それ以外の地域に関する節の区切りが最大の区切りとなる。②「山ぞいは雪」という表現のように特に結びつきやすい表現がある③いくつかの「晴雨状況」単語が文中に存在する場合、それらの天候の変化の様子により、晴れ→くもり→一時雪のように連続的な場合にはその表現を1つの大きな節としてとらえ、くもり→晴れ→雪のように大きな変化を含むときには、そこで節を区切り、て解釈する。④「て」「ひすか」のような語句は節の区切りと解釈され、その働きは「ひすか」の方が「て」よりも強い。⑤風の強さに関しては文中で最も近い風の状況のみを修飾する。』が言えること、及び『①の影響が最も強く、続いて②、③、④の順に解釈に影響する』(以上参考文献(6)からの抜粋)ことを心理実験から得ている。同氏は、この結果から、①時刻と地域の情報をともに明示する②「ひすか」を区切りに挿入する③風の場合には「北の風のさ南の風か」と差に⁷⁾のように適切な語を補うことの3方法を曖昧性解消の方法として提案している。

一方、半田創一氏は、相田氏の研究を踏まえ、天気予報文における解釈の選択確率を心理実験により求めている。⁷⁾同氏は、相田氏の結果に加え、①「て」を挿入することにより40%程、「ひすか」により60%程度解釈の選択確率を変えられる事ができる、②従って「て」や「ひすか」により人間の解釈選択過程を100%左右できるわけではないこと、の2点を明らかにしている。

以上、相田氏、半田氏の結論から次のことが言えよう。

① 文生成の時

(a) 時間と地域の情報(即ち、時区切語と地域名語)を必ず明示する

(i) “ととに”の語が用いれるときには常に用いる

(ii) 通常の区切りには「て」を用い、大きな区切り部に「ひすか」を用いる

(iii) 『晴れのさ曇り山浴いでは雨』とせず『一般地では晴れのさ曇り。山浴いでは一日雨』とか『一般地では晴れのさ曇り。山浴いでは初め晴れのさ雨』のようた二つの文に分解する

② 文解析の時

(a) 入力文が曖昧であることを入力者に告げ、曖昧でない文の再入力を求める

(i) 可能な限りの解釈を行ない、どの解釈を採るかを入力者に尋ねる

本論文のシステムでは、文生成の際には上記①の(a)(i)と②の(a)のようなことを行なって曖昧性の少ない文を作ることを行なったが、文解析の際には、例えば『今日は晴れのさ曇り山浴いは雨』に対しては『今日は、一般地は初めのさ晴れのさ曇り、山浴いは一日雨』というように新たな地域名語が現われたらそこで大きく切れると仮定することによって曖昧性の問題を避けた。しかし、これでは不十分なので、今後は、上記②の(a)(i)の實現が本システムにとって最も重要である。

第7章 結論

本論文では、天気予報の世界に限定した文表現と知識表現との相互変換を行なうプログラムを作り実際に動作させることを試みた。

まず第一章で、自然言語処理に、今日、いかなる意義があるかを見た後従来いかなる研究が行なわれてきたかを顧み、文解析・文生成を共にハラスよく扱った研究があまりなされていなかったことを指摘し、本研究の意義を明らかにした。

第二章では、自然言語処理と知識表現の従来の研究を概観し、今日の両分野における現状を把握し、問題点を指摘した。

第三章では、本研究の対象である天気予報の文法を作り、天気予報の文構造を明らかにした。次いで、天気予報の知識表現としてマトリックス形式かリスト形式が適していること、及びリスト形式の優越性について論じた。

第四章は、まずATN文法を概説し、次いで、天気予報文の解析にはATNが向いていることを述べ、本研究で作った文解析アルゴリズムの概要とその動作具体例を示した。

第五章では、知識表現から文生成をする様々な方式を論じ、詭点や“で”の重要性を明らかにし、次いで、文の簡約化について幾分詳細に論じ、そのアルゴリズムをも示した。

第六章では、本研究で作成したシステムの対象範囲と拡張性及び入出力文の多様性・曖昧性の問題について考察し、曖昧性解消の方法をいくつか提案した。

本論文では、文表現と知識表現との相互変換を、実際にプログラムを作って行なってみた。このことか、通常文解析の方が生成よりも難かしいと思われていたことが、実は必ずしも正しく、自然で分かりやすい文生成もきわめて難かしいことがわかった。また、知識表現の際には、推論機能が重要であることがわかった。今回の、天気予報世界に限られた場合の知識を踏み台に、より一般の場合での相互変換が可能となる。

謝辞

本論文の研究にあたり、終始親切に御指導下さ。藤崎博也教授ならびに広瀬啓吉助教授に深く感謝致します。

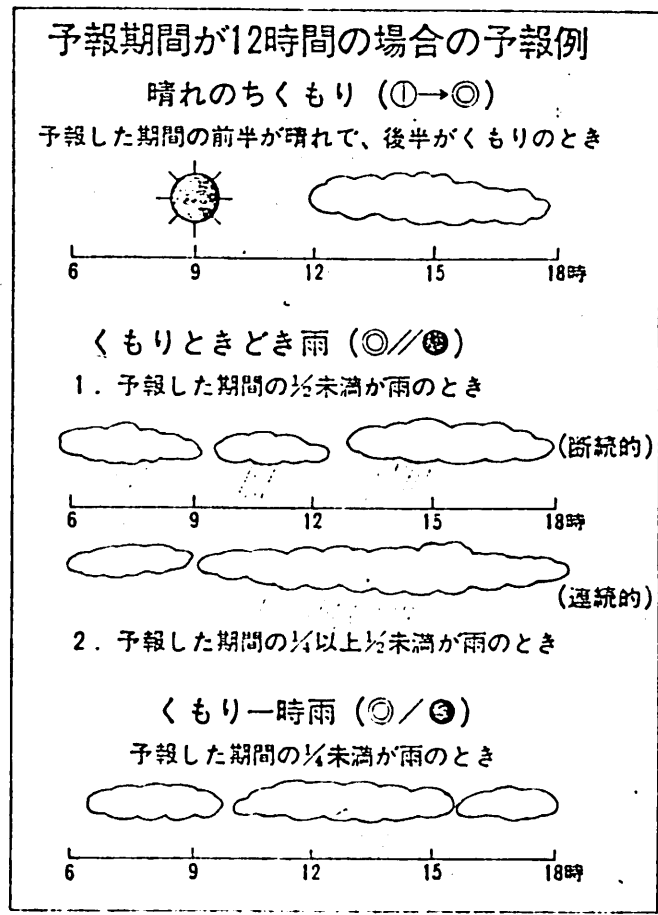
また、さまざまなお面におたり助言助力をいただいた星合忠・池守真人両先輩をはじめとする研究室の皆様にも感謝致します。

参考文献

- 2 ① Winograd, T: *Understanding Natural Language*, Academic Press Inc.
New York (1972)
- 2 ② Woods, W.A.: *Transition Network Grammars for Natural Language Analysis*, CACM, 13 (1970)
- 3) Fillmore, C: *The Case for Case In Universals in Linguistic Theory*, E. Bach and R.T. Harms (eds.). Holt, Rinehart & Winston, New York, (1968)
- 4) E.F. Codd: *Relational Completeness of Data Base Sublanguages*, Courant Computer Science Symposia 6, (1971)
- 5) 西田 富士夫 : 言語情報処理 , コロナ社 (1981)
- 2 ⑥ 相田 仁 : 自然言語における曖昧性の分析と処理 , 東京大学電気工学科卒業論文 (1980)
- 2 ⑦ 半田 創一 : 言語表現の曖昧性とその人間及び機械による処理 , 東京大学電気工学科卒業論文 (1981)

	内 容
のち	予報期間の1/2を境にする (期間の1/6ぐらいは前後してもよい)
一時	現象が連続して起こりその期間が 予報期間の1/4未満
時々	現象が断続して起こりその合計時 間が予報期間の1/2未満
	現象が一時連続して起こるか その時間が予報期間の1/4以上1/2未満

表A-1. 予報用語としての“のち”、“一時”
“時々”の意味内容



(注) 図A-1と表A-1
は、気象庁発行の
パンフレット『天気予
報の利用』から
引用した。

図A-1. 予報用語としての“のち”、“一時”
“時々”の意味内容

$[(CM1) \{ (x_{1-1} (y_{1-1})) (x_{1-2} (y_{1-2})) \dots (x_{1-m} (y_{1-m})) \}]$

$[(CM2) \{ (x_{2-1} (y_{2-1})) (x_{2-2} (y_{2-2})) \dots (x_{2-m'} (y_{2-m'})) \}]$

$[(CMn) \{ (x_{n-1} (y_{n-1})) (x_{n-2} (y_{n-2})) \dots (x_{n-m''} (y_{n-m''})) \}]$

例. $[(北部では) \{ (初め (雨)) (のち ((<刮>-時晴れ)) \}]$
 $[(南部では) \{ (初め (雨)) (のち (所により (<刮>-時は水)) \}]$

(注) CM: 地域名, x: 時区切語, y: 天候状況

(1) 天候の知識表現

$[(CM1) \{ (x_{1-1} (y_{1-1}) z_{1-1}) (x_{1-2} (y_{1-2}) z_{1-2}) \dots (x_{1-m} (y_{1-m}) z_{1-m}) \}]$

$[(CM2) \{ (x_{2-1} (y_{2-1}) z_{2-1}) (x_{2-2} (y_{2-2}) z_{2-2}) \dots (x_{2-m'} (y_{2-m'}) z_{2-m'}) \}]$

$[(CMn) \{ (x_{n-1} (y_{n-1}) z_{n-1}) (x_{n-2} (y_{n-2}) z_{n-2}) \dots (x_{n-m''} (y_{n-m''}) z_{n-m''}) \}]$

例. $[(北部では) \{ (朝のうさ(北の風) ぬか) (のち (朝の風) ぬか) \}]$

$[(南部では) \{ (-日 (南よりの風) NIL) \}]$

(2) 風の知識表現

↑ 風の強度に関する情報が無いということを示わしている。

都県名語 時間帯語

(TM JT (KWW WW))

(3) 天気予報文の知識表現

(注) 上記知識表現の細部構造を次のように略して記すこともある。

(i) $P \triangleq (CM_i)$ (RPS 地域名語のリスト) (北の風と南の風に共通)

$RT \triangleq (x_{i-j} (y_{i-j}))$, $Q \triangleq \{ RT_1 \ RT_2 \ \dots \ RT_m \}$ (RTの並び)

$W \triangleq (P \ Q)$ (RPS PとQのリスト), $WWW \triangleq (w_1 \ w_2 \ \dots \ w_m)$ (Wの並び)

以上天候の知識表現について。風は、KRT, KQ, KW, KWWWと記し、それ以外天候のRT, Q, W, WWWと同じ。

図 A-2. 実際に行った知識表現の書式

```

(PF X1)
*INPUT SENTENCE IS AS FOLLOWING :*
(天気予報 2020年 9月 25日 15時 50分)
(TM25 JT15 NS05 1500 5000)
*〈TOFUKEH-HEI〉 IS FOUND.*
*〈JIKAN-TAI-GO〉 IS FOUND.*
*THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.*
*〈KYOUDO〉 IS MISSING.*
(天気予報)
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 FAIL*
*〈TENKOU-SU〉 SUCCESS*
*TENKOU COMPLETED :*
(OK)
*THE SENTENCE*
(天気予報 2020年 9月 25日 15時 50分)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :*
(天気予報 2020年 9月 25日 15時 50分) ((天気予報)) ((予報)) ((予報)) ((予報))
((( )))
*OK! ACCEPTED.*
>
(GF00 KNOW)
(天気予報) 予報 予報
>
(S0 KNOW)
(天気予報) 予報 予報
>(SS0 KNOW)
(天気予報) 予報 予報
>

```

☒ A-3. 総合的動作例 (その1)

```

(PF X2)
*INPUT SENTENCE IS AS FOLLOWING :*
(ウイタマウ コウノ トクノ トクノ トクノ トクノ トクノ トクノ)
(THIS JI15 I305 T520 S700 Y505 SR00)
*〈TOFUJEN-HEI〉 IS FOUND.*
*〈JIKAN-TAI-GO〉 IS FOUND.*
*THE CANDIDATE FOR NAEE-SO-1ST IS FOUND.*
*〈KYOUDDO〉 IS MISSING.*
(OL I35" 26V テ"360)
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 SUCCESS*
*TENKOU COMPLETED :*
((OL I35" 26V))
*THE SENTENCE*
(ウイタマウ コウノ トクノ トクノ トクノ トクノ トクノ トクノ)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :*
(ウイタマウ コウノ) (((((E"Jイテ") ((E/ウチ (フユウ/カヒ" NIL)))))) ((E"Jイテ") ((E/ウチ ((OL
イ35" 26V)))))))))
*OK! ACCEPTED.*
>

(GF00 KNOW)
((E"Jイテ") E/ウチ OL I35" 26V テ"360。)
>
(SO KNOW)
(E"Jイテ" E/ウチ フユウ/カヒ" テ"360。)
>(SSO KNOW)
(E"Jイテ" E/ウチ フユウ/カヒ" テ"360。)
>

```

図A-3. 総合的動作例(その2)

```

(PF X3)
*INPUT SENTENCE IS AS FOLLOWING :*
(トクサケ 730 4230/40" 2717 4740/40" 0L 4417 267 7"347)
(TN30 J105 K505 J620 K555 TS20 SZ05 TS05 SP00)
*TOFUKEN-MEID> IS FOUND.*
*JIFAN-TAI-GO> IS FOUND.*
*THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.*
*KNYOUDO> IS MISSING.*
*TRY ONCE MORE*
*THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.*
*KNYOUDO> IS MISSING.*
(0L 4417 267 7"347)
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*TENKOU-KAI> SUCCESS*
*TENKOU COMPLETED :*
(0L 4417 267)
*THE SENTENCE*
(トクサケ 730 4230/40" 2717 4740/40" 0L 4417 267 7"347)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :*
(トクサケ 730 (((((0"0(477") ((77/07 (4230/40") NIL) (271700 (4740/40") NIL))) ((
(730"0(477") ((4417 ((0L 4417 267)))))))
*OK! ACCEPTED.*
>
(GFOO KNOW)
(イッ"0(477") 477 0L 4417 267 7"347。)
>
(SO KNOW)
(0"0(477" 77/07 4230/40" 7" 271700 4740/40" 7"347。)
>(SSO KNOW)
(0"0(477" 77/07 4230/40" 7" 271700 4740/40" 7"347。)
>

```

図A-3. 総合的動作例(その3)


```

(FF X5)
*INPUT SENTENCE IS AS FOLLOWING:
(197301 730 4017/60 269 /# 7% 7"3a2)
(TM30 JT05 KS45 TS05 JG05 TS00 SR00)
*〈TOKUKEN-MEI〉 IS FOUND.
*〈JIKAN-TAI-00〉 IS FOUND.
*THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.
*〈KYOUJOU〉 IS MISSING.
(269 /# 7% 7"3a2)
*THE CANDIDATE FOR TENROU-SO-1ST IS FOUND.
*〈TENROU-KAI〉 FAIL.
*〈TENROU-SO〉 SUCCESS.
*TENROU COMPLETED :
(269)
*TRY ONCE MORE.
*THE CANDIDATE FOR TENROU-SO-1ST IS FOUND.
*〈TENROU-KAI〉 FAIL.
*〈TENROU-SO〉 SUCCESS.
*TENROU COMPLETED :
(7%)
*THE SENTENCE
(197301 730 4017/60 269 /# 7% 7"3a2)
*IS ACCEPTED AS A WEATHER FORECAST ONE.
*AND ITS KNOWLEDGE REPRESENTATION IS THE FOLLOWING :
(197301 730 (((E"J477") ((J477 (4017/60) NIL)))) ((G0"J477") (05"7/7# (269)
) (/# (7%))))))
*OK! ACCEPTED.
>

(GF00 KNOW)
(G0"J477") 05"7/7# 269 7# /# 7% 7"3a2 .)
>
(S0 KNOW)
(E"J477" J477 4017/60 7"3a2 .)
>(SS0 KNOW)
(E"J477" J477 4017/60 7"3a2 .)
>

```

図A-3. 総合的動作例(その5)

```

(PF X6)
*INPUT SENTENCE IS AS FOLLOWING : *
(トクサウ アウツ アウ テ" /チ ア" トルア"アウ)
(TN30 JG00 TS05 KZ00 JG25 TS00 SR05)
*〈TOFUJEN-MEI〉 IS FOUND.*
*WARNING : 〈KAZE-SETSU〉 IS MISSING.*
(アウツ アウ テ" /チ ア" トルア"アウ)
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 FAIL*
*〈TENKOU-SO〉 SUCCESS*
*TENKOU COMPLETED : *
(アウ)
*TRY ONCE MORE*
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 FAIL*
*〈TENKOU-SO〉 SUCCESS*
*TENKOU COMPLETED : *
(アウ)
*THE SENTENCE*
(トクサウ アウツ アウ テ" /チ ア" トルア"アウ)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING : *
(トクサウ アウツ アウ (NIL (((アウ"アウ"アウ) ((アウツ (アウ)) (チ (アウ)))))))
*OK! ACCEPTED.*
>

(GFOO KNOW)
(アウ"アウ"アウ) アウツ アウ テ" /チ ア" テ"アウ。
>

```

☒ A-3. 総合的動作例(その6)

```

(PF X7)
*INPUT SENTENCE IS AS FOLLOWING :*
(2"0770 00P0 72/50" K" 01/070 PP 7B7 0L 7" 777"777"47"0 7EY 7"5a7)
(TN10 J115 K500 S100 J045 D600 TSUY TS20 K200 Ch15 TS05 SP00)
*<TOFUKEN-HEI> IS FOUND.*
*<JILAN-1AI-00> IS FOUND.*
*THE CANDIDATE FOR NAZE-S0-1ST IS FOUND.*
*<RYOUDO> WILL BE PERHAPS FOUND.*
(0L 7" 777"777"47"0 7EY 7"5a7)
*THE CANDIDATE FOR TENKOU-S0-1ST IS FOUND.*
*<TENKOU-KAI> FAIL.*
*<TENKOU-S0> SUCCESS.*
*TENKOU COMPLETED :*
(0L)
*TRY ONCE MORE.*
*THE CANDIDATE FOR TENKOU-S0-1ST IS FOUND.*
*<TENKOU-KAI> FAIL.*
*<TENKOU-S0> SUCCESS.*
*TENKOU COMPLETED :*
(7EY)
*THE SENTENCE*
(7"0770 00P0 72/50" K" 01/070 PP 7B7 0L 7" 777"777"47"0 7EY 7"5a7)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING : *
(7"0770 00P0) (((0"0477") ((01/070) (72/50") PP))) (((0770"0777"0) ((01/07 (0L))))))
(((777"777"47"0) ((01/07 (7EY))))))
*OK! ACCEPTED.*
>

(GFOO KNOW)
(0770"0777"0) 01/07 0L 7"5a"、 777"777"47"0) 01/07 7EY 7"5a7。)
>
(SO KNOW)
(0"0477" 01/070 72/50" K" PP 7B( 7"5a7。)
>(SSO KNOW)
(0"0477" 01/070 72/50" K" PP 7B( 7"5a7。)
>

```

図A-3. 総合的動作例(その7)


```

(PF 28)
*INPUT SENTENCE IS AS FOLLOWING :*
(2"J274) 3000 47/40" 4" 30/200 TP 200) 0L 7" 427"477"477" 200 (7) 20078 4 27414 7"2
37)
(TN10 JT15 K500 S10A J615 U600 150Y 1520 F100 CM15 1505 K700 1510 S103 T515 S100
)
*〈TOFUREN-MEI〉 IS FOUND.*
*〈JIKAN-TAI-GO〉 IS FOUND.*
*THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.*
*〈KYODOO〉 WILL BE PERHAPS FOUND.*
(0L 7" 427"477"477" 200 7" 20078 4 27414 7"2)
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 FAIL*
*〈TENKOU-SO〉 SUCCESS*
*TENKOU COMPLETED :*
(0L)
*TRY ONCE MORE*
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 FAIL*
*〈TENKOU-SO〉 SUCCESS*
*TENKOU COMPLETED :*
(2E2)
*TRY ONCE MORE*
*THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.*
*〈TENKOU-KAI〉 SUCCESS*
*TENKOU COMPLETED :*
((27478 4 27414))
*ERROR-S : CHIIKI-MEI OR JIKUGIRI-GO IS MISSING.* ——— *
*AT CHECK POINT 9.*
*THE SENTENCE*
(2"J274) 3000 47/40" 4" 30/200 TP 200) 0L 7" 427"477"477" 200 7" 20078 4 27414 7"2
37)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :*
(2"J274) 3000 (((E"J477" ))) ((30/200 (47/40" TP))) (((E"J477" ))) ((30/200 (0L)))
(((427"477"477" ))) ((NIL (2E2) (NIL ((27478 4 27414))))))
*OK! ACCEPTED.*
>
(GFOO KNOW)
(E"J477" 30/200 0L 7"2) " 427"477"477" (NIL) 200 7" (NIL) 27478 4 27414 7"2)
>
(SO KNOW)
(E"J477" 30/200 47/40" 4" TP 200) 7"2)
>(SSO KNOW)
(E"J477" 30/200 47/40" 4" TP 200) 7"2)
>

```

☒A-3. 総合的動作例(その8)


```

(FF 311)
*INPUT SENTENCE IS AS FOLLOWING :
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
*TOUJEN-NEI IS FOUND.*
*THE CANDIDATE FOR KALL-SO-1ST IS FOUND.*
*KALL-SO FAIL BUT KAZE-KAI SUCCESS*
*RYOUHO WILL BE PERHAPS FOUND.*
(FF 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
*THE CANDIDATE FOR TEMOU-SO-1ST IS FOUND.*
*TEMOU-KAI FAIL*
*TEMOU-SO SUCCESS*
*TEMOU COMPLETED :
(FF)
*TRY ONCE MORE*
*THE CANDIDATE FOR TEMOU-SO-1ST IS FOUND.*
*TEMOU-KAI FAIL*
*TEMOU-SO SUCCESS*
*TEMOU COMPLETED :
(FF)
*THE SENTENCE
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
*ON! ACCEPTED.*
>

(GF00 KNOW)
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
>

(SO KNOW)
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
>

(SSSO KNOW)
(FF 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320)
>

```

図A-3. 総合的動作例(その11)


```

(FF 113)
*INPUT SENTENCE IS AS FOLLOWING :
(107321 730 E6 9/HC 290 13/HC 6 937 74 7/940)
(T850 J05 K030 S210 K520 ST6A 1804 1500 SPO0)
*107321-HE12 IS FOUND.*
*11100-TAI-00 IS FOUND.*
*THE CANDIDATE FOR KAZE-S0-1ST IS FOUND.*
*KAZE-S0> FAIL BUT <KAZE-KAJ> SUCCESS*
*110000 WILL BE PERHAPS FOUND.*
(74 7/940)
*THE CANDIDATE FOR TENKOU-S0-1ST IS FOUND.*
*110000-KAJ FAIL*
*110000-S0 SUCCESS*
*110000 COMPLETED :
(74)
*THE SENTENCE*
(107321 730 E6 9/HC 290 13/HC 6 937 74 7/940)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :
(107321 730 (((E"0/HC") ((110000-KAJ) *074))) (((110000-S0) (
(110000 (74))))))
*OK! ACCEPTED.*
>

(GFOO KNOW)
(110000-S0) (110000 (74) 7/940.)
>

(SO KNOW)
(E"0/HC" (110000-KAJ) E6 9/HC 290 13/HC 6 937 (7/940) .)
(S50 KNOW)
(E"0/HC" (110000-S0) E6 9/HC 290 13/HC 6 937 (7/940) .)
>

```

図A-3. 総合的動作例(その13)

```

(FF X14)
*INPUT SENTENCE IS AS FOLLOWING :*
(ETSAKE TAU TAI/HC /# I#E' ETE/HC" H" TT W#I' NL T' D#I)
(TH30 J105 K500 J025 S700 K550 STGA D400 TSU# 1520 SP00)
*TOFUJEN-GLI> IS FOUND.*
*SUJIKAN-TAI-OO> IS FOUND.*
*THE CANDIDATE FOR KAZE-SO-IST IS FOUND.*
*KYUJUDO IS MISSING.*
*TRY ONCE MORE*
*KAZE-SO-IST IS IMPLICITLY FOUND.*
*KAZE-SO FAIL BUT KAZE-KAI SUCCESS*
*KYUJUDO WILL BE PERHAPS FOUND.*
(OI T' D#I)
*THE CANDIDATE FOR TENKOU-SO-IST IS FOUND.*
*TENKOU-KAI FAIL*
*TENKOU-SO SUCCESS*
*TENKOU COMPLETED :*
(OI)
*THE SENTENCE*
(ETSAKE TAU TAI/HC /# I#E' ETE/HC" H" TT W#I' NL T' D#I)
*IS ACCEPTED AS A WEATHER FORECAST ONE.*
*AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING :*
(ETSAKE TAU) (((E'D#I) ((D'X/2# (T'/HC") NIL) (/# ((T'/HC" I#E' ETE/HC")) TT
))) (((E'D#I) ((I#E' (NL))))))
*OK! ACCEPTED.*
>
(GF00 KNOW)
(E'D#I) (I#E' NL T' D#I)
>
(S0 KNOW)
(O'D#I) (E'D#I) (I#E' ETE/HC" H" TT W#I' T' D#I)
(S00 KNOW)
(O'D#I) (E'D#I) (I#E' ETE/HC" H" TT W#I' T' D#I)
>

```

図A-3. 総合的動作例(その14)

"THE SENTENCE"
 (OK J# DEU TUDOT#T FODIEBU IDOTX T"UBO) ← 入力文
 "IS ACCEPTED AS A WEATHER FORECAST ONE."
 "AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING : "
 ((("OK"J#T") (OK"KJOT# OK)) (J# (DEU)) (TUDOT#T (FODIEBU IDOTX)))) ← 入力文の知識表現

>(PRETTY (BSFN1 (CADAR ABC-IND))
 (OK"KJOT# OK , J# DEU , TUDOT#T FODIEBU IDOTX T"UBO) ← 上記知識表現からの、BSFN1による生成文
 >(PRETTY (BSFN2 (CADAR ABC-IND))
 (OK"KJOT# OK , J# DEU T" TUDOT#T FODIEBU IDOTX T"UBO) ← 上記知識表現からの、BSFN2による生成文

図A-4. 文生成関数 BSFN1 と BSFN2 の出力例 (その1)

"THE SENTENCE"
 (OK J# I#B" DEU T"ZU" TUDOT#T FODIEBU IDOT#T T"UBO) ← 入力文
 "IS ACCEPTED AS A WEATHER FORECAST ONE."
 "AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING : "
 ((("OK"J#T") (OK"KJOT# OK)) (J# ((OK I#B" DEU)) (TUDOT#T (FODIEBU IDOT#T)))) ← 入力文の知識表現
 "OK! ACCEPTED."
 "OI! ACCEPTED."

>(PRETTY (BSFN1 (GFN1 (CADAR ABC-IND)))
 (OK"KJOT# OK , J# I#B" DEU , TUDOT#T FODIEBU IDOTX T"UBO) ← 上記知識表現からの BSFN1による生成文
 >(PRETTY (BSFN2 (GFN1 (CADAR ABC-IND)))
 (OK"KJOT# OK , J# I#B" DEU T" TUDOT#T FODIEBU IDOTX T"UBO) ← 上記知識表現からの BSFN2による生成文

} 簡約化出力

図A-4. 文生成関数 BSFN1 と BSFN2 の出力例 (その2)
 ((注) 簡約化が REDUCE2により行なわれている)

49

```

+SFN1#1: ((GNB"XJQ# GNL)) (J# (GNL I#B" DEU))) (TUKA#T (HJOC#B Y IJHAK#X))
+REDUCE2#2: ((GNB"XJQ# GNL)) (J# (GNL I#B" DEU)))
+LIST-2ND#3: ((GNB"XJQ# GNL))
-LIST-2ND#3: GNL
+HELP1-RED2#3: ((GNB"XJQ# GNL)) (J# (GNL I#B" DEU)))
+LIST-2ND#4: ((GNB"XJQ# GNL))
-LIST-2ND#4: GNL
+LIST-2ND#4: ((J# (GNL I#B" DEU)))
-LIST-2ND#4: ((GNL I#B" DEU))
+LIST-2ND#4: ((J# (GNL I#B" DEU)))
-LIST-2ND#4: ((GNL I#B" DEU))
-HELP1-RED2#3: ((GNB"XJQ# GNL)) (J# ((I#B" DEU)))
-REDUCE2#2: ((GNB"XJQ# GNL)) (J# ((I#B" DEU)))
+REDUCE2#2: ((GNB"XJQ# GNL)) (TUKA#T (HJOC#B Y IJHAK#X))
+LIST-2ND#3: ((GNB"XJQ# GNL))
-LIST-2ND#3: GNL
+HELP1-RED2#3: ((GNB"XJQ# GNL)) (TUKA#T (HJOC#B Y IJHAK#X))
+LIST-2ND#4: ((GNB"XJQ# GNL))
-LIST-2ND#4: GNL
+LIST-2ND#4: (TUKA#T (HJOC#B Y IJHAK#X))
-LIST-2ND#4: (HJOC#B Y IJHAK#X)
+LIST-2ND#4: (TUKA#T (HJOC#B Y IJHAK#X))
-LIST-2ND#4: (HJOC#B Y IJHAK#X)
-HELP1-RED2#3: ((GNB"XJQ# GNL)) (TUKA#T (HJOC#B Y IJHAK#X))
-REDUCE2#2: ((GNB"XJQ# GNL)) (TUKA#T (HJOC#B Y IJHAK#X))
-SFN1#1: ((GNB"XJQ# GNL)) (J# ((I#B" DEU))) (TUKA#T (HJOC#B Y IJHAK#X))
+BSFN1#1: ((GNB"XJQ# GNL)) (J# ((I#B" DEU))) (TUKA#T (HJOC#B Y IJHAK#X))

```

図 A-5. GFN1 と REDUCE2

左図は、GFN1 が REDUCE2 を呼ぶ GFN1 の入力知識表現が簡約化される様子を示している。
丸

右図は、REDUCE2 自体の動作を示している。

出力は不格かに簡約化されているのかわかる。

図 A-6. REDUCE2 の動作例

```

+REDUCE2#2: ((GNB"XJQ# GNL)) (J# (GNL I#B" DEU)))
+LIST-2ND#3: ((GNB"XJQ# GNL))
-LIST-2ND#3: GNL
+HELP1-RED2#3: ((GNB"XJQ# GNL)) (J# (GNL I#B" DEU)))
+LIST-2ND#4: ((GNB"XJQ# GNL))
-LIST-2ND#4: GNL
+LIST-2ND#4: ((J# (GNL I#B" DEU)))
-LIST-2ND#4: ((GNL I#B" DEU))
+LIST-2ND#4: ((J# (GNL I#B" DEU)))
-LIST-2ND#4: ((GNL I#B" DEU))
-HELP1-RED2#3: ((GNB"XJQ# GNL)) (J# ((I#B" DEU)))
-REDUCE2#2: ((GNB"XJQ# GNL)) (J# ((I#B" DEU)))

```



```

04900:      (PRINT '<TOFUKEN-MEI> IS FOUND.')
```

```

05000:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K-TM') (RETURN NIL
)))
```

```

05100:      (T (SETQ TOFUKEN-MEI 'TM30)))
```

```

05200:      (COND ((MEMB '*JIKAN-TAI-GO' (GET * 'HINSHI))
```

```

05300:      (SETQ JIKAN-TAI-GO *)
```

```

05400:      (PRINT '<JIKAN-TAI-GO> IS FOUND.')
```

```

05500:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K-JT') (RETURN NIL
)))
```

```

05600:      (T (SETQ JIKAN-TAI-GO 'JT10)))
```

```

05700:      (SETQ XXX (CONS * X))
```

```

05800:      (COND ((MEMB '*CHIIKI-MEI' (GET * 'HINSHI))
```

```

05900:      (SETQ CHIIKI-MEI *)
```

```

06000:      (PRINT '<CHIIKI-MEI> IS FOUND.')
```

```

06100:      (PRINT CHIIKI-MEI)
```

```

06200:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K2') (RETURN NIL
)))
```

```

06300:      (T (SETQ CHIIKI-MEI NIL)))
```

```

06400:      (COND ((MEMB '*JIKUGIRI-GO' (GET * 'HINSHI))
```

```

06500:      (SETQ JIKUGIRI-GO *)
```

```

06600:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K3') (RETURN NIL
)))
```

```

06700:      (T (SETQ JIKUGIRI-GO NIL)))
```

```

06800:      (COND ((MEMB '*KAZE-SO' (GET * 'HINSHI))
```

```

06900:      (SETQ KAZE-SO-1ST *)
```

```

07000:      (PRINT 'THE CANDIDATE FOR KAZE-SO-1ST IS FOUND.')
```

```

07100:      (COND ((NOT (MOVE))
```

```

07200:      (SETQ * 'SR00)
```

```

07300:      (PRINT 'WARNING : <SHUURYOU-GO> IS MISSING. ')
```

```

07400:      (PRINT 'DASHIOU IS ASSUMPTED.')
```

```

07500:      (PRINT 'AT CHECK POINT K4.5'))))
```

```

07600:      (T (COND ((EQ FLAG T)
```

```

07700:      (COND ((MEMB '*TENKOU-SO' (GET * 'HINSHI))
```

```

07800:      (SETQ X XXX)
```

```

07900:      (PRINT 'WARNING : <KAZE-SETSU> IS MISSING. ')
```

```

08000:      (SETQ ABCD-IN NIL ABCD-OUT NIL)
```

```

08100:      (RETURN T))
```

```

08200:      (T (PRINT 'ERROR : <KAZE-SETSU> IS MISSING!')
```

```

(RETURN NIL))))
```

```

08300:      (T (PRINT '<KAZE-SO-1ST> IS IMPLICITLY FOUND.))))))
```

```

08400:      (COND ((MEMB '*SETSUZOKU-GO' (GET * 'HINSHI))
```

```

08500:      (SETQ SETSUZOKU-GO *)
```

```

08600:      (COND ((NULL KAZE-SO-1ST)
```

```

08700:      (PRINT 'NO <KAZE-SO> IS FOUND.')
```

```

08800:      (PRINT 'THIS IS NOT A WEATHER FORECAST SENTENCE.')
```

```

08900:      (RETURN NIL)))
```

```

09000:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K5') (RETURN NIL
)))
```

```

09100:      (COND ((MEMB '*KAZE-SO' (GET * 'HINSHI))
```

```

09200:      (SETQ KAZE-SO-2ND *)
```

```

09300:      (SETQ KAZE-KAT T KAZE-SO NIL)
```

```

09400:      (PRINT 'KAZE-SO FAIL BUT KAZE-KAT SUCCESS')
```

```

09500:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K6') (RETURN
NIL))))))
```

```

09600:      (T (PRINT
```

```

09700:      'BOTH KAZE-SO-1ST AND SETSUZOKU-GO ARE FOUND. BU
```

```

T KAZE-SO-2ND IS NOT FOUND.')
```

```

09800:      (PRINT 'SYNTACTICALLY INVALID SENTENCE')
```

```

09900:      (RETURN NIL))))
```

```

10000:      (T (COND ((NULL KAZE-SO-1ST) (PRINT 'ERROR-ON KAZE-SO') (RETURN
NIL)))
```

```

10100:      (T (SETQ KAZE-KAT NIL KAZE-SO T))))))
```

```

10200:      (COND ((EQUAL * 'STGA)
```

```

10300:      (PRINT 'RYODOU WILL BE PERHAPS FOUND.')
```

```

10400:      (SETQ DOAI NIL)
```

```

10500:      (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT K7') (RETURN NIL
)))
```

```

)
10600: (COND ((MEMB "CHIKI-MEI" (GET * 'HINSHI))
10700: (SETQ CHIKI-MEI *)
10800: (COND ((NOT (MOVE)) (PRINT "AT CHECK POINT K8") (RETUR
N NIL))))))
10900: (COND ((MEMB "JIKUGIRI-GO" (GET * 'HINSHI))
11000: (SETQ JIKUGIRI-GO *)
11100: (COND ((NOT (MOVE)) (PRINT "AT CHECK POINT K9") (RETUR
N NIL))))))
11200: (COND ((MEMB "FURU-SHI" (GET * 'HINSHI))
11300: (COND ((EQ * 'FRM) (SETQ TOMONI T)) (T (SETQ TOMONI N
IL)))
11400: (COND ((NOT (MOVE)) (PRINT "AT CHECK POINT K10") (RET
URN NIL))))))
11500: (COND ((MEMB "DOAI" (GET * 'HINSHI))
11600: (SETQ DOAI *)
11700: (COND ((NOT (MOVE)) (PRINT "AT CHECK POINT K11") (RETU
RN NIL))))))
11800: (COND ((EQUAL * 'TSUY) (SETQ KYOUDO T) (COND ((NULL DOAI) (SE
TQ DOAI 'KAKA))))
11900: (T (PRINT "ILLEGAL SENTENCE : <GA> IS FOUND BUT <TSUYO
NU> IS NOT. *)))
12000: (T (PRINT "<KYOUDO> IS MISSING.") (SETQ KYOUDO NIL) (SETQ X (C
ONS * X))))
12100: (COND ((EQ <KAZE-KAI> T)
12200: (SETQ KAZE (LIST (LIST KAZE-SO-1ST SETSUZOKU-GO KAZE-SO-2ND)
))
12300: ((EQ <KAZE-SO> T) (SETQ KAZE (LIST KAZE-SO-1ST)))
12400: (T (PRINT "THERE ARE BUGS IN THIS PROGRAM!!!!!!")
12500: (RETURN NIL)))
12600: (SETQ J2 J1 J1 J0 J0 JIKUGIRI-GO)
12700: (SETQ XXXX X)
12800: (COND ((NOT (MOVE))
12900: (SETQ * 'SR00)
13000: (PRINT "WARNING ; NO WORDS!")
13100: (PRINT "DESHIOU IS ASSUMPTED.")))
13200: (COND ((MEMB "SHUURYOU-GO" (GET * 'HINSHI))
13300: (SETQ EFLAG T)
13400: (PRINT "DESHIOU IS FOUND."))
13500: (T (COND ((MEMB "KEIZOKU-GO" (GET * 'HINSHI))
13600: (SETQ XXXX X)
13700: (COND ((NOT (MOVE)) (PRINT "ERROR-K15")))))
13800: (COND ((MEMB "CHIKI-MEI" (GET * 'HINSHI))
13900: (COND ((NOT (MOVE)) (PRINT "ERROR-K16")))))
14000: (COND ((MEMB "JIKUGIRI-GO" (GET * 'HINSHI))
14100: (COND ((NOT (MOVE)) (PRINT "ERROR-K17"))))
14200: (COND ((MEMB "TENKOU-SO" (GET * 'HINSHI)) (SETQ EFLAG T) (
SETQ X XXXX))
14300: (T (SETQ EFLAG NIL) (SETQ X XXXX))))))
14400: (COND ((EQ FLAG T)
14500: (SETQ CHO CHIKI-MEI)
14600: (COND ((EQ EFLAG NIL)
14700: (COND ((NULL JIKUGIRI-GO)
14800: (SETQ HFLAG-J T)
14900: (COND ((NULL KYOUDO) (SETQ HFLAG-K T) (SETQ KRT
(LIST KAZE)))
15000: (T (SETQ HFLAG-K NIL) (SETQ KRT (LIST KAZ
E DOAI))))))
15100: (T (SETQ HFLAG-J NIL)
15200: (COND ((NULL KYOUDO) (SETQ HFLAG-K T) (SETQ K
RT (LIST JIKUGIRI-GO KAZE))))
15300: (T (SETQ HFLAG-K NIL) (SETQ KRT (LIST J
IKUGIRI-GO KAZE DOAI))))))
15400: (SETQ FLAG NIL)
15500: (T (COND ((NULL CHO) (SETQ CHO (FN1 TOFUKEN-MEI 2))))
15600: (SETQ P (LIST CHO))

```

```

15700: (COND ((NULL JIKUGIRI-GO) (SETQ JIKUGIRI-GO (FN2 JIK
AN-TAI-GO J1 2))))
15800: (COND ((NOT KYODO) (SETQ DOAI NIL)))
15900: (SETQ KRT (LIST JIKUGIRI-GO KAZE DOAI))
16000: (SETQ KQ (APPEND KQ (LIST KRT)))
16100: (SETQ KW (LIST F KQ))
16200: (SETQ KWW (APPEND KWW (LIST KW)))
16300: (SETQ KQ NIL)))
16400: (T (COND ((NULL CHIKI-MEI)
16500: (COND ((NULL JIKUGIRI-GO)
16600: (PRINT 'CHIKI-MEI OR JIKUGIRI-GO IS MISSING.
")
16700: (PRINT 'SYNTACTICALLY INVALID")
16800: (RETURN NIL)))
16900: (COND ((EQ HFLAG-J T) (SETQ J1 (FN2 JIKAN-TAI-GO J2
J0)) (SETQ J1 (LIST J1)))
17000: (T (SETQ J1 NIL)))
17100: (COND ((EQ HFLAG-K T)
17200: (COND ((EQ TOMONI T) (SETQ D1 (LIST DOAI))) (
T (SETQ D1 (LIST NIL))))))
17300: (T (SETQ D1 NIL)))
17400: (SETQ KRT (APPEND J1 KRT D1))
17500: (COND ((NEQ J1 NIL) (SETQ J1 (CAR J1))))
17600: (SETQ KQ (APPEND KQ (LIST KRT)))
17700: (COND ((EQ EFLAG NIL)
17800: (COND ((EQ KYODO NIL) (SETQ KRT (LIST JIKUGI
RI-GO KAZE) HFLAG-J NIL HFLAG-K T))
17900: (T (SETQ KRT (LIST JIKUGIRI-GO KAZE DOA
I) HFLAG-J NIL HFLAG-K NIL))))
18000: (T (COND ((EQ KYODO NIL) (SETQ DOAI NIL)))
18100: (SETQ KRT (LIST JIKUGIRI-GO KAZE DOAI))
18200: (SETQ KQ (APPEND KQ (LIST KRT)))
18300: (COND ((NULL CHO) (SETQ CHO (FN1 TOFUEN-ME
I 2))))
18400: (SETQ F (LIST CHO))
18500: (SETQ KW (LIST F KQ))
18600: (SETQ KWW (APPEND KWW (LIST KW)))
18700: (SETQ KQ NIL)))
18800: (T (COND ((NULL CHO) (SETQ CHO (FN1 TOFUEN-MEI 2)))
18900: (SETQ F (LIST CHO))
19000: (SETQ CHO CHIKI-MEI)
19100: (COND ((EQ HFLAG-J T) (SETQ J1 (FN2 JIKAN-TAI-GO J
2 2)) (SETQ J1 (LIST J1)))
19200: (T (SETQ J1 NIL)))
19300: (COND ((EQ HFLAG-K T)
19400: (COND ((EQ TOMONI T) (SETQ D1 (LIST DOAI)))
(T (SETQ D1 (LIST NIL))))))
19500: (SETQ KRT (APPEND J1 KRT D1))
19600: (COND ((NEQ J1 NIL) (SETQ J1 (CAR J1))))
19700: (SETQ KQ (APPEND KQ (LIST KRT)))
19800: (SETQ KW (LIST F KQ))
19900: (SETQ KWW (APPEND KWW (LIST KW)))
20000: (COND ((EQ EFLAG NIL)
20100: (COND ((NULL JIKUGIRI-GO)
20200: (SETQ HFLAG-J T)
20300: (COND ((NULL KYODO) (SETQ HFLAG-K T
) (SETQ KRT (LIST KAZE)))
20400: (T (SETQ HFLAG-K NIL) (SETQ KR
T (LIST KAZE DOAI))))))
20500: (T (SETQ HFLAG-J NIL)
20600: (COND ((NULL KYODO) (SETQ HFLAG-K
T) (SETQ KRT (LIST JIKUGIRI-GO KAZE)))
20700: (T (SETQ HFLAG-K NIL) (SETQ
KRT (LIST JIKUGIRI-GO KAZE DOAI))))))
20800: (T (COND ((NULL JIKUGIRI-GO) (SETQ JIKUGIRI-
GO (FN2 JIKAN-TAI-GO 1 2))))

```

```

20900: (COND ((NULL KYODO) (SETQ DOAI NIL)))
21000: (SETQ P (LIST CHO))
21100: (SETQ KRT (LIST JINUGIRI-GO KAZE DOAI))
21200: (SETQ KQ (APPEND KQ (LIST KRT)))
21300: (SETQ KW (LIST P KQ))
21400: (SETQ KWW (APPEND KWW (LIST KW))))))
21500: (COND ((EQ EFLAG T) (SETQ ABCD-IN KWW) (SETQ ABCD-OUT (TRANS KWW)) (
RETURN T))
21600: (T (PRINT "TRY ONCE MORE") (GO TAG))))
21700:
21800: (DEFUN MOVE NIL
21900: (PROG NIL
22000: (COND ((NULL X) (PRINT "ERROR-1 : NO WORD IS FOUND.") (RETURN NIL))
22100: (T (SETQ * (CAR X) X (CHR X)) (RETURN T))))))
22200:
22300: (DEFUN MEMB (* ENSEMBLE)
22400: (COMMENT MEM IS MEMBER)
22500: (COND ((NULL ENSEMBLE) NIL)
22600: (T (COND ((EQUAL * (CAR ENSEMBLE)) T) (T (MEMB * (CDR ENSEMBLE)))
))))
22700:
22800: (DEFUN PTP (S V1 V2)
22900: (PROG NIL
23000: (PUTPROP S (LIST V1) 'HINSHI)
23100: (PUTPROP V2 (LIST V1) 'HINSHI)
23200: (PUTPROP S V2 'IR)
23300: (PUTPROP V2 S 'IR)
23400: (RETURN T)))
23500:
23600: (DEFUN PREPARE NIL
23700: (PROG (FLAG)
23800: (SETQ KANA K ROME R)
23900: (COND ((EQ KANA 1)
24000: (SETQ ROME 2)
24100: (PRINT
24200: "CHARACTER MODE IS SET TO KANA-MOJI. IF YOU WANT TO TAKE RO
MA-JI MODE , THEN EVALUATE THE FUNCTION PREPARE AGAIN AFTER SETTING R TO 1 AND
K TO 2.))
24300: (SETQ FLAG T))
24400: ((EQ ROME 1)
24500: (SETQ KANA 2)
24600: (PRINT
24700: "CHARACTER MODE IS SET TO ROMA-JI. IF YOU WANT TO TAKE KANA
-MOJI MODE , THEN EVALUATE THE FUNCTION PREPARE AGAIN AFTER SETTING K TO 1 AND
R TO 2.))
24800: (SETQ FLAG T))
24900: ((EQ KANA 2)
25000: (SETQ ROME 1)
25100: (PRINT
25200: "CHARACTER MODE IS SET TO ROMA-JI. IF YOU WANT TO TAKE KANA
-MOJI MODE , THEN EVALUATE THE FUNCTION PREPARE AGAIN AFTER SETTING K TO 1 AND
R TO 2.))
25300: (SETQ FLAG T))
25400: ((EQ ROME 2)
25500: (SETQ KANA 1)
25600: (PRINT
25700: "CHARACTER MODE IS SET TO KANA-MOJI. IF YOU WANT TO TAKE R
OMA-JI MODE , THEN EVALUATE THE FUNCTION PREPARE AGAIN AFTER SETTING R TO 1 AN
D K TO 2.))
25800: (SETQ FLAG T))
25900: (T (PRINT
26000: "CHARACTER MODE IS NOT SET NEITHER KANA-MOJI NOR ROMA-JI.
SET K TO 1 IF YOU WANT TO INPUT SENTENCES IN KANA-MOJI. OR SET R TO 1 IF YOU WA
NT TO INPUT THEM IN ROMA-JI. AFTER DOING IT , EVALUATE THE FUNCTION PREPARE AG
AIN!"))
26100: (SETQ FLAG NIL)))

```

(注)関数MOVE---解析のポインタを次の語に進める関数。

関数MEMB---第一引数が第二引数(リスト)の要素か否かを判定する関数

関数PTP---辞書に登録された語彙(素形記号)をその内部記号と関連付ける関数

関数PREPARE---ユーザーが予め指定した使用文字モード(片カナ文字モード、ローマ文字モード)のたろくを行ない片カナ文字用もしくはローマ文字用の辞書を用意し、その旨のメッセージを出力する関数

```

26200: (COND ((EQ FLAG T) (DICTIONARY))
26300: (T (PRINT
26400: ' A FUNCTION TO PREPARE THE DICTIONARY FOR PARSING HAS NOT
YET EVALUATED. *)
(RTURN NIL)))
26500: (SETQ PRINTLEVEL 0 PRINTLENGTH 0)
26600: (RETURN T)))
26700: (RETURN T)))
26800:
26900: (DEFUN DICTIONARY NIL
27000: (FROG NIL
27100: (COND ((EQ KANA 1)
27200: (KANA-MONJI)
27300: (PRINT "カタカナ" / "ローマ字" "カタカナ" "ローマ字" "カタカナ" "ローマ字"
. *)
(RTURN T))
27400: ((EQ ROMA 1)
27500: (ROMA-MONJI)
27600: (PRINT
27700: "ROMA-MOJI NO JISHO O JYUNBI SHIMASHITA NODE ROMA-JI DE N
YUURIKU SHITE KUDASAI. *)
(RTURN T))
28000: (T (PRINT
28100: "MOJI-MODE IS NOT SET . AND SOMETHING IS WRONG! BUGS ARE I
N THIS SYSTEM !!!")
(RTURN NIL))))))
28200: (DEFUN KANA-MONJI NIL
28300: (FROG NIL
28400: (FTP 'アウツテ) 'JIKAN-TAI-GO' 'JT00)
28500: (FTP 'アウツテ) 'JIKUGIRI-GO' 'JG00)
28600: (FTP 'アウ) 'JIKAN-TAI-GO' 'JT05)
28700: (FTP 'アウ) 'JIKUGIRI-GO' 'JG05)
28800: (FTP 'アウ) 'TENKOU-SO' 'TS00)
28900: (FTP 'イフシ) 'SETSUZOKU-GO' 'SZ00)
29000: (FTP 'イフシ) 'JIKUGIRI-GO' 'JG03)
29100: (FTP 'イフシ) 'CHIIKI-MEI' 'CHOA)
29200: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM00)
29300: (FTP 'イフシ) 'JIKUGIRI-GO' 'JG05)
29400: (FTP 'イフシ) 'SETSUZOKU-GO' 'SZ03)
29500: (FTP 'イフシ) 'STGA)
29600: (FTP 'イフシ) 'CHIIKI-MEI' 'CH0B)
29700: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM05)
29800: (FTP 'イフシ) 'KARA)
29900: (FTP 'イフシ) 'KAZE-SO' 'KS00)
30000: (FTP 'イフシ) 'KAZE-SO' 'KS05)
30100: (FTP 'イフシ) 'JIKAN-TAI-GO' 'JT10)
30200: (FTP 'イフシ) 'TENKOU-SO' 'TS05)
30300: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM10)
30400: (FTP 'イフシ) 'JIKUGIRI-GO' 'JG10)
30500: (FTP 'イフシ) 'JIKUGIRI-GO' 'JG15)
30600: (FTP 'イフシ) 'JIKAN-TAI-GO' 'JT15)
30700: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM15)
30800: (FTP 'イフシ) 'CHIIKI-MEI' 'CH03)
30900: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM20)
31000: (FTP 'イフシ) 'CHIIKI-MEI' 'CH01)
31100: (FTP 'イフシ) 'CHIIKI-MEI' 'CH03)
31200: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM25)
31300: (FTP 'イフシ) 'VERB' 'TSUY)
31400: (FTP 'イフシ) 'VERB' 'TSUI)
31500: (FTP 'イフシ) 'KEIZOKU-GO' 'KZ00)
31600: (FTP 'イフシ) 'SHUURYOU-GO' 'SR00)
31700: (FTP 'イフシ) 'KEIZOKU-GO' 'KZ05)
31800: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM30)
31900: (FTP 'イフシ) 'SETSUZOKU-GO' 'SZ05)
32000: (FTP 'イフシ) 'CHIIKI-MEI' 'CH03)
32100: (FTP 'イフシ) 'TOFUKEN-MEI' 'TM35)
32200: (FTP 'イフシ) 'SHUURYOU-GO' 'SR05)
32300: (FTP 'イフシ) 'SHUURYOU-GO' 'SR05)

```

(注) 関数 DICTIONARY --- 関数 PREPARE の機能上の本体をなしている関数。ユーザーが予め指定した使用文字モード(片カナ文字モード、ローマ文字モード)に応じて片カナ文字用辞書とローマ文字用の辞書を使用し、その旨のメッセージを出力する。但し、関数 PREPARE のように使用文字モードが指定されているか否かのチェックをすることは行わない。

関数 KANA-MONJI --- 片カナ文字用辞書を使用可能な状態にする関数

- 32400: (PTP 'フクシマ' 'FUKU=SHI' 'FKTM)
- 32500: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS10)
- 32600: (PTP 'カゼノカサ' '*KAZE-SO' 'KS15)
- 32700: (PTP 'チイノチ' '*CHIINI-MEI' 'CM05)。
- 32800: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS20)
- 32900: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS25)
- 33000: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG20)
- 33100: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG20)
- 33200: (PTP 'テンノウチ' '*TENKOU-SO' 'TS10)
- 33300: (PTP 'テンノウチ' '*TENKOU-SO' 'TS15)
- 33400: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG25)
- 33500: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG27)
- 33600: (PTP 'テンノウチ' '*TENKOU-SO' 'TS20)
- 33700: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS30)
- 33800: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS35)
- 33900: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG30)
- 34000: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG35)
- 34100: (PTP 'チイノチ' '*CHIINI-MEI' 'CM03)。
- 34200: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS40)
- 34300: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS45)
- 34400: (PTP 'チイノチ' '*CHIINI-MEI' 'CM10)。
- 34500: (PTP 'チイノチ' '*CHIINI-MEI' 'CM15)。
- 34600: (PTP 'セツスゾク' '*SETSUZOKU-GO' 'SZ10)
- 34700: (PTP 'チイノチ' '*CHIINI-MEI' 'CM20)。
- 34800: (PTP 'トフクノチ' '*TOFUKEN-MEI' 'TM40)
- 34900: (PTP 'ドアイ' '*DOAI' 'DA00)
- 35000: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS50)
- 35100: (PTP 'カゼイノカサ' '*KAZE-SO' 'KS55)
- 35200: (PTP 'テンノウチ' '*TENKOU-SO' 'TS25)
- 35300: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG40)
- 35400: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG45)
- 35500: (PTP 'ジクウノチ' '*JIKUGIRI-GO' 'JG50)
- 35600: (PTP 'キゴウ' '*KIGOU' 'TEN)
- 35700: (PTP 'キゴウ' '*KIGOU' 'MARU)
- 35800: (RETURN T))
- 35900:
- 36000: (DEFUN ROMA-MONJI NIL
- 36100: (FROG NIL
- 36200: (PTP 'アメ' '*TENKOU-SO' 'TS00)
- 36300: (PTP 'ASA=NO=UCHI' '*JIKUGIRI-GO' 'JG00)
- 36400: (PTP 'ASATE=WA' '*JIKAN-TAI-GO' 'JT00)
- 36500: (PTP 'ASU=WA' '*JIKAN-TAI-GO' 'JT05)
- 36600: (PTP 'IBARAGI=KEN' '*TOKEN=MEI' 'TM00)
- 36700: (PTP 'ICHIJI' '*SETSUZOKU-GO' 'SZ00)
- 36800: (PTP 'ICHINICHI=JYUU' '*JIKUGIRI-GO' 'JG03)
- 36900: (PTP 'IFFANCHI=DEWA' '*CHIINI-MEI' 'CM0A)
- 37000: (PTP 'OSOKU=MATTE' '*JIKUGIRI-GO' 'JG05)
- 37100: (PTP 'GA' '*STGA' 'STGA)
- 37200: (PTP 'KA' '*SETSUZOKU-GO' 'SZNA)
- 37300: (PTP 'KANAGAWA=KEN' '*TOFUKEN-MEI' 'TM05)
- 37400: (PTP 'KITA=NO=KAZE' '*KAZE-SO' 'KS00)
- 37500: (PTP 'KITAYORI=NO=KAZE' '*KAZE-SO' 'KS05)
- 37600: (PTP 'KYOU=WA' '*JIKAN-TAI-GO' 'JT10)
- 37700: (PTP 'KUMORI' '*TENKOU-SO' 'TS05)
- 37800: (PTP 'GUNMA=KEN' '*TOFUKEN-MEI' 'TM10)
- 37900: (PTP 'GOGO' '*JIKUGIRI-GO' 'JG10)
- 38000: (PTP 'GOGO=WA' '*JIKUGIRI-GO' 'JG15)
- 38100: (PTP 'KONTA=WA' '*JIKAN-TAI-GO' 'JT15)
- 38200: (PTP 'SAITAMA=KEN' '*TOFUKEN-MEI' 'TM15)
- 38300: (PTP 'SAPPANBU=DEWA' '*CHIINI-MEI' 'CM00)
- 38400: (PTP 'SHIZUOKA=KEN' '*TOFUKEN-MEI' 'TM20)
- 38500: (PTP 'ZENINI=DE' '*CHIINI-MEI' 'CM01)
- 38600: (PTP 'CHIBA=KEN' '*TOFUKEN-MEI' 'TM25)
- 38700: (PTP 'CHICHIBU=CHIHON=DEWA' '*CHIINI-MEI' 'CM03)
- 38800: (PTP 'STUYOKU' '*VERB' 'TSUY)
- 38900: (PTP 'DE' '*KEIZOKU-GO' 'KZ00)

} 60
 280
 30

} 80
 280
 36

(注) 関数 ROMA-MONJI --- ロ-マ文字用辞彙を使用可能な状態にする関数


```

39000: (PTP 'DESHIQU ''SHUURYOU-GO' 'SR00)
39100: (PTP 'DESUGA ''KEIZOKU-GO' 'KZ05)
39200: (PTP 'TOKIDOKI ''SETSUZOKU-GO' 'SZ05)
39300: (PTP 'TOKORO=NI=YORI ''CHIIKI-MEI' 'CM03)
39400: (PTP 'TOCHIGI=KEN ''TOFUKEN-MEI' 'TM35)
39500: (PTP 'TOMONI ''FUKU-SHI' 'FKTH)
39600: (PTP 'TO=NARU=DESHIQU ''SHUURYOU-GO' 'SR05)
39700: (PTP 'NANSEI=NO=KAZE ''KAZE-SO' 'KS10)
39800: (PTP 'NANTOU=NO=KAZE ''KAZE-SO' 'KS15)
39900: (PTP 'NANBU=DEWA ''CHIIKI-MEI' 'CM05)
40000: (PTP 'NISHI=NO=KAZE ''KAZE-SO' 'KS20)
40100: (PTP 'NISHIYORI=NO=KAZE ''KAZE-SO' 'KS25)
40200: (PTP 'NITCHYUU ''JIKUGIRI-GO' 'JG20)
40300: (PTP 'NITCHYUU=WA ''JIKUGIRI-GO' 'JG20)
40400: (PTP 'NIWAKA=AME ''TENKOU-SO' 'TS10)
40500: (PTP 'NIWAKA=YUKI ''TENKOU-SO' 'TS15)
40600: (PTP 'NOCHI ''JIKUGIRI-GO' 'JG25)
40700: (PTP 'HAJIME=NO=UCHI ''JIKUGIRI-GO' 'JG27)
40800: (PTP 'HARE ''TENKOU-SO' 'TS20)
40900: (PTP 'HIGASHI=NO=KAZE ''KAZE-SO' 'KS30)
41000: (PTP 'HIGASHIYORI=NO=KAZE ''KAZE-SO' 'KS35)
41100: (PTP 'HIRUGORO=KARA ''JIKUGIRI-GO' 'JG30)
41200: (PTP 'HIRUMAE=KARA ''JIKUGIRI-GO' 'JG35)
41300: (PTP 'HEIYABU=DEWA ''CHIIKI-MEI' 'CM08)
41400: (PTP 'HOKUSEI=NO=KAZE ''KAZE-SO' 'KS40)
41500: (PTP 'HOKUTOU=NO=KAZE ''KAZE-SO' 'KS45)
41600: (PTP 'HOKURU=DEWA ''CHIIKI-MEI' 'CM10)
41700: (PTP 'HOKURU=YAMAZOI=DEWA ''CHIIKI-MEI' 'CM15)
41800: (PTP 'MATAWA ''SETSUZOKU-GO' 'SZ10)
41900: (PTP 'YAMAZOI=DEWA ''CHIIKI-MEI' 'CM20)
42000: (PTP 'YAMANASHI=KEN ''TOFUKEN-MEI' 'TM40)
42100: (PTP 'YAYA ''DOAI' 'DA00)
42200: (PTP 'MINAMI=NO=KAZE ''KAZE-SO' 'KS50)
42300: (PTP 'MINAMIYORI=NO=KAZE ''KAZE-SO' 'KS55)
42400: (PTP 'YUKI ''TENKOU-SO' 'TS25)
42500: (PTP 'YOI=NO=UCHI ''JIKUGIRI-GO' 'JG40)
42600: (PTP 'YOI=NO=UCHI=WA ''JIKUGIRI-GO' 'JG45)
42700: (PTP 'YOKU=WA ''JIKUGIRI-GO' 'JG50)
42800: (PTP '/, ''KIGOU' 'FIRI)
42900: (RETURN T)))
43000:
43100: (DEFUN FN1 (TOFUKEN-MEI CH)
43200: (COND ((EQ CH 'CM00) 'CM08) ((EQ CH 'CM20) 'CM08) ((EQ CH 2) 'CM01) (T
'CM0A)))
43300:
43400: (DEFUN FN2 (JIKAN-TAI-GO J2 JO)
43500: (PROG (X)
43600: (SETQ X (LIST J2 JO))
43700: (COND ((EQ JIKAN-TAI-GO 'JT10)
43800: (COND ((EQUAL X '(1 2)) (RETURN 'JG03))
43900: ((EQUAL X '(1 JG25)) (RETURN 'JG27))
44000: ((EQUAL X '(1 JG20)) (RETURN 'JG00))
44100: (T (RETURN '??1))))))
44200: ((EQ JIKAN-TAI-GO 'JT15)
44300: (COND ((EQUAL X '(1 2)) (RETURN 'JG40))
44400: ((EQUAL X '(1 JG25)) (RETURN 'JG27))
44500: (T (RETURN '??2))))))
44600: ((EQ JIKAN-TAI-GO 'JT05)
44700: (COND ((EQUAL X '(1 2)) (RETURN 'JG03))
44800: ((EQUAL X '(1 JG25)) (RETURN 'JG27))
44900: ((EQUAL X '(1 JG20)) (RETURN 'JG00))
45000: (T (RETURN '??3))))))
45100: ((EQ JIKAN-TAI-GO 'JT00)
45200: (COND ((EQUAL X '(1 2)) (RETURN 'JG03))
45300: ((EQUAL X '(1 JG25)) (RETURN 'JG27))
45400: ((EQUAL X '(1 JG20)) (RETURN 'JG00))))))

```

(注) 関数 FN1 --- 未知の地域名語を文脈から推測・決定する関数
 関数 FN2 --- 未知の時区切語を文脈から推測・決定する関数

```

45500:
45600: (DEFUN TRANS (X)
45700:   (COND ((NULL X) NIL)
45800:         ((ATOM (CAR X)) (CONS (GET (CAR X) 'IR) (TRANS (CDR X))))
45900:         (T (CONS (TRANS (CAR X)) (TRANS (CDR X)))))
46000:
46100: (DEFUN TKN NIL
46200:   (COMMENT TKN IS 'TENKOU-SETSU KAISEKI KANSUU')
46300:   (PROG (* JO J1 J2 FLAG HFLAG EFLAG XX JIKUGIRI-GO CHIIKI-MEI TOKO TENKO
U-SO-1ST
46400:     TENKOU-SO-2ND SETSUZOKU-GO TENKOU RT Q P WW W CHO)
46500:     (SETQ J2 1 J1 1 JO 1)
46600:     (SETQ FLAG T)
46700:     (SETQ XX (TRANS X))
46800:     (PRINT XX)
46900:   TAG (COND ((NULL X)
47000:             (PRINT 'ERROR-2 : THERE IS NO SENTENCE TO BE PARSED')
47100:             (PRINT 'AT CHECK POINT 1')
47200:             (RETURN NIL)))
47300:   (MOVE)
47400:   (COND ((MEMB 'CHIIKI-MEI' (GET * 'HINSHI))
47500:         (SETQ CHIIKI-MEI *)
47600:         (COND ((EQ * 'CM03)
47700:               (SETQ TOKO (LIST *))
47800:               (PRINT 'TOKORO-NI-YORI IS FOUND')
47900:               (CHIIKI-MEI NIL)))
48000:         (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT 2') (RETURN NIL)))
48100:         (T (SETQ CHIIKI-MEI NIL)))
48200:   (COND ((MEMB 'JIKUGIRI-GO' (GET * 'HINSHI))
48300:         (SETQ JIKUGIRI-GO *)
48400:         (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT 3') (RETURN NIL)))
48500:         (T (SETQ JIKUGIRI-GO NIL)))
48600:   (COND ((EQ * 'CM03)
48700:         (PRINT 'TOKORO-NI-YORI IS PERHAPS FOUND.')
48800:         (COND ((AND (NULL TOKO) (OR (NEQ CHIIKI-MEI NIL) (NEQ JIKUGIRI-
I-GO NIL))))
48900:         (SETQ TOKO (LIST *))
49000:         (PRINT 'TOKORO-NI-YORI IS EXACTLY FOUND.')
49100:         (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT 4.') (RETURN
N NIL))))
49200:   (T (PRINT 'ERROR-3 : ') (PRINT 'AT CHECK POINT 5.') (RETURN
TURN NIL))))
49300:   (SETQ J2 J1 J1 JO JO JIKUGIRI-GO)
49400:   (COND ((MEMB 'TENKOU-SO' (GET * 'HINSHI))
49500:         (SETQ TENKOU-SO-1ST *)
49600:         (PRINT 'THE CANDIDATE FOR TENKOU-SO-1ST IS FOUND.')
49700:         (SETQ FLAG-TS1 T)
49800:         (COND ((NOT (MOVE))
49900:               (SETQ * 'SROO)
50000:               (PRINT 'WARNING-1 : DESIDOU IS MISSING.')
50100:               (PRINT 'NOTE : DESIDOU IS ASSUMPTED.')))
50200:         (T (COND ((EQ FLAG T)
50300:                  (PRINT 'ERROR-4 : TENKOU IS MISSING .')
50400:                  (PRINT 'AT CHECK POINT 6')
50500:                  (PRINT 'THIS SENTENCE IS NOT A WEATHER FORECAST ONE!
50600:                  ')
50600:                  (RETURN NIL))
50700:         (T (SETQ FLAG-TS1 NIL) (PRINT 'TENKOU-SO-1ST IS IMPLI
CITLY FOUND.'))))
50800:   (COND ((MEMB 'SETSUZOKU-GO' (GET * 'HINSHI))
50900:         (SETQ SETSUZOKU-GO *)
51000:         (COND ((NOT (MOVE)) (PRINT 'AT CHECK POINT 7.') (RETURN NIL))
51100:         (COND ((MEMB 'TENKOU-SO' (GET * 'HINSHI))

```

(注)関数TRANS --- リスト内の全てのATOMを、表層文字から内部記号へもしくはその逆へ変換する関数
関数TKK --- 天候節新関数(Tenkousetsu Kaiseki Kansuu)

```

51200: (SETQ TENKOU-SO-2ND *)
51300: (PRINT "<TENKOU-KAI> SUCCESS")
51400: (SETQ <TENKOU-KAI> T <TENKOU-SO> NIL)
51500: (COND ((NOT (MOVE))
51600: (SETQ * 'SROO)
51700: (PRINT "WARNING-1 : DESHIYOU IS MISSING.")
51800: (PRINT "DESIYOU IS ASSUMPTED.")
51900: (SETQ EFLAG T)))
52000: (T (PRINT "ERROR-5 : THE 2ND TENKOU-SO IS MISSING.") (R
ETURN NIL))))
52100: (T (COND ((EQ FLAG-TS1 T)
52200: (SETQ <TENKOU-KAI> NIL <TENKOU-SO> T)
52300: (PRINT "<TENKOU-KAI> FAIL")
52400: (PRINT "<TENKOU-SO> SUCCESS!"))
52500: (T (PRINT "ERROR : INVALID SYNTAX ") (RETURN NIL))))))
52600: (COND ((EQ <TENKOU-KAI> T)
52700: (SETQ TENKOU (LIST TENKOU-SO-1ST SETSUZOKU-GO TENKOU-SO-2ND))
)
52800: ((EQ <TENKOU-SO> T) (SETQ TENKOU TENKOU-SO-1ST))
52900: (T (PRINT "ERROR-5 : ")))
53000: (SETQ TENKOU (APPEND TOKO (LIST TENKOU)))
53100: (PRINT "TENKOU COMPLETED :")
53200: (PRINT (TRANS TENKOU))
53300: (SETQ TOKO NIL)
53400: (COND ((MEMB "NEIZOKU-GO" (GET * 'HINSHI))
53500: (SETQ EFLAG NIL)
53600: (COND ((NOT (MOVE))
53700: (PRINT "WARNING-2 : THIS SENTENCE IS NOT COMPLETE.")
53800: (PRINT "AT CHECK POINT 8.")
53900: (RETURN NIL))))))
54000: (COND ((MEMB "SHUURYOU-GO" (GET * 'HINSHI)) (SETQ EFLAG T)))
54100: (COND ((EQ FLAG T)
54200: (COND ((EQ EFLAG NIL)
54300: (SETQ CHO CHIKI-MEI)
54400: (COND ((EQ JIKUGIRI-GO NIL) (SETQ RT TENKOU) (SETQ HFL
AG T)
54500: (T (SETQ RT (LIST JIKUGIRI-GO TENKOU)) (SETQ HFL
AG NIL))))
54600: (SETQ FLAG NIL))
54700: (T (COND ((NULL CHIKI-MEI) (SETQ CHO (FN1 TOFUKEN-MEI
1)))
54800: (T (SETQ CHO CHIKI-MEI))))
54900: (SETQ P (LIST CHO))
55000: (COND ((NULL JIKUGIRI-GO) (SETQ JIKUGIRI-GO (FN2 JIK
AN-TAI-GO J1 2))))
55100: (SETQ RT (LIST JIKUGIRI-GO TENKOU))
55200: (SETQ Q (APPEND Q (LIST RT)))
55300: (SETQ W (LIST P Q))
55400: (SETQ WW (APPEND WW (LIST W)))
55500: (SETQ Q NIL))))
55600: (T (COND ((NEQ CHIKI-MEI NIL)
55700: (COND ((NULL CHO) (SETQ CHO (FN1 TOFUKEN-MEI CHIKI-
MEI))))
55800: (SETQ P (LIST CHO))
55900: (SETQ CHO CHIKI-MEI)
56000: (COND ((EQ HFLAG T) (SETQ J1 (FN2 JIKAN-TAI-GO J2 2)
) (SETQ RT (LIST J1 RT))))
56100: (SETQ Q (APPEND Q (LIST RT)))
56200: (SETQ W (LIST P Q))
56300: (SETQ WW (APPEND WW (LIST W)))
56400: (SETQ Q NIL)
56500: (COND ((EQ EFLAG NIL)
56600: (COND ((NULL JIKUGIRI-GO) (SETQ RT TENKOU) (S
ETQ HFLAG T)
56700: (T (SETQ RT (LIST JIKUGIRI-GO TENKOU))
(SETQ HFLAG NIL))))))

```

```

56800: (T (SETQ P (LIST CHO))
56900: (COND ((NULL JIKUGIRI-GO) (SETQ JIKUGIRI-GO
(FN2 JIKAN-TAI-GO 1 2))))
57000: (SETQ RT (LIST JIKUGIRI-GO TENKOU))
57100: (SETQ Q (APPEND Q (LIST RT)))
57200: (SETQ W (LIST P Q))
57300: (SETQ WW (APPEND WW (LIST W)))
57400: (SETQ Q NIL)))
57500: (T (COND ((NULL JIKUGIRI-GO)
57600: (PRINT "ERROR-5 : CHIIKI-MEI OR JIKUGIRI-G
0 IS MISSING."))
57700: (PRINT "AT CHECK POINT 9."))
57800: (COND ((EQ HFLAG T) (SETQ J1 (FN2 JIKAN-TAI-GO J2
J0)) (SETQ RT (LIST J1 RT))))
57900: (SETQ Q (APPEND Q (LIST RT)))
58000: (SETQ RT (LIST JIKUGIRI-GO TENKOU))
58100: (SETQ HFLAG NIL)
58200: (COND ((EQ EFLAG T)
58300: (COND ((NULL CHO) (SETQ CHO (FN1 TOFUKEN-ME
I 1))))
58400: (SETQ P (LIST CHO))
58500: (SETQ Q (APPEND Q (LIST RT)))
58600: (SETQ W (LIST P Q))
58700: (SETQ WW (APPEND WW (LIST W)))
58800: (SETQ Q NIL))))))
58900: (COND ((EQ EFLAG T)
59000: (SETQ ABC-IN WW)
59100: (SETQ ABC-OUT (TRANS WW))
59200: (PRINT "THE SENTENCE")
59300: (PRINT X-SHOW)
59400: (PRINT "IS ACCEPTED AS A WEATHER FORECAST ONE.")
59500: (PRINT "AND ITS KNOWLEDGE REPRESENTATION IS AS THE FOLLOWING
: ")
59600: (SETQ KNOW (LIST TOFUKEN-MEI JIKAN-TAI-GO (LIST ABC-IN ABC-I
N)))
59700: (PRINT (TRANS KNOW))
59800: (RETURN T))
59900: (T (PRINT "TRY ONCE MORE") (SETQ X (CONS * X)) (GO TAG))))))
60000:
60100:(DEFUN GF1 (WW)
60200: (COND ((EQUAL (CDR WW) NIL)
60300: (COND ((EQUAL FLAG NIL) (SETQ FLAG T) (GF2 WW))
60400: (T (APPEND (LIST 'KZ00) (GF2 WW)))))
60500: (T (SETQ FLAG T) (APPEND (GF2 WW) (GF1 (CDR WW)))))
60600:
60700:(DEFUN GF11 (WW)
60800: (COND ((EQUAL (CDR WW) NIL)
60900: (SETQ W-FLAG T)
61000: (COND ((EQUAL FLAG NIL) (SETQ FLAG T) (APPEND (GF22 WW) '(SROO)
'(MARU))))
61100: (T (APPEND (LIST 'KZ05 'TEN) (GF22 WW) '(SROO) '(MARU))))))
61200: (T (SETQ W-FLAG NIL) (SETQ FLAG T) (APPEND (GF22 WW) (GF11 (CDR W
W)))))
61300:
61400:(DEFUN GF2 (WW) (APPEND (CAAR WW) (GF (CADAR WW))))
61500:
61600:(DEFUN GF22 (WW) (APPEND (CAAR WW) (GFN1-1 (CADAR WW))))
61700:
61800:(DEFUN GFO (KNOW)
61900: (PROG (FLAG)
62000: (SETQ FLAG NIL)
62100: (SETQ WW (CADR (CADR KNOW)))
62200: (RETURN (TRANS (GF1 WW)))))
62300:
62400:(DEFUN GFOO (KNOW)
62500: (PROG (FLAG)

```

(注) 関数GF1---関数GFOの補助関数
 関数GF11---関数GFOOの補助関数
 関数GF2---関数GF1の補助関数
 関数GF22---関数GF11の補助関数

関数GFO---"で"や"でしょう"を補うだけの天気文生成関数
 関数GFOO---"で"や"でしょう"を補うとともに簡約化を行
 行なう天気文生成関数。天気文生成関数の
 の一応の最終版也。

```

62600: (SETQ FLAG NIL)
62700: (SETQ WW (CADR (CADR KNOW)))
62800: (RETURN (TRANS (GF11 WW))))
62900:
63000: (DEFUN QF (Q)
63100: (COND ((EQUAL (CDR Q) NIL) (CAR Q)) (T (APPEND (CAR Q) (QF (CDR Q)))))
63200:
63300: (DEFUN HELP1-RED2 (RT1 RT2)
63400: (PROG (L2 FAT1 FAT2 QQ)
63500: (SETQ QQ (LIST RT1 RT2))
63600: (SETQ FLAG NIL)
63700: (SETQ FAT1 (CAR (LIST-2ND RT1)))
63800: (SETQ L2 (LENGTH (LIST-2ND RT2)))
63900: (COND ((EQ L2 1)
64000: (SETQ FAT2 (CAR (LIST-2ND RT2)))
64100: (COND ((ATOM FAT1)
64200: (COND ((NOT (ATOM FAT2))
64300: (COND ((EQUAL (CAR FAT2) FAT1)
64400: (SETQ FLAG T)
64500: (SETQ QQ (LIST RT1 (APPEND (LIST (CAR RT
2) (LIST (CDR FAT2)) (CDR RT2)))))))
64600: (RETURN QQ))
64700: (T (COND ((NOT (ATOM FAT2))
64800: (COND ((EQUAL (CAR FAT1) (CAR FAT2))
64900: (SETQ FLAG T)
65000: (SETQ QQ (LIST RT1 (APPEND (LIST (CAR
RT2) (LIST (CDR FAT2)) (CDR RT2)))))))
65100: (RETURN QQ))))
65200: ((EQ L2 2)
65300: (SETQ FAT2 (CADR (LIST-2ND RT2)))
65400: (COND ((ATOM FAT1)
65500: (COND ((NOT (ATOM FAT2))
65600: (COND ((EQUAL (CAR FAT2) FAT1)
65700: (SETQ FLAG T)
65800: (SETQ QQ (LIST RT1 (APPEND (LIST (CAR RT
2) (LIST (CDR FAT2)) (CDR RT2)))))))
65900: (RETURN QQ))
66000: (T (COND ((NOT (ATOM FAT2))
66100: (COND ((EQUAL (CAR FAT1) (CAR FAT2))
66200: (SETQ FLAG T)
66300: (SETQ QQ (LIST RT1
66400: (APPEND (LIST (CAR RT2)
(LIST (CAR (LIST-2ND RT2)) (CDR FAT2)) (CDR RT2)))))))
66500: (RETURN QQ))))))
66600:
66700: (DEFUN REDUCE2 (RT1 RT2)
66800: (PROG (L1)
66900: (SETQ L1 (LENGTH (LIST-2ND RT1)))
67000: (COND ((EQ L1 1) (RETURN (HELP1-RED2 RT1 RT2)))
67100: ((EQ L1 3) (RETURN (HELP1-RED2 RT1 RT2)))
67200: (T (PRINT "ERROR"))))
67300:
67400: (DEFUN LIST-2ND (RT)
67500: (COND ((NULL RT) (PRINT "ERROR : RT IS NIL"))
67600: ((EQ (LENGTH RT) 1) (PRINT "ERROR : RT IS ONE-LIST LONG ."))
67700: (T (CADR RT)))
67800:
67900: (DEFUN PRETTY (X)
68000: (COND ((NULL X) NIL)
68100: ((ATOM (CAR X)) (CONS (CAR X) (PRETTY (CDR X))))
68200: (T (APPEND (PRETTY (CAR X)) (PRETTY (CDR X)))))
68300:
68400: (DEFUN GFN1 (Q)
68500: (COND ((NULL Q) NIL)
68600: ((NULL (CDR Q)) (PRETTY (CAR Q)))
68700: ((NULL (CDR Q)) (PRETTY (NSF2 (REDUCE2 (CAR Q) (CDR Q))))))

```

(注) 関数 QF --- 関数 GF2 の補助関数
 関数 HELP1-RED2 --- 関数 REDUCE2 の
 補助関数。REDUCE2 の機能上の本体。
 関数 REDUCE2 --- “文表現レベルでの簡約化”を
 行なう関数

関数 LIST-2ND --- リストの第二要素を値として返す
 関数
 関数 PRETTY --- リストのカッコの深さを1にする関数
 関数 GFN1 ----- 簡約化を行なう文生成関数

```

68000: (T (SETQ QQ (REDUCE2 (CAR Q) (CADR Q)))
68900: (COND ((EQ FLAG NIL) (PRETTY Q))
69000: (T (PRETTY (APPEND QQ (CDR (REDUCE2 (CAR Q) (CADR Q))))
))))))
69100:
69200: (DEFUN GFN1-1 (Q)
69300: (COND ((NULL Q) NIL)
69400: ((NULL (CDR Q)) (APPEND (PRETTY (CAR Q))))
69500: ((NULL (CDR Q)) (PRETTY (BSF2 (REDUCE2 (CAR Q) (CADR Q))))))
69600: (T (SETQ QQ (REDUCE2 (CAR Q) (CADR Q)))
69700: (COND ((EQ FLAG NIL) (PRETTY Q))
69800: (T (PRETTY (APPEND QQ (CDR (REDUCE2 (CAR Q) (CADR Q))))
))))))
69900:
70000: (DEFUN BSF1 (Q)
70100: (COND ((NULL Q) NIL)
70200: (T (COND ((NULL (CDR Q)) (APPEND (CAR Q) '(SROO)))
70300: (T (APPEND (CAR Q) '(TEN) (BSF1 (CDR Q)))))))
70400:
70500: (DEFUN BSF2 (Q)
70600: (COND ((NULL Q) (PRINT 'ERROR))
70700: (T (COND ((NULL (CDR Q)) (CAR Q))
70800: (T (COND ((NULL (CDR Q))
70900: (COND ((EQ W-FLAG T) (APPEND (CAR Q) '(KZOO) (
CADR Q))))
71000: (T (APPEND (CAR Q) '(KZOO) (CADR Q))))))
71100: (T (APPEND (CAR Q) '(TEN) (BSF2 (CDR Q)))))))
)
71200:
71300: (DEFUN SO (KNOW) (PRETTY (TRANS (S (CADR KNOW))))))
71400:
71500: (DEFUN SSO (KNOW) (PRETTY (TRANS (SS (CADR KNOW))))))
71600:
71700: (DEFUN S (KWW)
71800: (COND ((NULL (CDR KWW))
71900: (APPEND (APPEND (CAAR KWW) (F (CADAR KWW))) (LIST 'SROO 'MARU)))
72000: (T (APPEND (APPEND (CAAR KWW) (F (CADAR KWW))) (S (CDR KWW))))))
72100:
72200: (DEFUN SS (KWW)
72300: (COND ((NULL (CDR KWW))
72400: (APPEND (APPEND (CAAR KWW) (PROD (CADAR KWW))) (LIST 'SROO 'MARU
)))
72500: (T (APPEND (APPEND (CAAR KWW) (PROD (CADAR KWW))) (SS (CDR KWW))
))))
72600:
72700: (DEFUN F (KQ)
72800: (COND ((NULL (CDR KQ)) (SETQ EFLAG T) (G (CAR KQ)))
72900: (T (SETQ EFLAG NIL) (APPEND (G (CAR KQ)) (F (CDR KQ))))))
73000:
73100: (DEFUN G (KRT)
73200: (COND ((EQ (CADR KRT) 'DAOO)
73300: (COND ((NULL EFLAG) (SETQ * 'TSUY)) (T (SETQ * 'TSUI)))
73400: (LIST (CAR KRT) (CADR KRT) 'STGA 'DAOO *))
73500: ((EQ (CADR KRT) 'KARA)
73600: (COND ((NULL EFLAG) (SETQ * 'TSUY)) (T (SETQ * 'TSUI)))
73700: (LIST (CAR KRT) (CADR KRT) 'STGA *))
73800: ((EQ (CADR KRT) NIL)
73900: (COND ((NULL EFLAG) (LIST (CAR KRT) (CADR KRT) 'KZOO))
74000: (T (LIST (CAR KRT) (CADR KRT))))))
74100: (T (PRINT
74200: "ERROR-??? : THE KNOWLEDGE REPRESENTATION INPUTTED IS SYNTA
CTICALLY INVALID!"))))
74300:
74400: (DEFUN BSFN1 (Q-IN) (TRANS (BSF1 Q-IN)))
74500:
74600: (DEFUN BSFN2 (Q-IN) (TRANS (BSF2 Q-IN)))

```

(主) 関数 GFN1-1 --- 簡納化を行なう天候文生成関数
 関数 BSF1 --- 常に語点を打つという方針の天候文生成関数
 関数 BSF2 --- 語点, "で"を適宜挿入する方針の天候文生成関数
 関数 BSFN1 --- 関数 BSF1の補助関数
 関数 BSFN2 --- 関数 BSF2の補助関数

関数 SO --- "で"や"では"を補うための風文生成関数
 関数 SSO --- "で"や"では"を補うとともに簡納化を行なう風文生成関数。風文生成関数の一応の最終版もの。
 関数 S --- 関数 SOの補助関数
 関数 SS --- 関数 SSOの補助関数
 関数 F --- 関数 Sの補助関数
 関数 G --- 関数 Fの補助関数

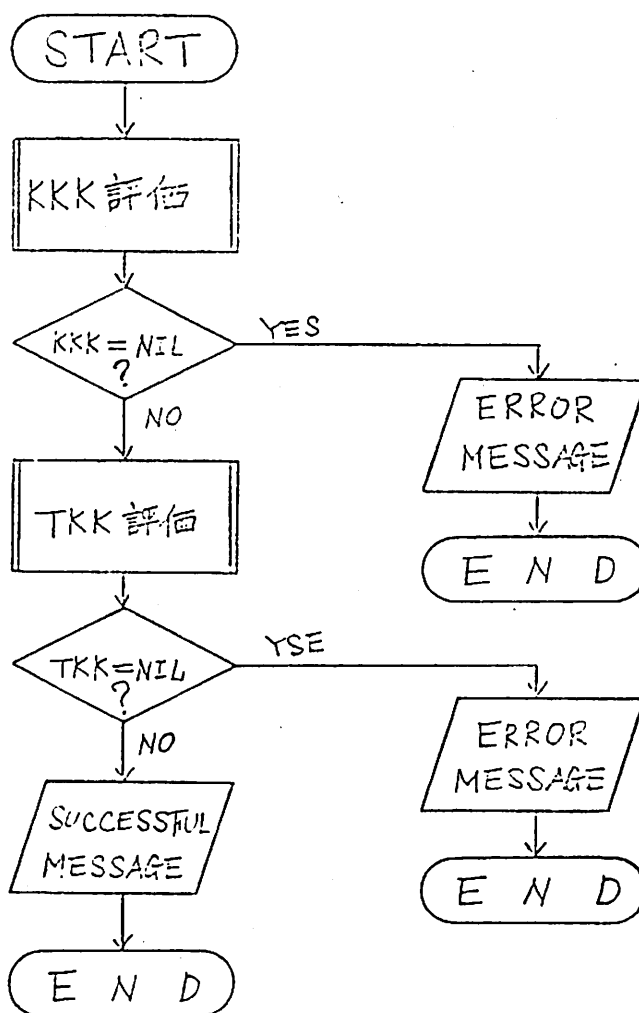
```

74700:
74800:(DEFUN LAST-CUT (NRT) (PROG (HOLD) (HELP-LC NRT) (RETURN HOLD)))
74900:
75000:(DEFUN HELP-LC (NRT)
75100:  (COND ((NULL NRT) NIL)
75200:  ((ATOM NRT) (PRINT "ERROR : ARGUMENT IS NOT A LIST BUT AN ATOM !!
*))
75300:  ((NULL (CDR NRT)) NIL)
75400:  (T (SETQ HOLD (APPEND HOLD (LIST (CAR NRT)))) (HELP-LC (CDR NRT)
)))
75500:
75600:(DEFUN LIST-3RD (NRT)
75700:  (COND ((ATOM NRT) (PRINT "ERROR : ARGUMENT IS NOT A LIST BUT AN ATOM !!
*))
75800:  ((NULL NRT) (PRINT "LIST IS NIL."))
75900:  ((NULL (CDR NRT)) (PRINT "ERROR : LIST-LENGTH OF THE LIST IS ONE.
*))
76000:  ((NULL (CDDR NRT)) (PRINT "ERROR : LIST-LENGTH OF THE LIST IS TWO
.*"))
76100:  (T (CADDR NRT))))
76200:
76300:(DEFUN RED2-K (NRT1 NRT2)
76400:  (COND ((EQUAL (LIST-3RD NRT1) (LIST-3RD NRT2))
76500:  (COND ((EQUAL (LIST-3RD NRT1) NIL) (SETQ FLAG-TOMONI NIL) (LIST
NRT1 NRT2))
76600:  (T (SETQ FLAG-TOMONI T)
76700:  (LIST (LIST (LAST-CUT NRT1) (LAST-CUT NRT2)) (LIST-3RD
NRT1))))))
76800:  (T (SETQ FLAG-TOMONI NIL) (LIST NRT1 NRT2))))
76900:
77000:(DEFUN PROD (KQ)
77100:  (COND ((NULL KQ) (PRINT "ERROR : PROD-1"))
77200:  ((NULL (CDR KQ)) (F KQ))
77300:  ((NULL (CDDR KQ))
77400:  (SETQ KQQ (REDUCE2 (CAR KQ) (CADR KQ)))
77500:  (FF (RED2-K (CAR KQQ) (CADR KQQ))))
77600:  (T (PRINT "ERROR : OVER POWER-PROD *))))
77700:
77800:(DEFUN FF (KQQQ) (COND ((EQ FLAG-TOMONI T) (FFF KQQQ)) (T (F KQQQ))))
77900:
78000:(DEFUN FFF (KQQQ) (LIST (PRETTY (CAR KQQQ)) 'STGA 'FKTM (CADR KQQQ) 'TSUY)
)
78100:
78200:(DEFUN カ NIL (PROG NIL (SETQ K 1 R 2) (PREPARE) (RETURN T)))
78300:
78400:(DEFUN ROUMA NIL (PROG NIL (SETQ K 2 R 1) (PREPARE) (RETURN T)))
78500:
JET45200I END OF DATA
E:END S
JET45355I SAVED IN DATA SET ('A5775.KAMEDAS.VDATA')
>>LOGOFF
JET11061I USER COMMAND PROFILE BEING STORED
JSN=F24509,CT=01.84,ET=00:30:12,MEMORY=04KB,OUT=OF,FILE=(ONR:10,1425KR:12),
ACCT=(1109,140299,19701).
JET10080I A5775 TSS SESSION ENDED TIME=17:45:31 DATE=82-03-15+

```

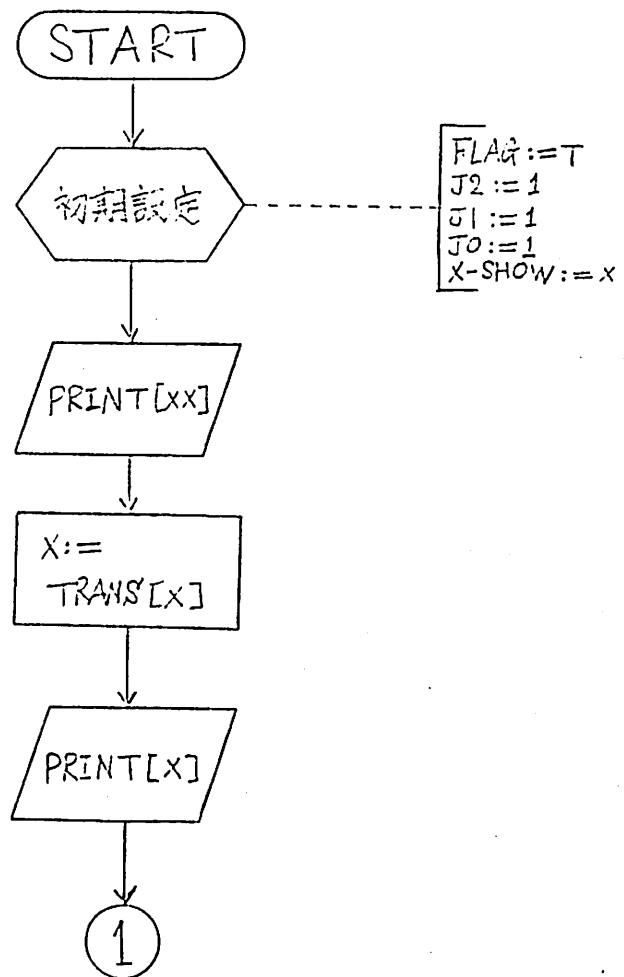
(③) 関数 LAST-CUT --- リストの最右端要素を取り
除く関数
関数 HELP-LC ---- 関数 LAST-CUT の本体
関数 LIST-3RD --- リストの第三要素を値として
返す関数
関数 RED2-K --- 風文において論理的簡約
を行なう為の関数

関数 PROD --- 簡約化を行なう風文生成関数
関数 FF --- 関数 PROD の補助関数
関数 FFF --- 関数 FF の補助関数
関数 カ --- 使用文字モードを片カナに指定し、片
カナ用特字を用いる関数
関数 ROUMA --- 使用文字モードをローマ字に指定し、
ローマ文字用特字を用いる関数



(注) 天気予報文解析関数PFは、風速解析関数KKKと
 天候解析関数TKKとから成っている。TKKとKKK
 とは、ほとんど同一の動作をするが、KKKの動作の方が
 やや複雑である。KKKの動作の流れ図が右図8.に
 示してある。

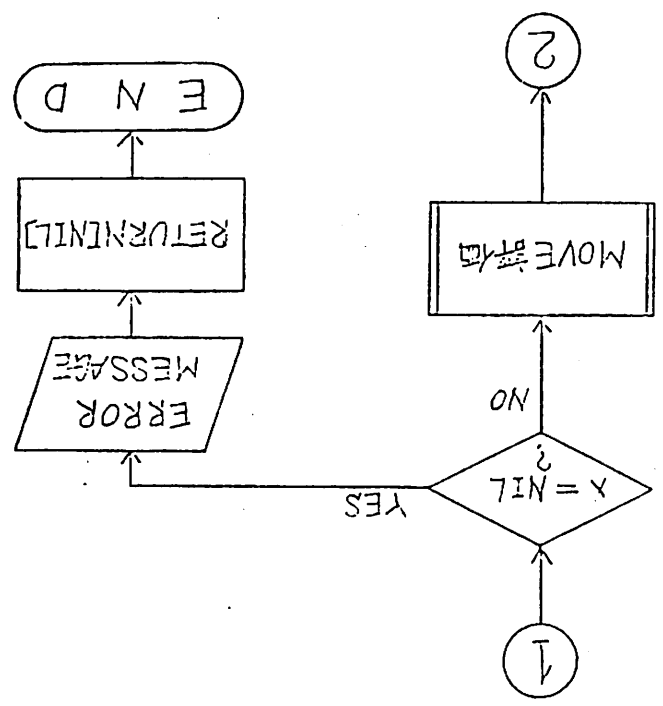
図A-7. 天気予報文解析関数PFの動作の流れ図

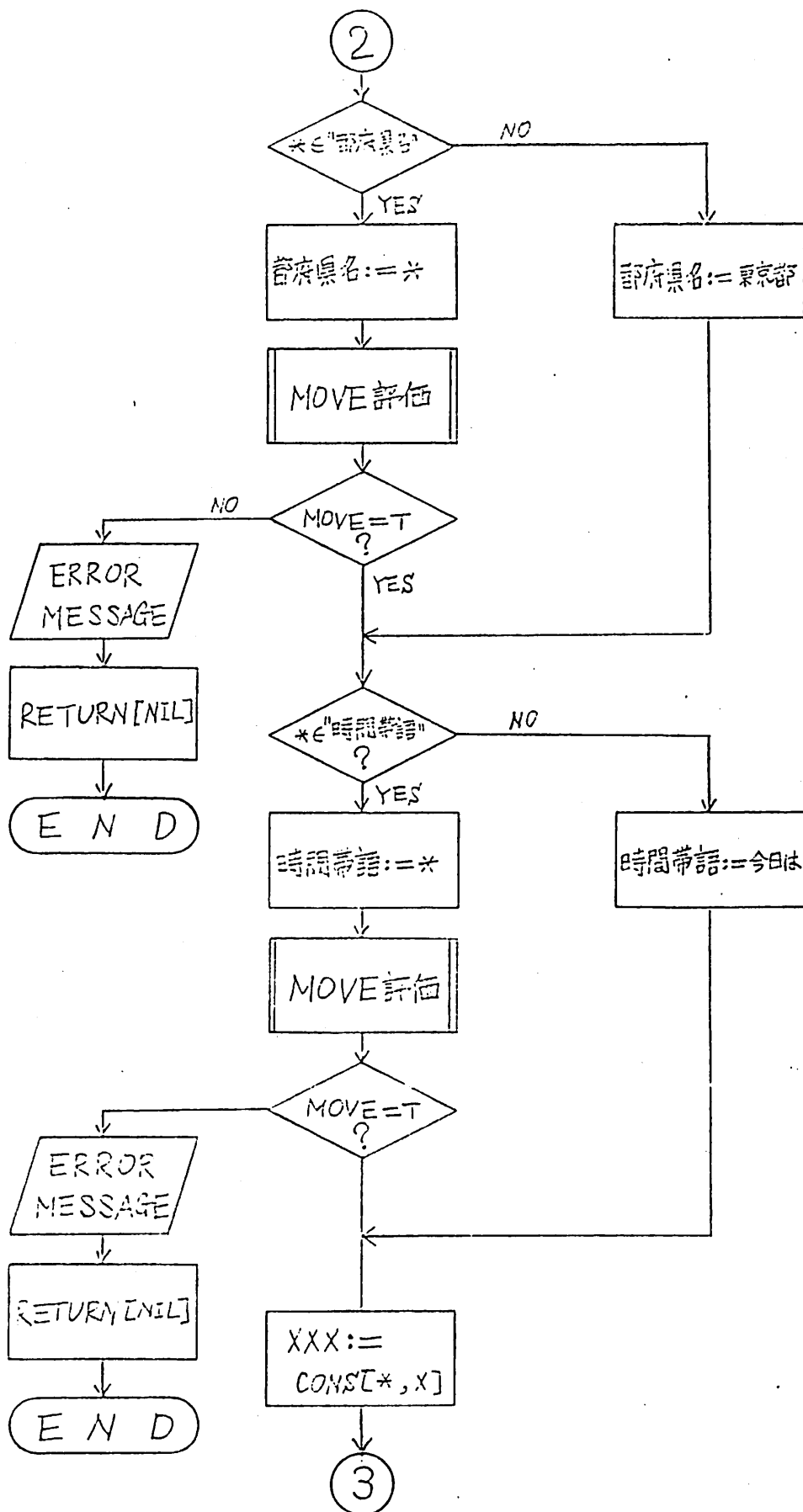


(注) 付録 8 (65頁 ~ 87頁) は、風節解析関数 KKK の流れ図であるが、強度情報関係の動作を無視すれば、そのまま天候節解析関数 TKK の流れ図である (但し、ほんの一部の書き換えあり)。

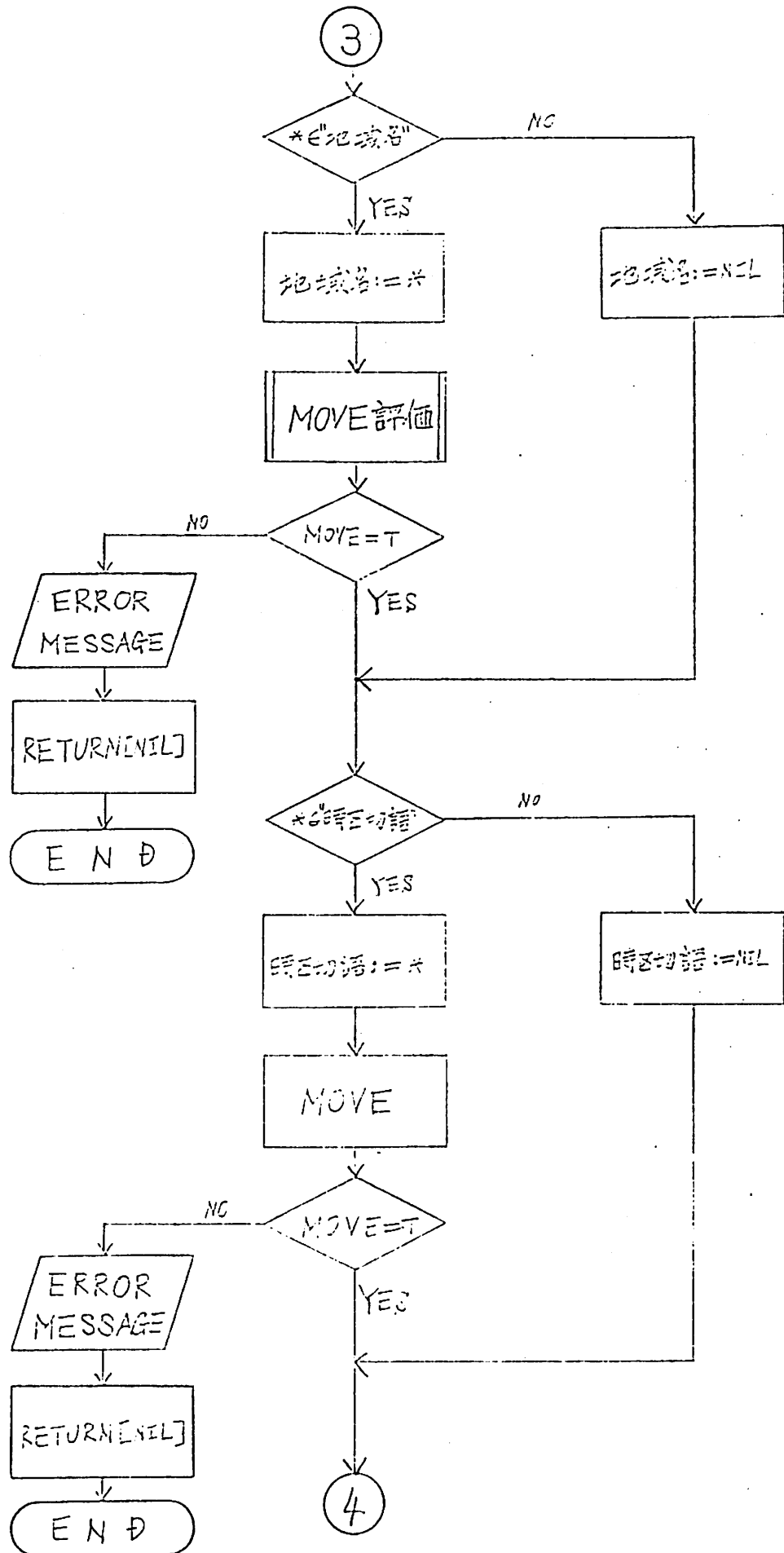
図A-8. 風節解析関数 KKK の動作の流れ図

XA-8. (C) (≠)



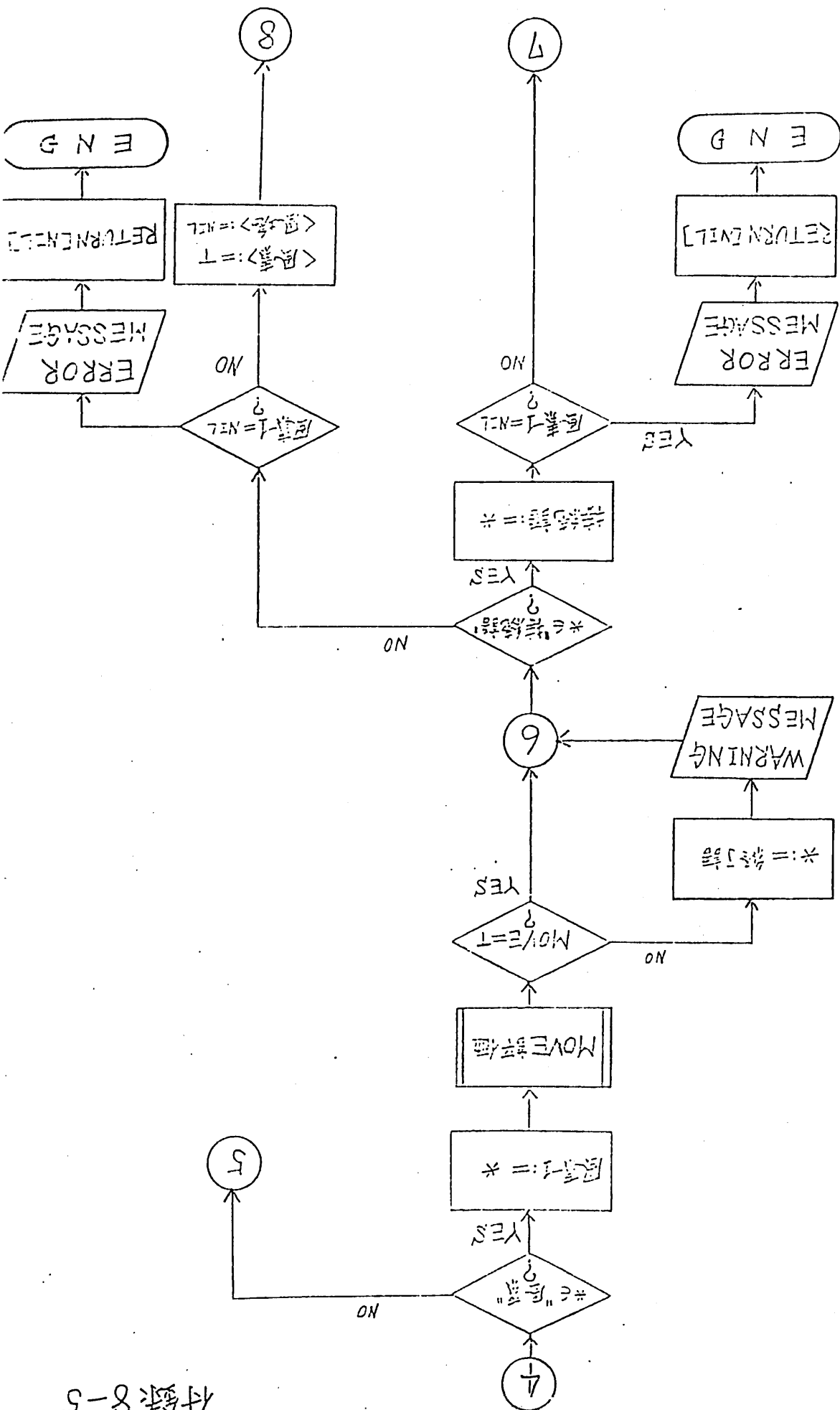


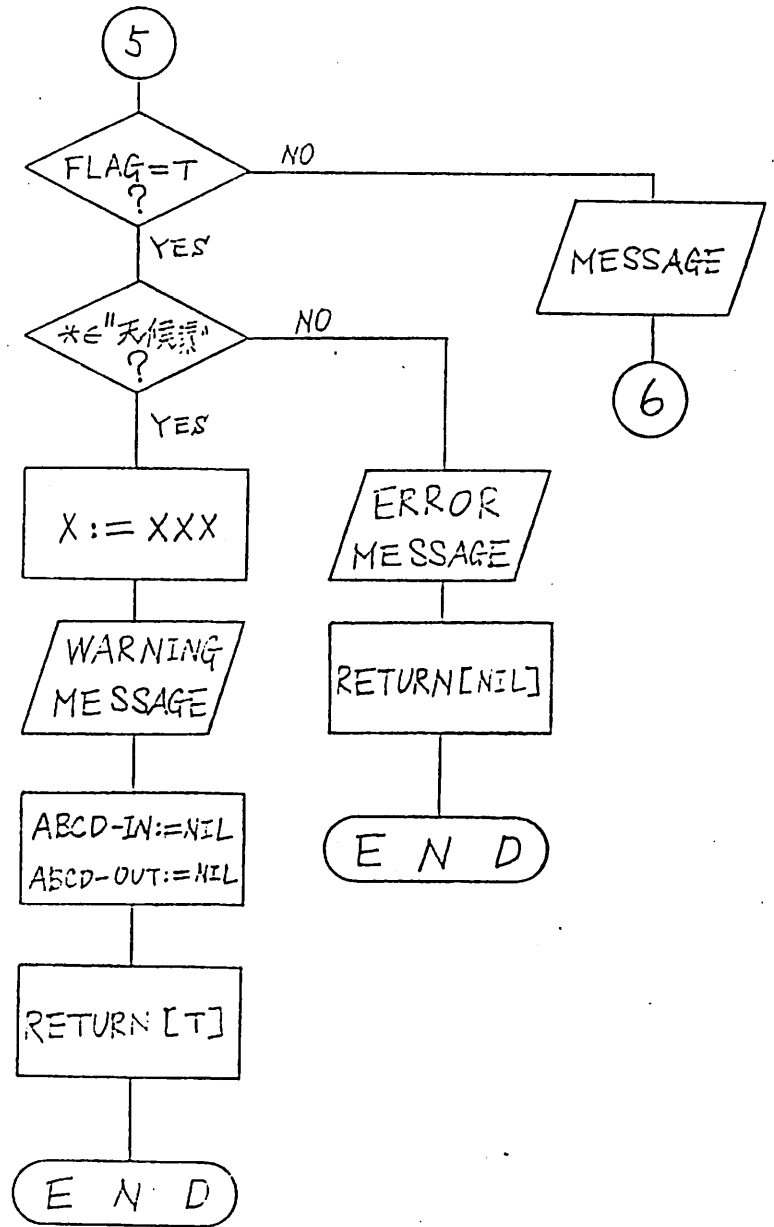
図A-8. (つづき)



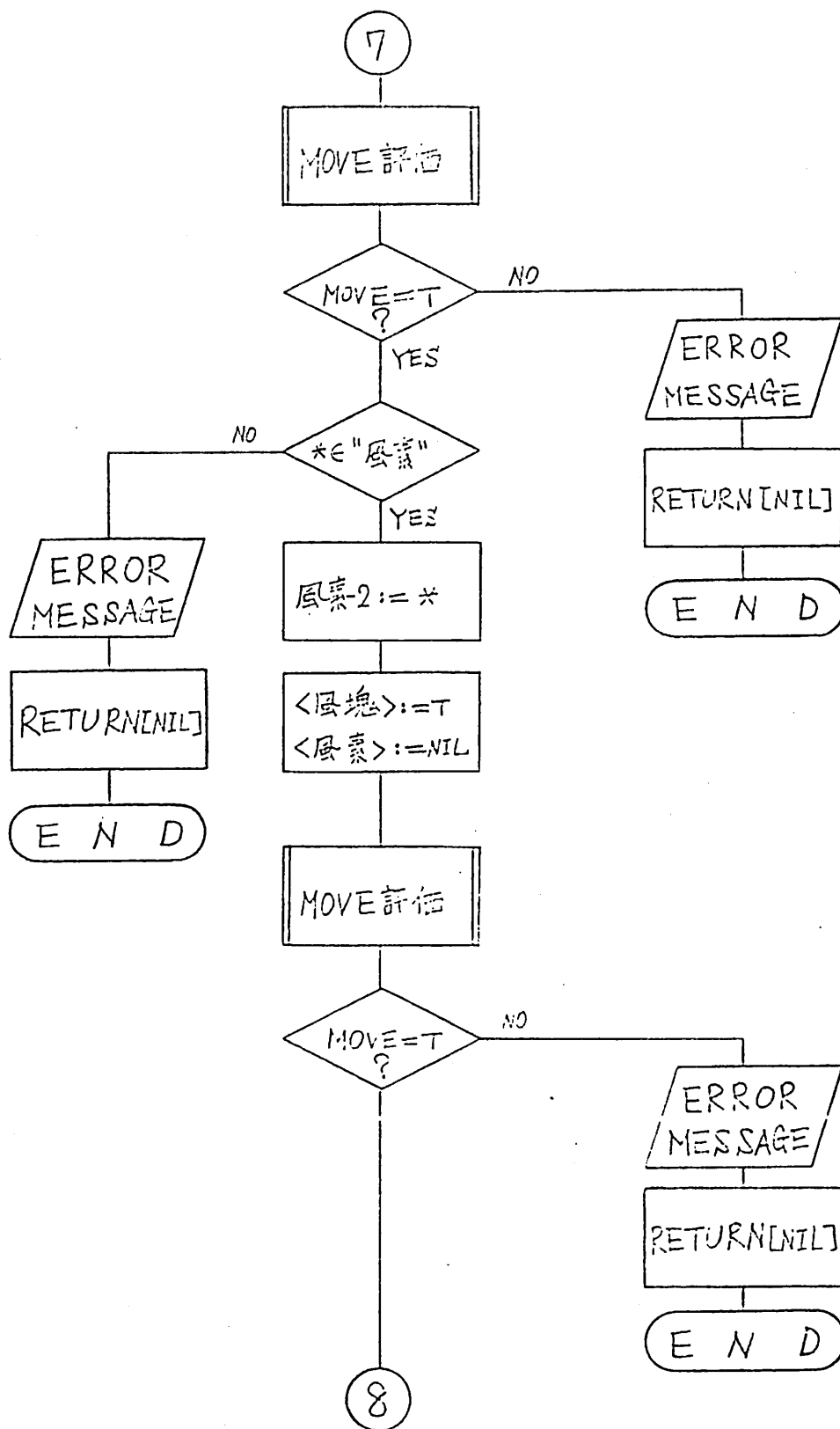
図A-8. (つづき)

☒A-8. (77"7)





図A-8. (つづき)



図A-8. (つづき)

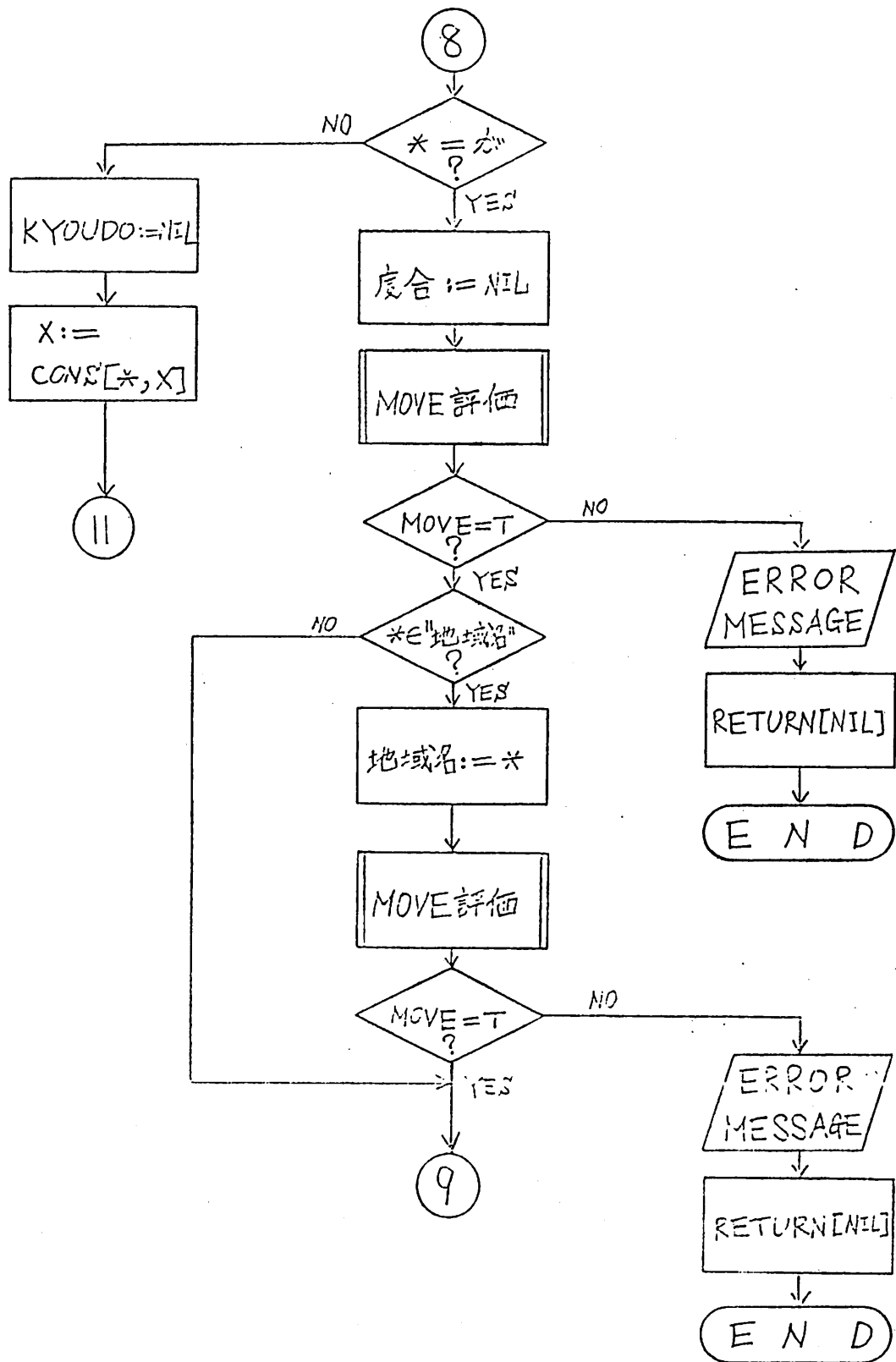


図 A-8. (つづき)

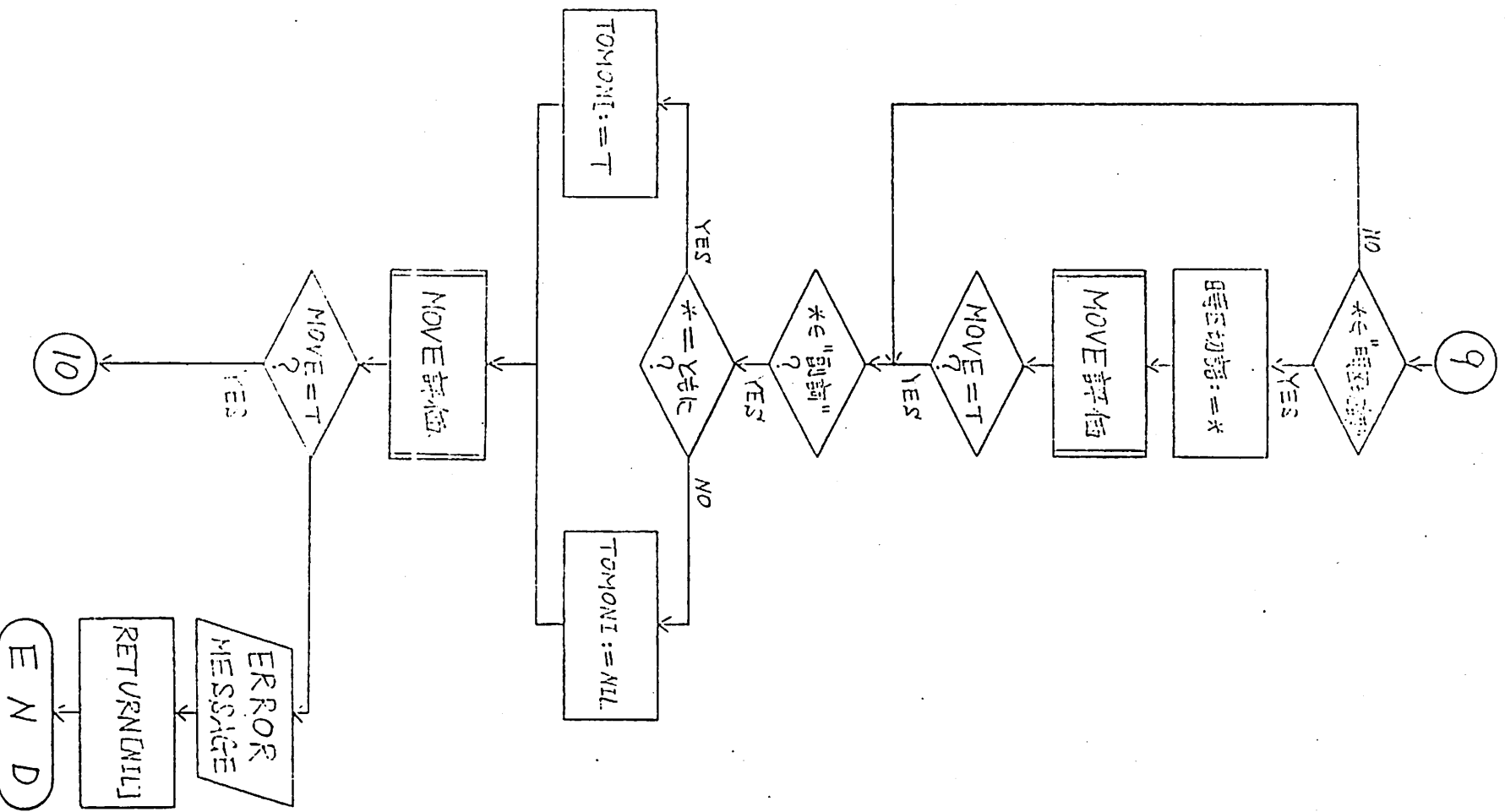
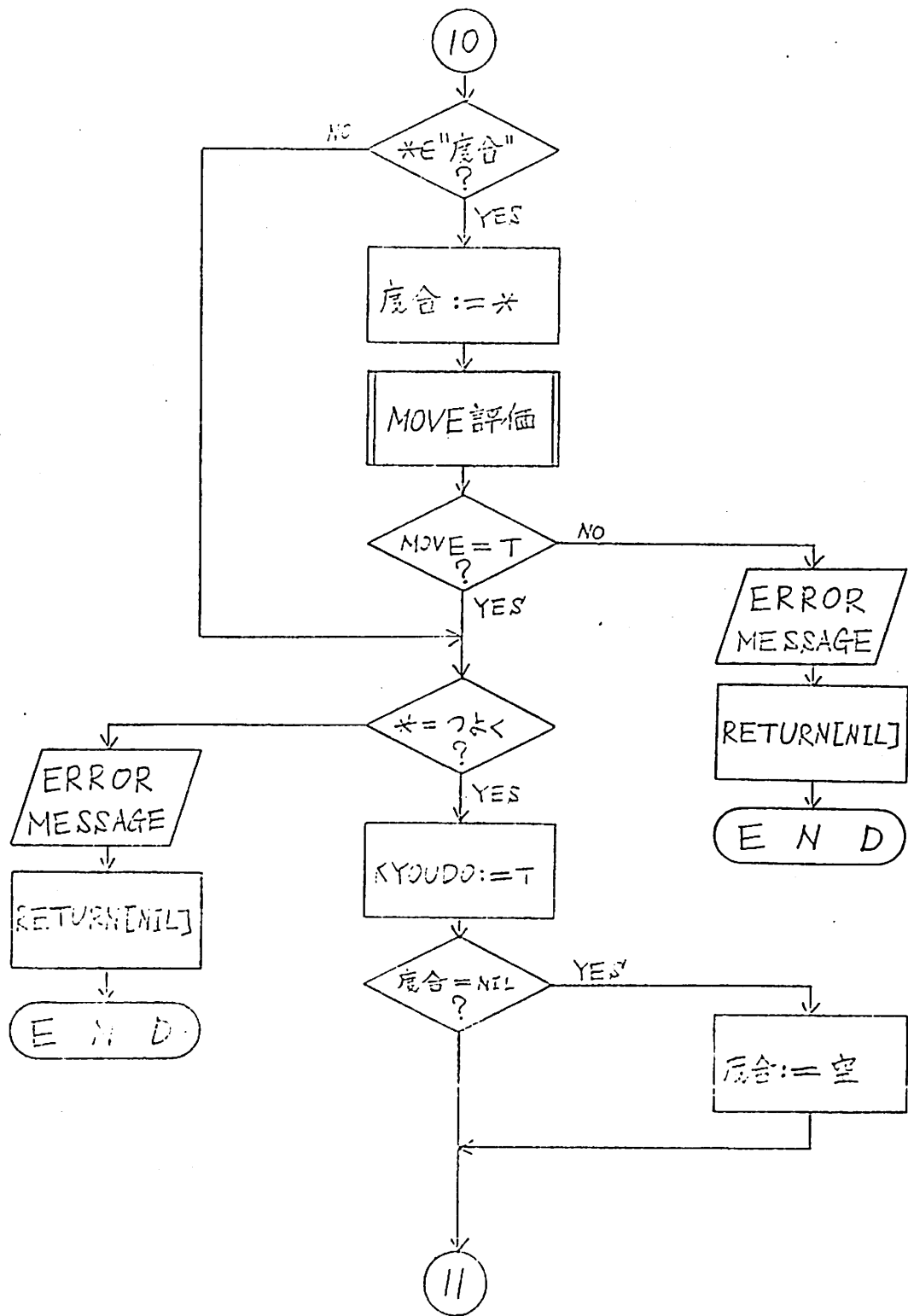
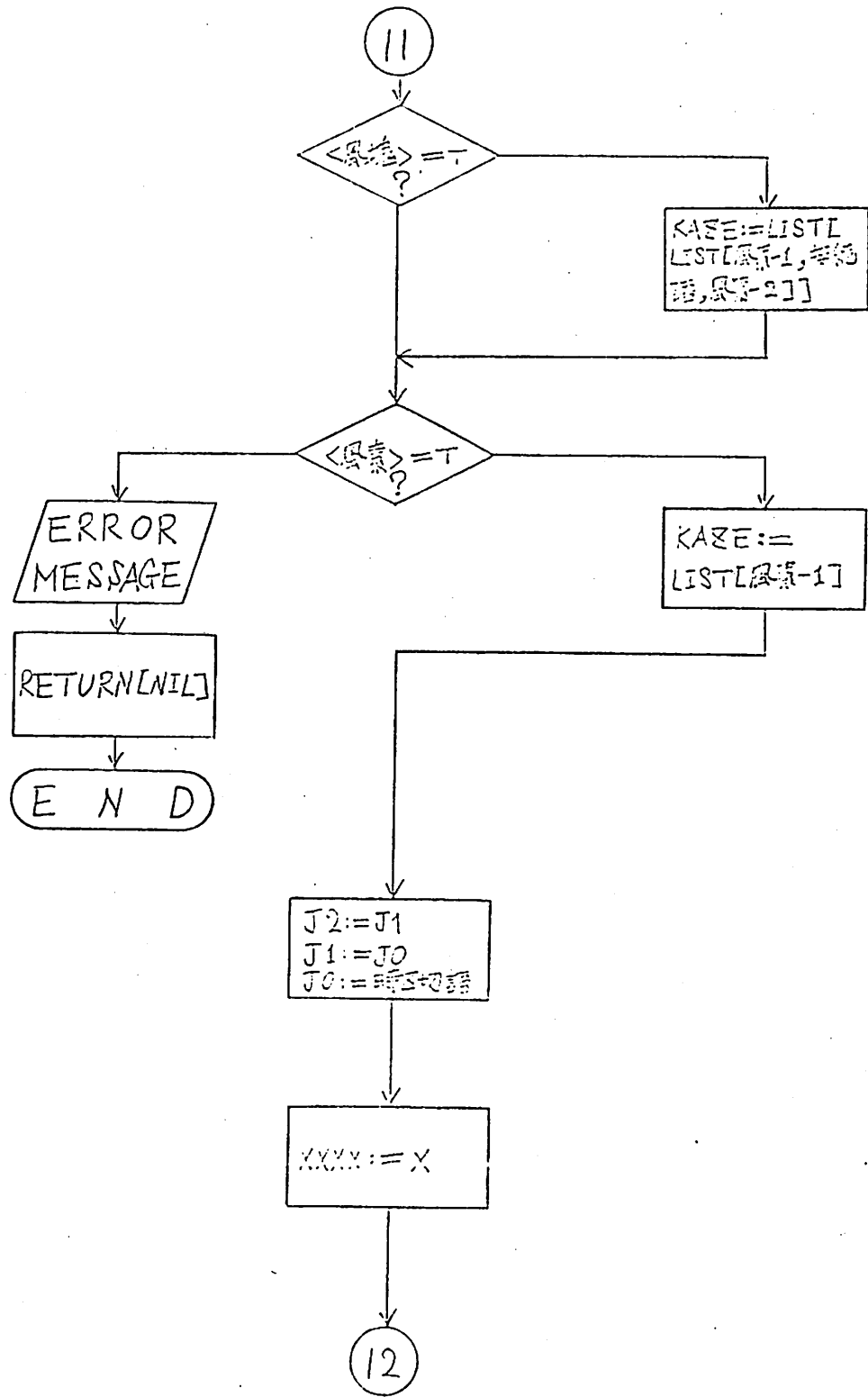


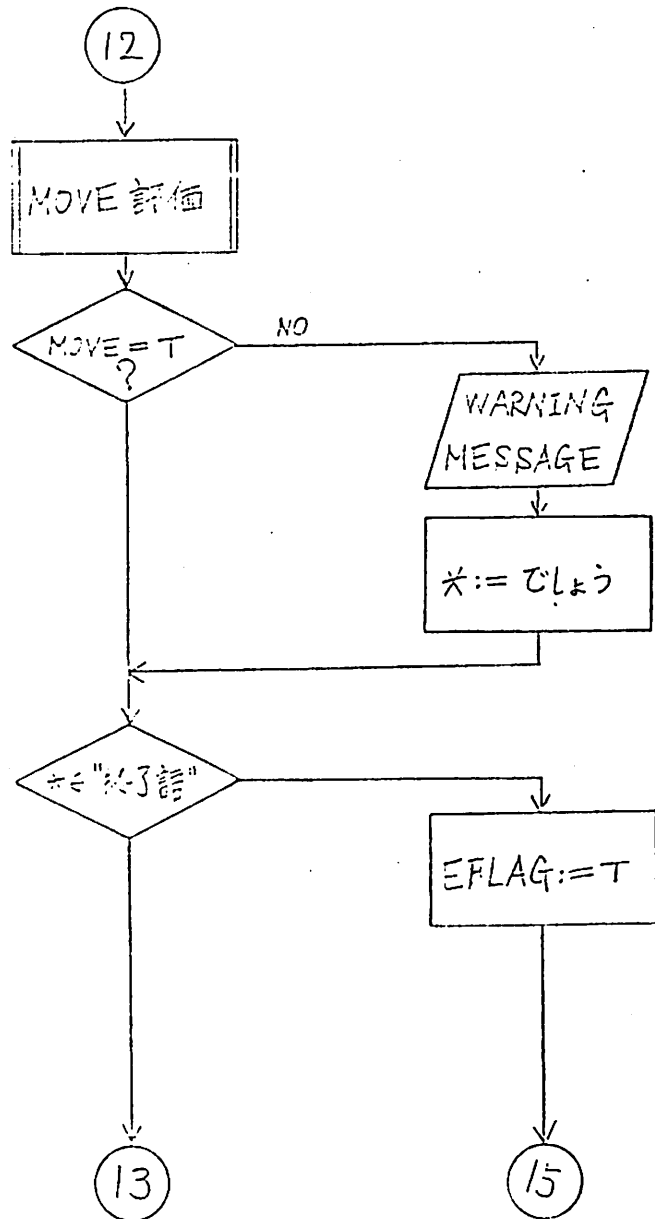
図 A-8. (つづ)



図A-8. (つづき)

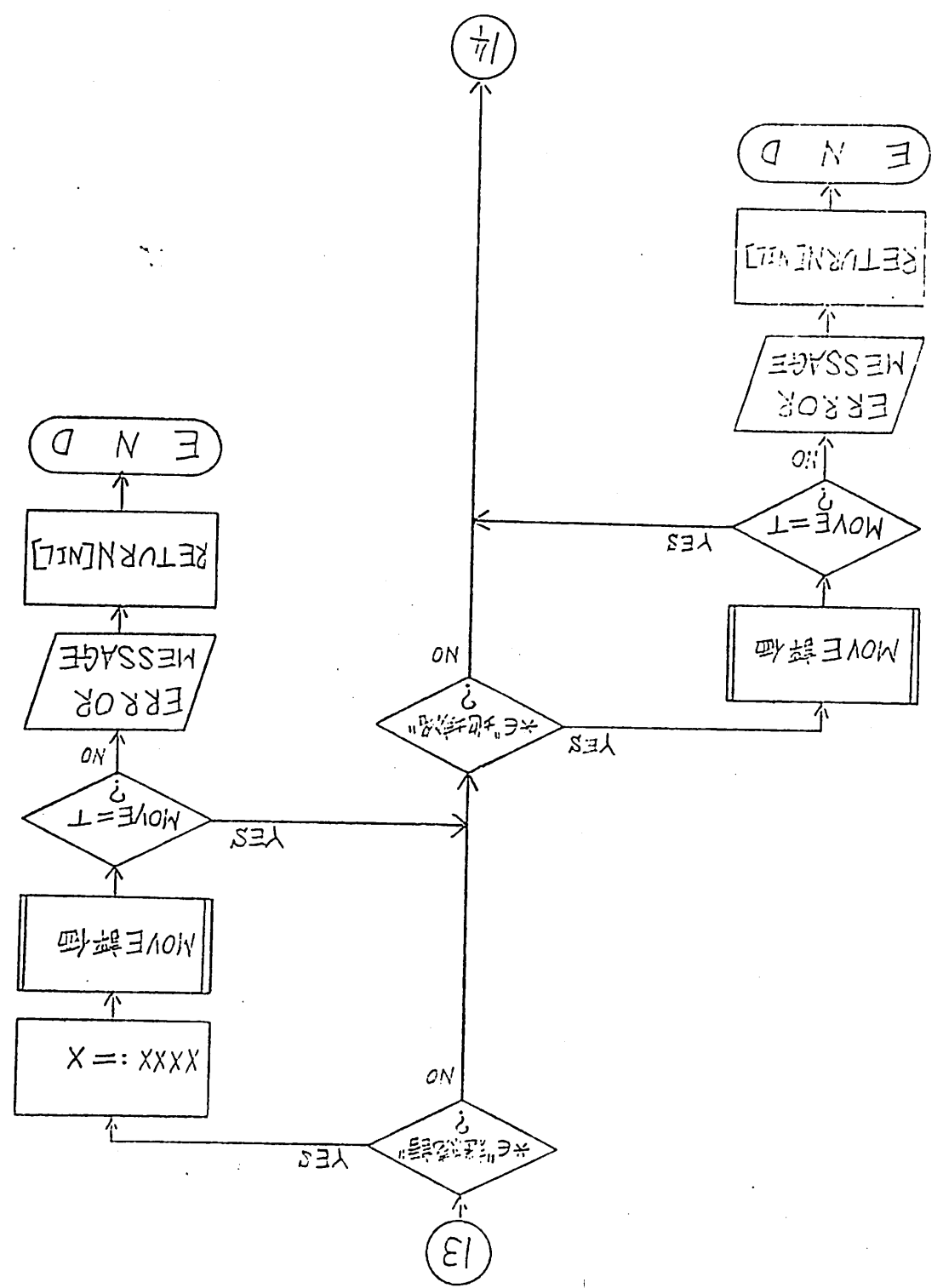


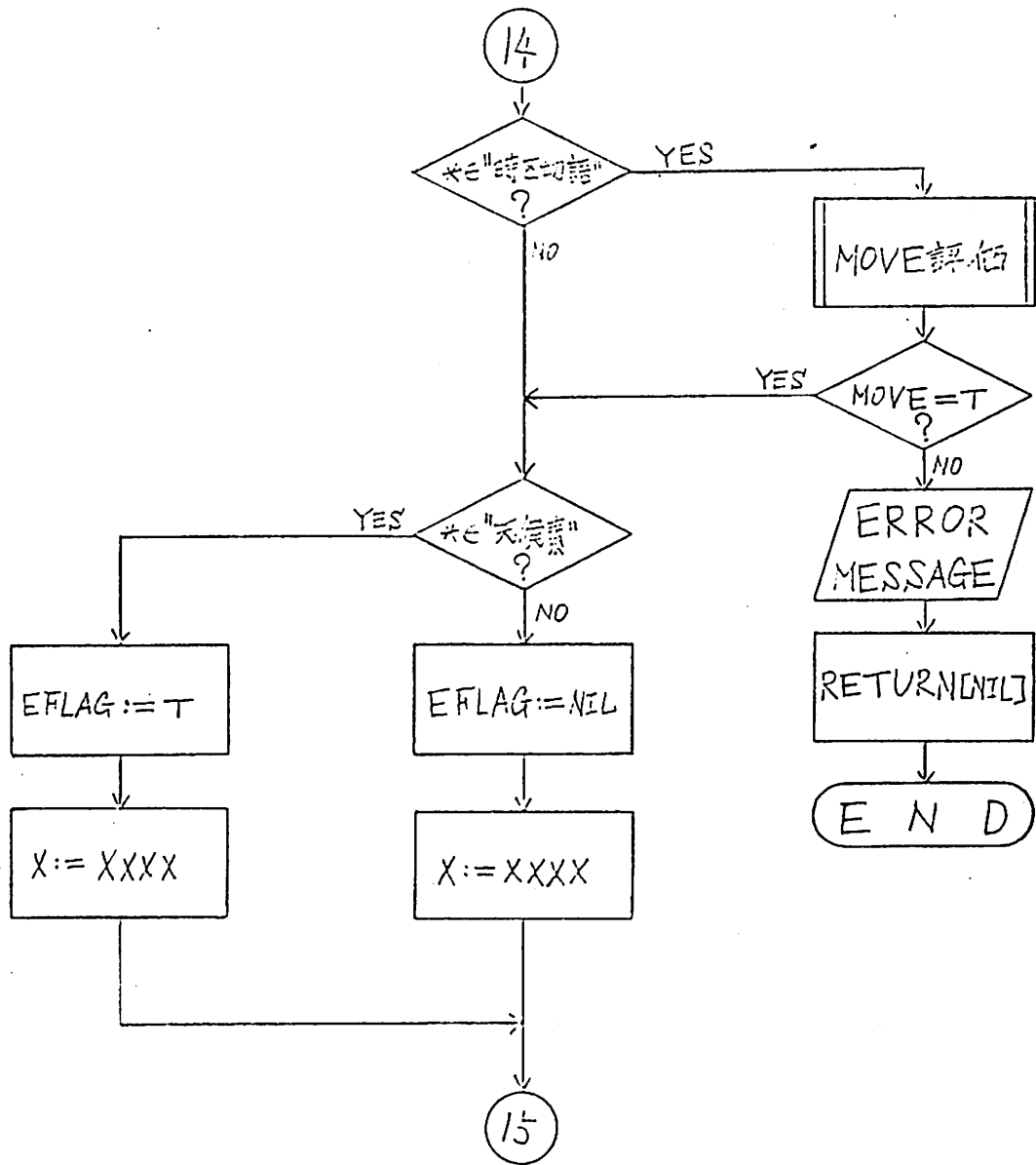
図A-8. (つづき)



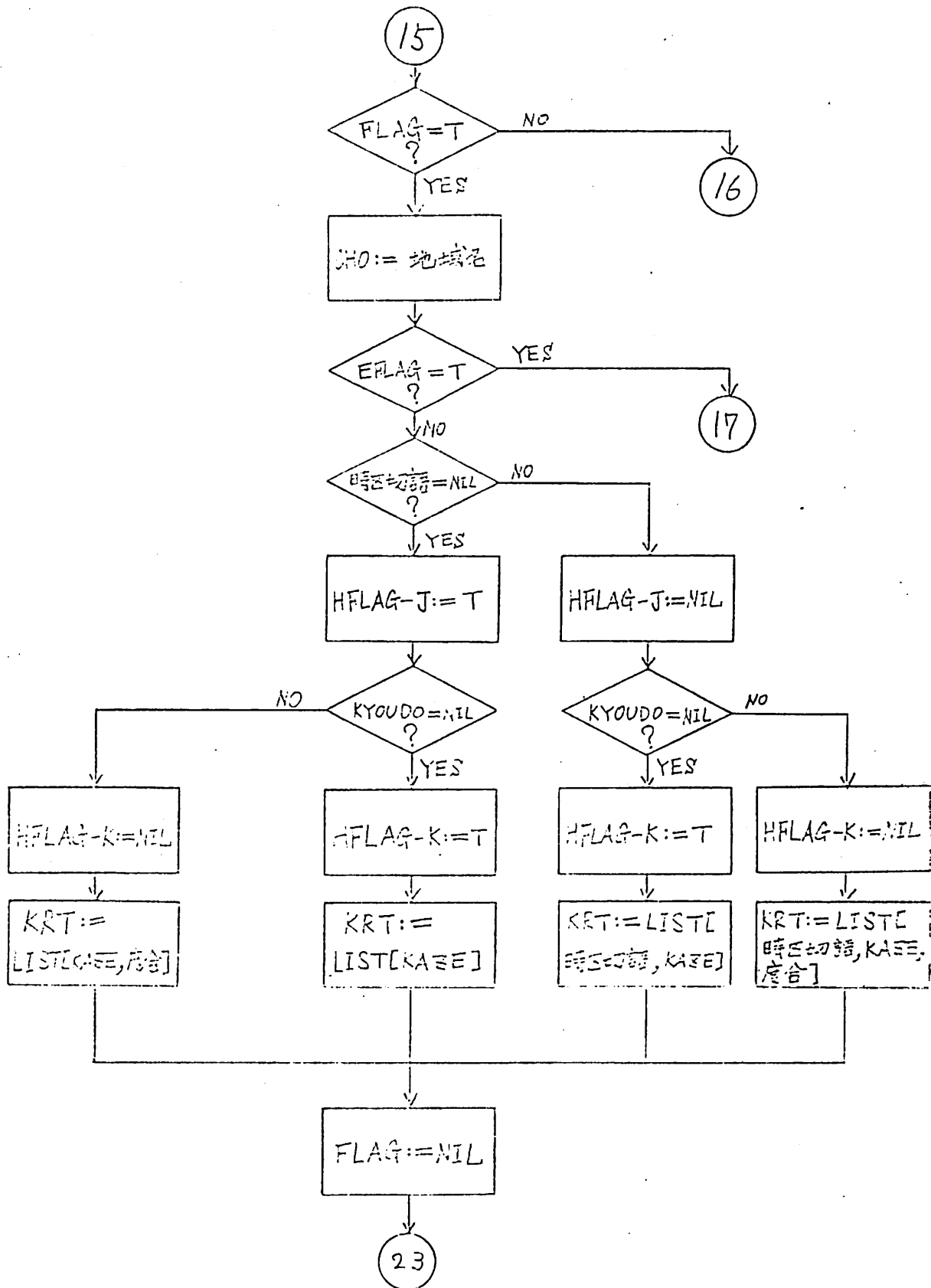
☒ A-8. (つづき)

圖 8-4 (續)

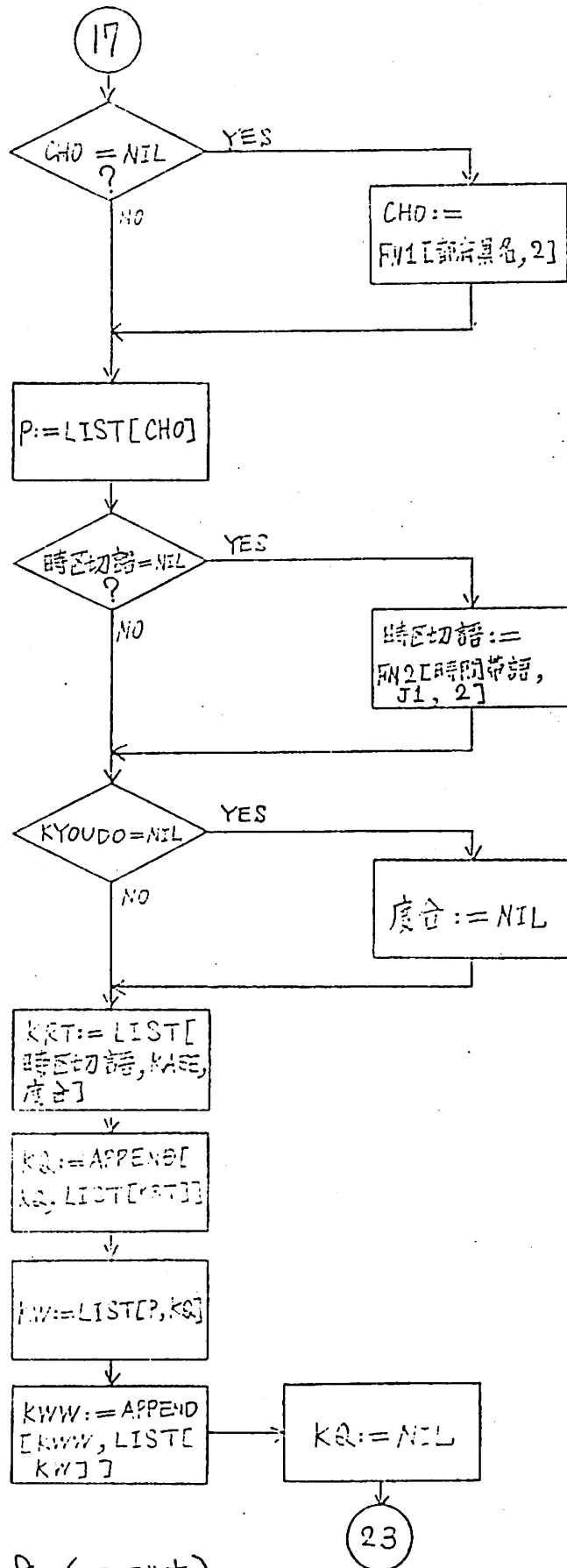




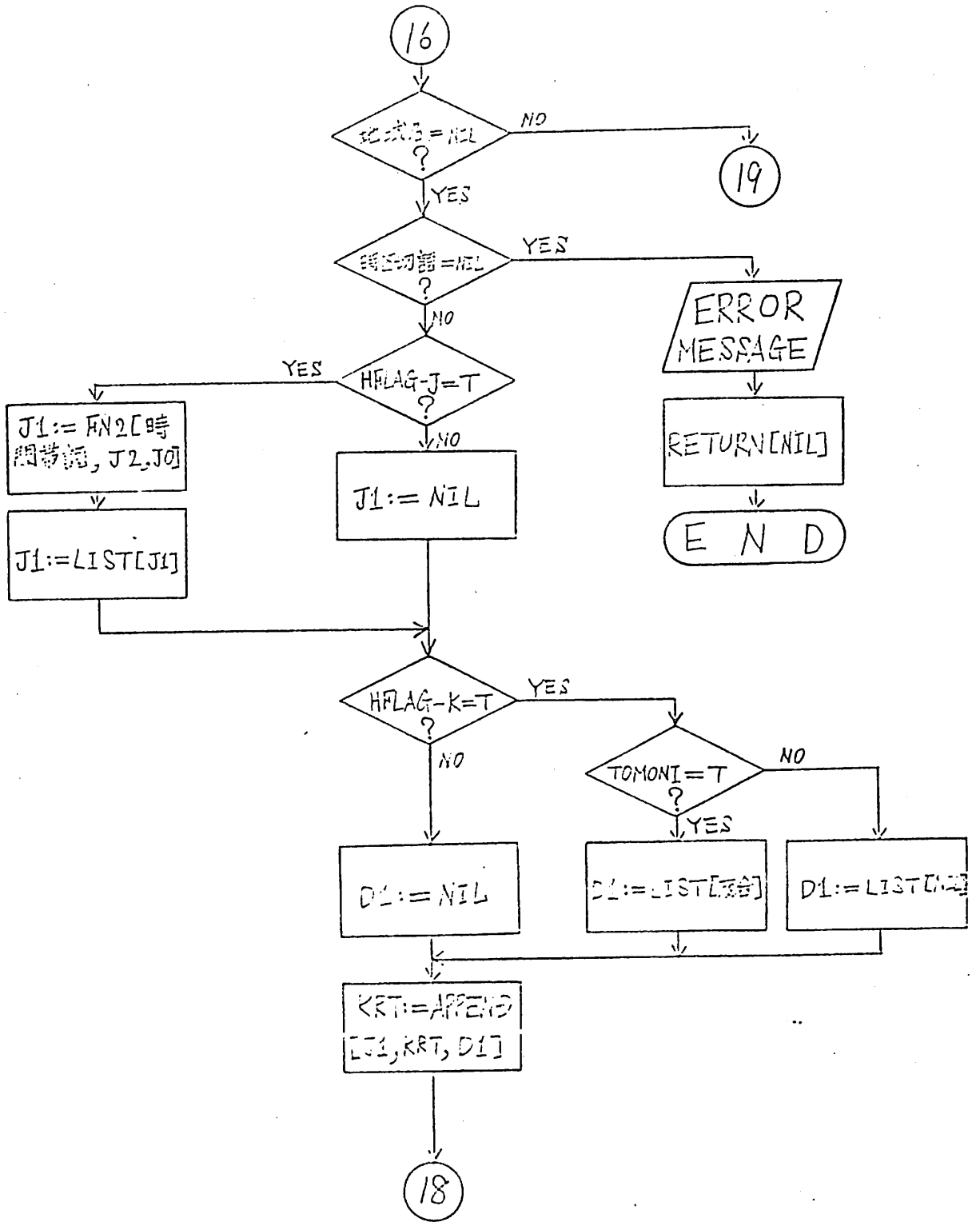
図A-8. (つづき)



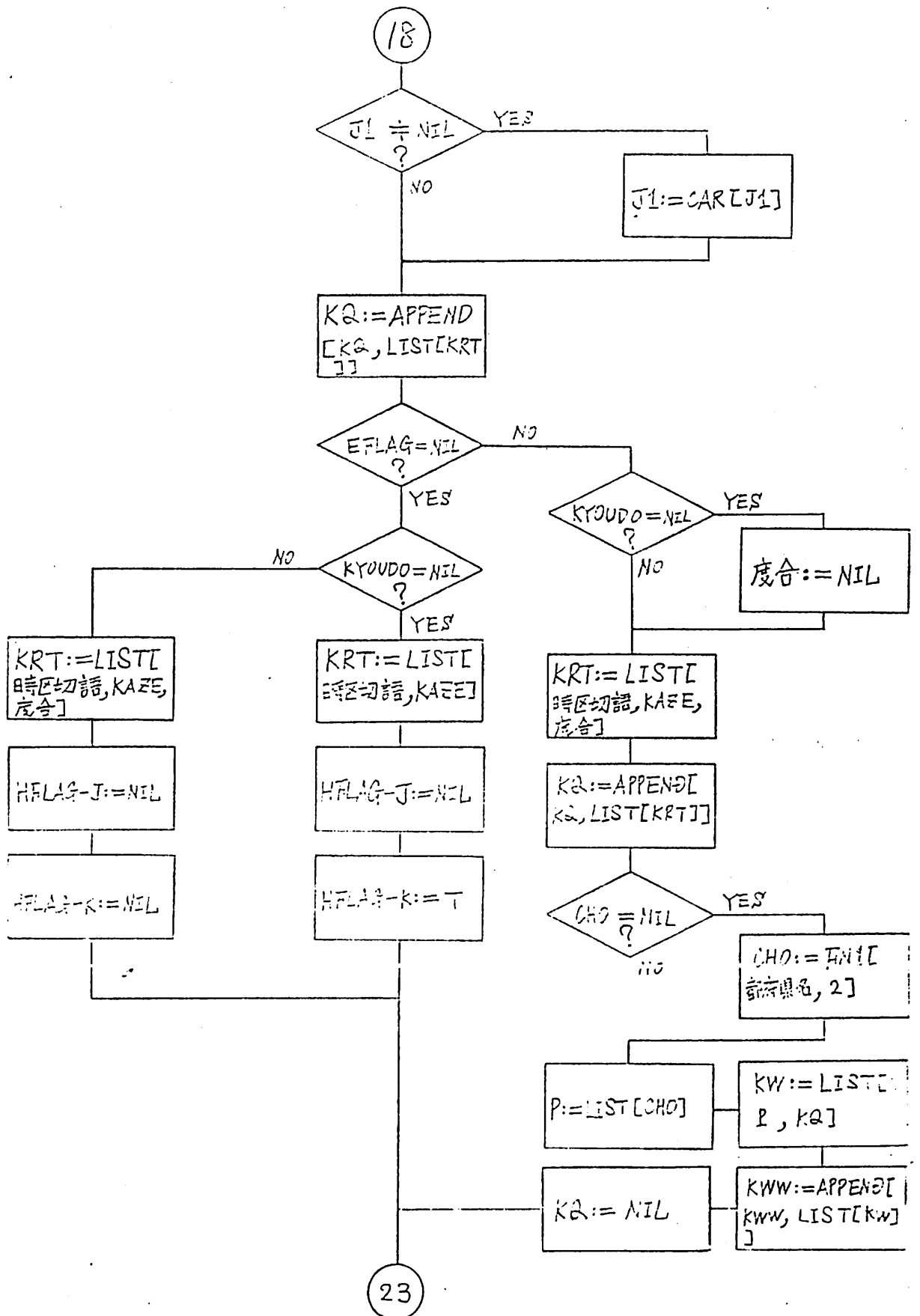
図A-8. (つづき)



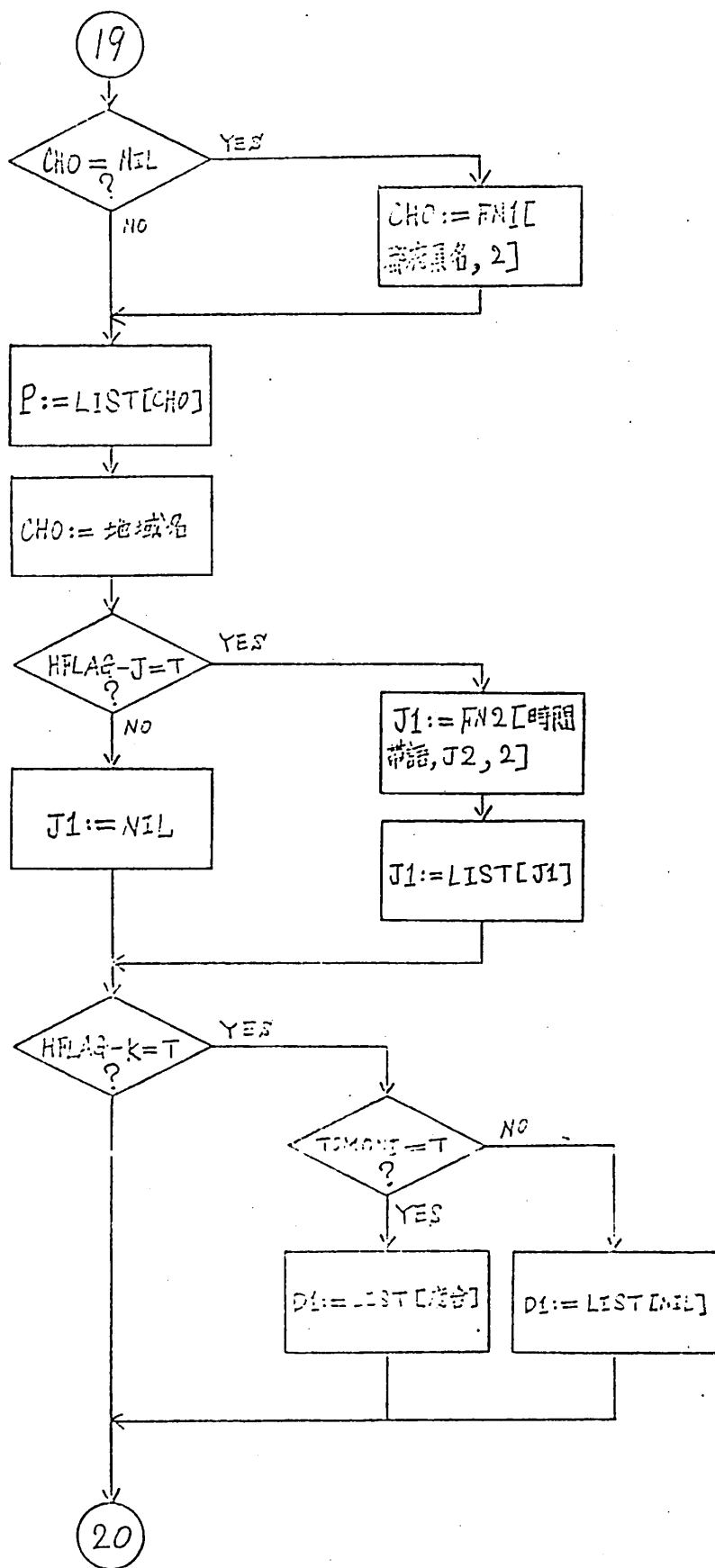
図A-8. (つづき)



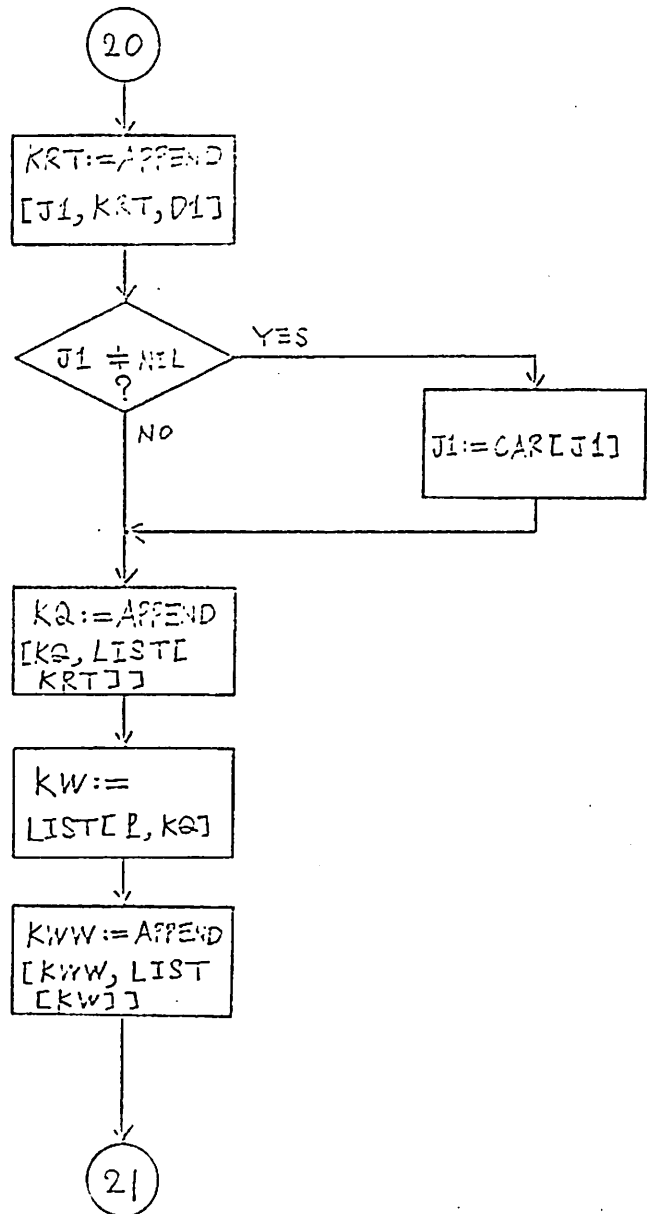
図A-8. (つづき)



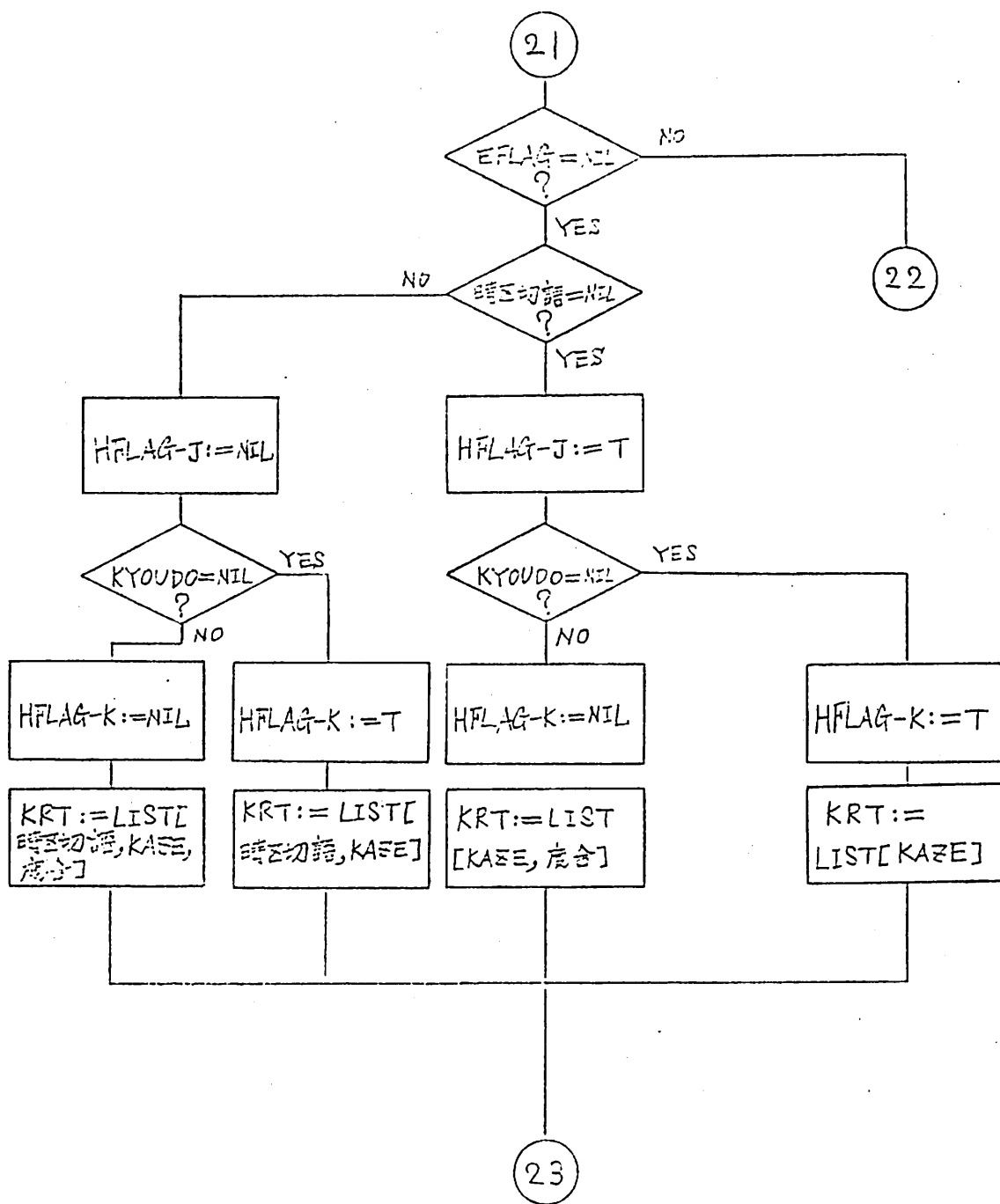
図A-8. (つづき)



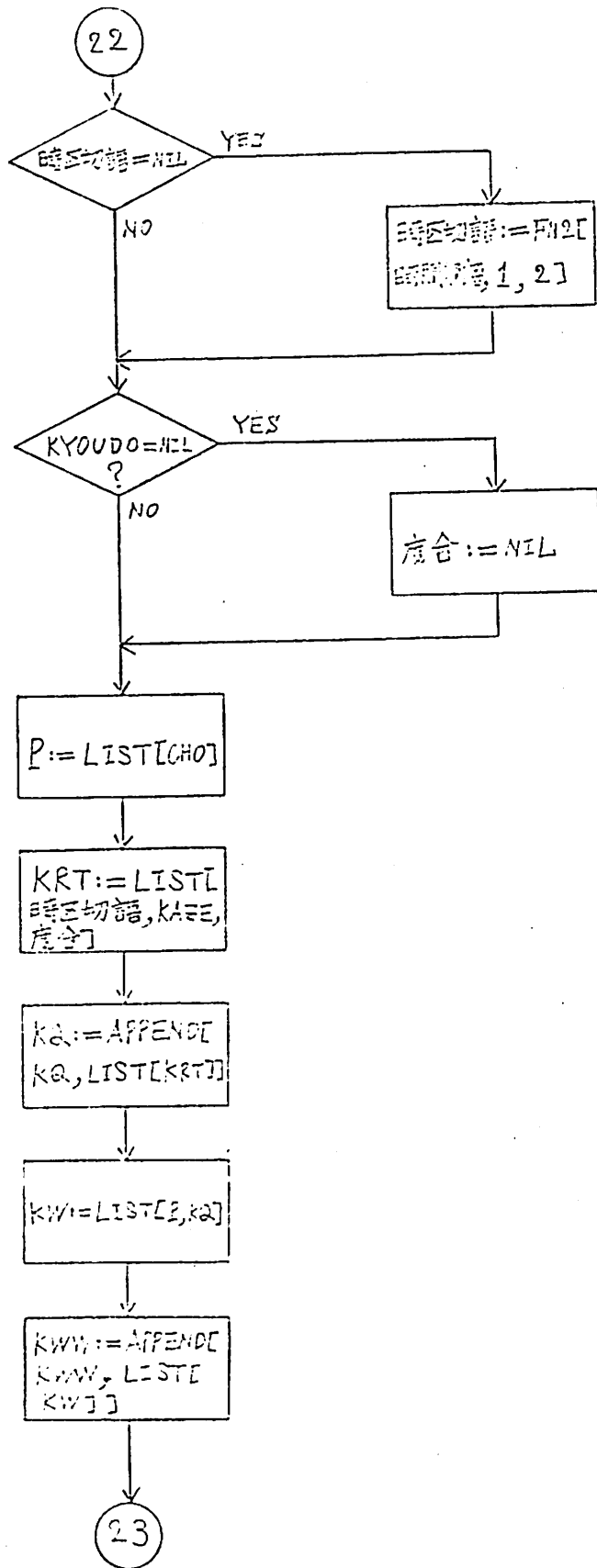
図A-8. (つがき)



図A-8 (つづき)



図A-8. (つづき)



図A-8. (つづき)

□A-8 (000) (000)

