

自然言語理解システム IMAGES-I における構文処理について

横田, 将生
九州大学医学部附属病院医療情報室

続木, 健二
九州大学大学院総合理工学研究科情報工学専攻 | 日本電信電話公社

田町, 常夫
九州大学大学院総合理工学研究科情報システム学専攻

<https://doi.org/10.15017/17521>

出版情報 : 九州大学大学院総合理工学報告. 2 (2), pp.97-105, 1981-01-20. 九州大学大学院総合理工学研究科
バージョン :
権利関係 :



自然言語理解システム IMAGES-I における 構文処理について

横田 将生*・続木 健二**・田町 常夫***

(昭和 55 年 10 月 31 日受理)

On Syntactic Processing of Natural Language Understanding System, IMAGES-I

Masao YOKOTA, Kenji TSUZUKI
and Tuneo TAMATI

Abstract

There are a lot of computer systems which interpret or understand natural language. Most of them are based on the verb-centered case theory, and considerably successful in some restricted fields. In general, such restriction of fields brings decrease in syntactic and semantic complexities which are roughly proportional to the number of verbs used in the selected field. This has often resulted in ad hoc mixture of syntactic and semantic processings in the conventional systems. However, in order to construct some general and expandable system, such mixture should be avoided.

The authors have recently constructed an experimental understanding system named IMAGES-I (Interlingual understanding Model Aiming at General purpose System-I). This system interprets sentences in an interlanguage which can realize still deeper semantic structures than the case theory, and moreover its syntactic and semantic processors are independent of each other. The syntactic processor transforms surface structures into surface dependency structures and transfers them to the semantic processor where semantic structures are generated from them and semantic anomalies, ambiguities and paraphrase relations are detected. This paper discusses the general performances of IMAGES-I and gives especially the details of its syntactic processing.

1. 緒 言

人間の言語処理機構に相当するものを計算機上で実現しようとする試みが、ここ 10 数年、人工知能研究の主要な一分野として活発に行われてきている。そして、かなりの数の言語理解システムが作製されている。ところで、そのようなシステムのほとんどが、文の構文解析処理は余り重視しておらず、いかに容易に表層構造から目標とする意味構造を得るかという意味中心でしかも手続依存の処理を行なっている。このような方式は仕事や話題を限ればかなり強力であるが¹⁾²⁾、人間が未知の事柄を文の形で学習する際、頼り

になるのは文法だけであるというような事実³⁾を考慮すると一般性や拡張性に難があると言える。従って種々の言語への適用可能性も含めて、一般的で拡張性のある方式になるためには、構文処理と意味処理を十分な程度独立させ、しかも手続中心から辞書中心にする必要がある。今回筆者らが試作した自然言語理解システム IMAGES-I (Interlingual understanding Model Aiming at General purpose System-I)⁶⁾はこのような思想に基づいて設計されている。

Fig. 1 は IMAGES-I の概念図である。この図に示すようにこのシステムでは、構文処理部および意味処理部が完全に分離しており、互いに独立に機能する。これは、本システムに採用されている意味表現が完全に language-free であるという特徴による⁴⁾。そしてこれらの各処理部は辞書の内容を変更すること

* 九州大学医学部附属病院医療情報室

** 情報工学専攻修士課程 (現在電電公社)

*** 情報システム学専攻

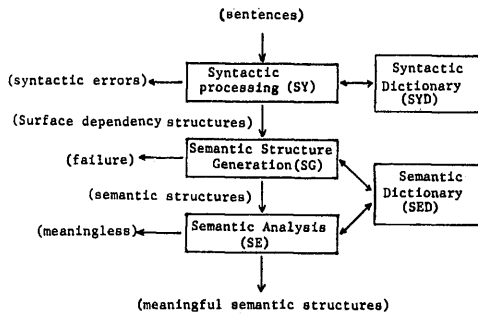


Fig. 1 Configuration of IMAGES-I.

により、更に複雑な処理や異った言語の処理が可能になるように配慮されている⁵⁾。本論文では IMAGES-I の構文処理部について述べる。

2. IMAGES-I の基本的機能⁴⁾⁶⁾

IMAGES-I は、まず入力された自然語文（今回は英文）を構文処理部（SY）において構文辞書（SYD）を参照することにより表層依存構造に変換する。そしてつぎの意味構造生成部（SG）で表層依存構造は意味辞書（SED）に登録されている各単語の概念結合操作手続に従って意味構造に変換される。最後に意味解析部（SE）において SED の各単語の概念部を参照することにより意味解析を行い、有意味な意味構造を出力する。以上の処理過程において SY では構文的に異常なもの、SG では意味構造の生成が不可能なもの、SE では無意味な意味構造がそれぞれ注釈つきで棄却される。また本システムでは各過程で生じた曖昧さ（多義）は全て次の過程に渡される。

本来、自然言語理解システム（最終的な IMAGES の目標）は Fig. 2 に示すように入力文字列の真偽判定まで行わねばならないが、IMAGES-I の機能は基本的には意味解釈レベルである。真偽判定を行うためには、推論機構や膨大な言語的・非言語的知識を用意

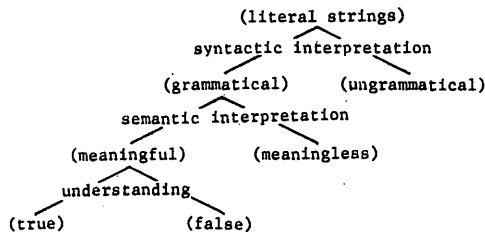


Fig. 2 Hierarchy of natural language processings.

しなければならない¹⁴⁾。

3. 構文処理部の構成

Fig. 1 に示した IMAGES-I の構文処理部を更に詳しく図示したものが Fig. 3 である。ここではその各処理過程について説明する。

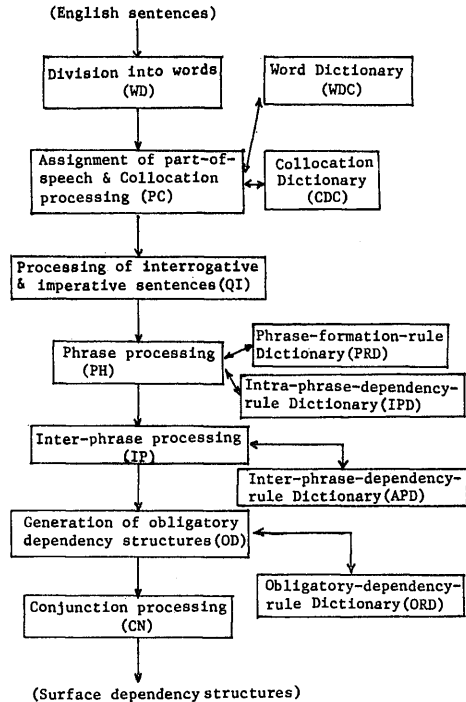


Fig. 3 Syntactic processor of IMAGES-I.

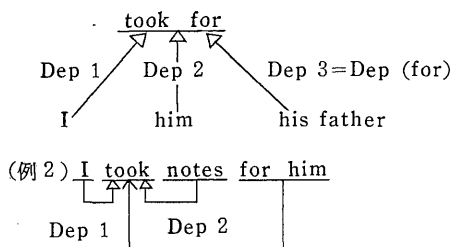
3.1 品詞割り当ておよび熟語処理

IMAGES-I に入力可能な文の種類は感嘆文を除いた能動態の平叙文、疑問文、命令文であり、Fig. 3 のWDで単語に分割され、PCにおいて Table 2 に示す単語辞書（WDC）により、それらの単語に品詞（Table 1）が割り当てられる。その際、熟語可能性情報の欄が“1”であるものは更に Table 3 の熟語辞書（CDC）が参照され入力文中に熟語があれば、まとめられた全体に一つの品詞が割り当てられる。

品詞割り当てでは多品詞語による多義があり熟語処理では熟語である場合（例1）とそうでない場合（例2）があるのが普通である。

（例1）I took him for his father

(VB)



例1の“took…for”は熟語であり、一つの動詞とされ必須的依存パターン (VDP と略記)、DP-4 (Table 7) が割り当てられるべきで、例2では“took”だけが DP-1 の動詞とされるべきである。

Table 1 List of part of speech.

Part of speech	Symbol	Word
Noun	NU	stick, fishing
Pronoun	PN	you, who
Adjective	AJ	silent, whose
Adverb	AD	quickly, when
Preposition	PR	with, from
Conjunction	CJ	after, than
Auxiliary verb	AV	can, do
Negative	NT	not, never
Verb	VB	go, tower

Table 2 Structure of Word Dictionary (syntactic part).*

TALL	OAJ/
MAN	ONU/
MOVE	ONU/OVB(1,8,)/
TO	OPR/OTO/
THERE	OAD/
THAT	OAJ/OCJ/OPN/
TAKE	1VB(1,2,)/

*The structure of each item is represented as $H\{\alpha Q[(N,)]\} /$, where $H, \alpha, Q, N, []$, and $\{ \}$ mean respectively entry word, collocation information, part of speech, the number of VDP, option, and repetition.

Table 3 Structure of Collocation Dictionary (syntactic part).*

BELONG TO	2VB(1,)/
COME BACK	2VB(8,)/
GO BACK	2VB(8,)/
TAKE FOR	2VB(1,)/

*The structure is the same as that of Table 2 but α means the number of the words within the collocation.

3.2 疑問文および命令文の処理

入力文が疑問文であるか否かは、文末に“?”があるか否かで判定できる。疑問文である場合は Fig. 4

に示す手順で平叙文の文型に変換する⁷⁾。この処理において助動詞 do を用いない疑問文 (例3, 例4) では多義が生じる場合がある。

(例3) Is the man, at work, in this room?

(例4) Ought anything, to eat, to be sold here?

また疑問詞を含む疑問文の場合には後の依存構造処理段階において疑問詞語群を倒置構文に対するのと同様な手法で文の未充足語群に相当するもの (Fig. 4 の“what you do make?”では what は動詞 make の目的格語群である) として認識することが必要である。

命令文は通常、動詞が文頭に来て、“you”が省略されていると考えられるので先頭にシステム自身を指示する“you”を補う。

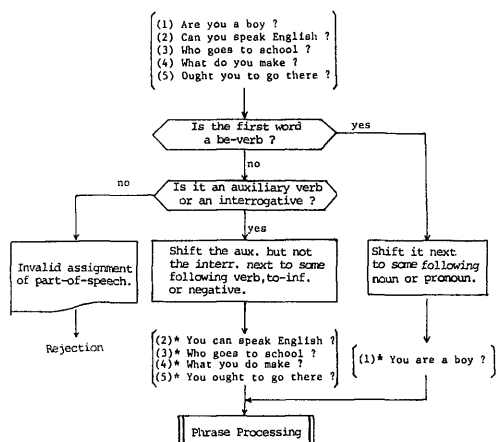


Fig. 4 Transformation of interrogative sentences into declarative ones.

3.3 語群処理

入力された単語列は語群形成規則によって分割され、更に語群内依存規則によって、語群内における単語間の governor と dependant が決定される。

3.3.1 語群形成

語群形成規則は文脈自由な書き換え規則であって、本実験システムでは Table 4 に示すものを用いた。丸括弧はその品詞をもつ単語の存在の任意性を、* は個数の任意性を表わす。入力文の単語列は先頭から順に語群にまとめられていく。また、これらの規則は上から下に優先順位が付けられており、語群形成時にお

ける例5のような多義²⁾は生じないようにしている。

(例5) young ladies

(AJGP) (NUGP)……語群系列1

(NUGP)……語群系列2

Table 4 List of phrase formation rules.

Phrase	Symbol	Rule
Noun phrase	NUGP	(AD)*+(AJ)*+NU
Noun phrase	NUGP	PN
Prep. phrase	PRGP	PR+(AD)*+(AJ)*+NU
Prep. phrase	PRGP	PR+PN
Adj. phrase	AJGP	(AD)*+AJ
Adv. phrase	ADGP	(AD)*+AD
Verb phrase	VBGP	(AV)+(NT)+VB
To-inf. phrase	TIGP	(NT)+TO+VB
Conj. phrase	CJGP	CJ

3.3.2 語群内依存構造形成

語群内依存規則は任意的なものと必須的なものより成る。必須的依存規則とは Table 4 に示すような各語群単位が意味的に最低限充足するために必ず存在しなければならない依存関係を規定したものである。本実験で用いた語群内依存規則を Table 5 に示す。この表で…→は任意、…→は必須を表わし、位置としての左辺、右辺には意味がない。また、dependant と governor が同一品詞である依存規則（ここでは副詞のときだけ）が適用される場合は両者が隣接している場合だけとしている。その他の依存規則は、隣接している必要はない。このため依存規則の適用に関して左から右に優先順位を低くしている。

Table 5 List of intra-phrase dependency rules.

Phrase	Rule
NUGP	AD---→AD, AD---→AJ, AJ---→NU
PRGP	AD---→AD, AD---→AJ, AJ---→NU, NU---→PR
AJGP	AD---→AD, AD---→AJ
ADGP	AD---→AD
TIGP	NT---→TO, VB---→TO
VBGP	AV---→VB, NT---→VB

ところで依存関係が概念的結合関係と完全に一致しなければならないという立場では、本稿に示すものでは不十分である。たとえば例6の副詞間の依存関係は概念的結合関係を表わしているとは考えられない。こういう立場では個々の単語の意味を考慮したかなり複雑な規則を設定する必要がある。R. C. Schank らの概念依存構造 (conceptual dependency)²⁾は、表層構造からの変換アルゴリズムを明確にはしていないが

このような立場のものである。すなわちこの問題は例7のように、できるだけ簡便に文の持つ概念情報を表層に近いレベルで把握しようとする方法論での限界を示していると考えられる。

(例6) I slept here very early yesterday.

(例7) I made him go.

これに対して IMAGES-I では依存構造は更に深いレベルの特定の言語に依存しない概念構造に変換するための一つの過程でしかないので、大すじで概念結合関係と一致しておれば十分である。というのは筆者らの依存構造から概念構造を生成する方法は単語意味辞書の結合操作部⁴⁾に手続きとして記述してあるためかなり自由な概念結合操作を行わせることができるからである。したがって依存規則は簡単に品詞情報だけで構成できるようにしている。

3.4 語群間処理

文全体の意味を捉えるためには動詞を中心に意味解釈が進められる。文中に出現している動詞語群はそれぞれ、ある範囲内の他品詞語群を支配 (govern) している。この支配範囲をブロックと呼ぶ⁵⁾。語群にまとめられた入力文は、ブロックに分割され、そのブロック内で依存構造が生成される。

3.4.1 ブロック分割⁶⁾

語群系列は以下の手順でブロックに分割される。

1) 接続詞語群 (コンマを含む) は、その前後がブロックの境界となり、語群自身でブロックを構成する。

(例8) …//, //……//and//…

2) 関係詞語群、疑問詞語群、that 節の that はその直前がブロックの境界となり、自らは後続のブロックに含まれる。

(例9) I think//that she is ill.

(例10) The picture//for which…

(例11) This is//what I wanted.

3) 1), 2) を適用してなお1ブロックに2個の動詞語群を含む場合は、後の動詞語群の直前がブロックの境界となる。これは原形不定詞、TO不定詞、分詞などが存在する場合である。

(例12) I made him//sing a song.

(例13) She likes//playing the piano.

このブロック分割法では、関係詞や that 節の that 等が省略されている文に関しては、(例14)に示

ような不都合が生じる。正しくは破線の方であるが、この方法では実線の部分で分割される。この問題への対処のし方は3.5で述べる。

(例14) This is the book /I//like best.

3.4.2 語群対生成

語群間任意依存規則を適用するために、ブロック内の語群で対を作り出す⁷⁾。これは依存規則を辞書形式で登録し、それを適用するための前処理としては有効である。その手続きおよび例は以下のようなものである。

1) 動詞語群はすべての語群と対を作る (例15における一)。

2) 動詞語群が関係していない場合は、隣接する2語群が対を作る (例15の[...])。

(例15) Tom shows Jim the letter to Mary.

実際、動詞語群が関係しない場合、他の語群を飛び越して依存するものは見当らないようである。またそれを考慮すると依存関係の交差を調べなければならなため処理が複雑化する。

3.4.3 語群間任意依存関係

語群間任意依存関係とは、ブロック内の動詞等が必須的に要求する依存関係 (3.5 参照) とは独立に形成される一般的な依存関係である。本実験では Table 6 に示す文脈自由な規則を用いた。規則の左辺、右辺には意味があり、その順になっている時のみ、その規則が適用される。なお DR-5, DR-6 はブロックを越えて依存構造を形成するものである。

これらの規則を適用すれば、(例16) の場合等で D1, D2 のような曖昧さを生じる。

(例16) Tom approaches Jim with a gun.

Table 6 List of inter-phrase dependency rules.

Symbol	Rule
DR-1	NUGP → VBGP
DR-2	VBGP ← PRGP
DR-3	NUGP ← PRGP
DR-4	VBGP ← ADGP
DR-5	NUGP ← NUGP*
DR-6	VBGP ← ADGP**

*relative pronoun

**relative adverb

←: 必須依存関係 (3.5 参照)

←: 任意依存関係

また、これらの規則の中には含まれていないが、例17のように前置詞語群がブロックの先頭にある場合は後続の動詞語群に係るという文脈依存の規則も用いている。

(例17) On Sundays they go to church.

3.5 語群間必須依存関係

前置詞語群内における前置詞とそれの目的語である名詞語群との関係のように、ブロック単位として意味的に最低限度充足するために語群間においても必須的な依存関係が存在する。この場合、例18, 19, 20のように動詞、形容詞、名詞が governor となって必須的な依存構造を構成する。

(例18) I think him honest.

(例19) (Make him) aware of the fact.

(例20) (His) anxiety for money...

このような依存関係には特定のパターンがあり、動詞に関しては Table 7 に示す37個を抽出している。これらのパターンでは例7で説明したように個々の要素の依存関係よりも構成要素の配列順序により、まとまった意味が伝達されると考える方が自然である。一般に格構造により意味解釈・理解を行う場合、無限個存在する可能性のある格要素⁹⁾ から適切な名称を与えながら有限個の必須要素を抽出する必要性が生じてくる。その際、言語外の客観的対象に言及しながら抽出を行う場合¹¹⁾¹⁰⁾ を除いてはその作業は研究者の主観に

Table 7 List of inter-phrase obligatory dependency patterns concerning English verbs.*

DP-1	NUGP+VBGP+NUGP	DP-23	NUGP+VBGP+NUGP+THTN
DP-2	NUGP+VBGP+NUGP+AJGP	DP-24	NUGP+VBGP+WHCN
DP-3	NUGP+VBGP+NUGP+NUGP	DP-25	NUGP+VBGP+WHTN
DP-4	NUGP+VBGP+NUGP+PRGP	DP-26	NUGP+VBGP+NUGP+WHTN
DP-5	NUGP+VBGP+PRGP+NUGP	DP-27	NUGP+VBGP+NUGP+WHCN
DP-6	[So]+NUGP+VBGP	DP-28	GRDN+VBGP+GRDN
DP-7	NUGP+VBGP+{[so]}	DP-29	TIFN+VBGP+TIFN
DP-8	NUGP+VBGP	DP-30	GRDN+VBGP+{AJGP NUGP PRGP}
DP-9	[There]+VBGP+NUGP	DP-31	TIFN+VBGP+{AJGP NUGP PRGP}
DP-10	NUGP+VBGP+AJGP	DP-32	NUGP+VBGP+{it}+PRGP+{TIFN WHTN NUNN}
DP-11	NUGP+VBGP+PRGP	DP-33	{it}+VBGP+{WHTN TIFN NUNN}
DP-12	[Here]+VBGP+NUGP	DP-34	{it}+VBGP+{AJGP NUGP GRDN PRGP}
DP-13	NUGP+VBGP+TIFN	DP-35	NUGP+VBGP+PTPN
DP-14	NUGP+VBGP+NUGP+TIFN	DP-36	[There]+VBGP+[no]+GRDN
DP-15	NUGP+VBGP+{there}+{to}+{be}+NUGP	DP-37	NUGP+VBGP+ADGP
DP-16	NUGP+VBGP+{it}+{AJGP+TIFN NUGP PRGP GRDN}		
DP-17	NUGP+VBGP+NUGP+BIFN		
DP-18	NUGP+VBGP+PRGP+PTPN		
DP-19	NUGP+VBGP+NUGP+PTPN		
DP-20	NUGP+VBGP+GRDN		
DP-21	NUGP+VBGP+{THTN LSTN}		

*TIFN:To-infinitive nexus, THTN:That-clause, GRDN:Gerund nexus, BIFN:Bare-infinitive nexus, PTPN:Past-participle nexus, PRPN:Present-participle nexus, NUNN:Noun clause, LSTN:Least-clause, WHCN:Interrogative clause, WHTN:Interrogative-to-infinitive.

頼らざるを得ず、非常に曖昧であった。これに対して筆者らの方法⁴⁾は、表層依存構造からその意味構造である中間言語表現に変換するときの手続きが governor に特有のものであれば必須的依存関係, governor によらず一般的なものであれば任意的依存関係とする必然性のあるもので、その抽出過程は明確に理由づけることができる。

3.5.1 語群間必須依存規則の適用

文の骨格的意味構造を決定するのは動詞であり、その必須的依存構造パターン（以下、必須パターンと略す）が構文的に最も重要である。個々の動詞は一般に複数個の必須パターンを持ち、使用された議論世界（universe of discourse）におけるその動詞の意味とほぼ一対一に対応する（この場合の意味とは概念部および結合操作部⁴⁾のことである）。このため、入力文中に使われた動詞がどの必須パターンに属するものかが正しく選択されなければ満足な意味処理は期待できない。実際動詞の意味辞書は、必須パターンに応じて結合操作部が異なり、場合によっては概念部さえ異なることもある。例えば、“make”は必須パターンが DP-1 のときは“製造”の意であり、DP-17 のときは単なる“使役”の意になる。

処理手続を動詞“move”を例に挙げて説明する。“move”には他動詞と自動詞があり、それぞれ DP-1, DP-8 の必須依存構造になる。次の例文では、例 21 が他動詞、例 22 が自動詞の場合である。なお矢印（→）は 3.4 での結果を示している。

(例21) He moved the chair to the fire.
(NUGP)(VBGP) (NUGP) (PRGP)

(例22) The earth moves round the sun.
(NUGP) (VBGP) (PRGP)

例 21, 例 22 が入力されると、共に DP-1, DP-8 の構造を持つものとして試される。

1) DP-1 が選択された場合の結果

例 21……必須語群充足

例 22……動詞語群後方の名詞語群が未充足

例 22 はブロックが一つなので、充足すべき名詞語群は見つからず、パターン選択が誤りであることがわ

かり棄却される。一方、例 21 は必須語群が全て充足されているので、結果として次のような依存構造が生じる。

(例 21') He moved the chair to the fire.

その場合、任意依存構造のうち必須パターンに含まれるものはこの例の He moved のように必須依存構造に変換される。

2) DP-8 が選択された場合の結果

例 21, 例 22……共に必須語群を充足している
但しこの場合処理結果はそれぞれ次のようになる。

(例21'') He moved the chair to the fire

(例22') The earth moves round the sun.

すなわち例 21 に対する処理結果においては“the chair”が孤立してしまい、governor を持たなくなる。これは動詞以外は（ブロックが2個以上あるときは動詞も）governor を持つという重要な規則に違反するのでこれも棄却される。一方、例 22 に対する処理結果は未充足語群も孤立語群もないので DP-8 の選択が適切であったことがわかる。

3.5.2 未充足語群情報を用いたブロック結合

文中に出現する個々の動詞は、必須的に結合する語群をもっているが、関係詞節等によって分断されると、必須語群が未充足の状態となる。以下、次の例文を用いて、その処理方法を説明する。

(例23) The man // who made Tom // go //
came back to England.

ブロック分割、任意依存構造生成、ブロック内での必須依存構造生成の結果は上記例文中に示しているようになる。“make”は必須パターン DP-1, 17, “go”と“come”は DP-8 の構造をもつので、その取り得る組み合わせは Table 8 のようになる。

Table 8 Block information of Example 22.

No.	Block	Case I		Case II	
		Missing	VDP	Missing	VDP
1	The man	—	0	—	0
2	who made Tom	—	1	—	17
3	go	NUGP (-1)	8	NUGP (-1)	8
4	came back to England	NUGP (-1)	8	NUGP (-1)	8

第1ブロックは述語を含まない名詞語群なので、どの必須パターンとも照合されず、パターン番号は0とし、未充足語群情報は言及されない。また未充足語群

† 例21に関しては、実際にはここまでの段階で下のような依存構造もできている。

He moved the chair to the fire.

の欄において括弧内の数は、動詞語群との相対的位置を表わし、 $X_g(-n)$ は動詞語群から前（左）方へ順に（ X_g だけを）数えて n 番目、 $X_g(n)$ は逆に後（右）方 n 番目の X_g なる語群が未充足であることを示している。以下はこれらの情報を用いたブロック結合の処理手順である。

(Step 1) 文の先頭から順に未充足語群 $X_g(m)$ を持つブロックを探す。

(Step 2) 他のブロックから未充足語群 $X_g(m)$ を満足する語群（ $m < 0$ なら前方、 $m > 0$ なら後方）を探索し結合する。その際、充足された未充足語群情報 $X_g(m)$ を削除する。

(Step 3) 以上 (Step 1)、(Step 2) の手続を未充足語群が全て満足されるか、それらを満す語群が存在しなくなるまで続ける（未充足語群が全て満たされるときにだけブロック結合が成功したことになる）。

但し以上の手続において、TO不定詞語群や分詞語群がブロック内にある場合は、前方の名詞語群は未充足語群に含めない。原形不定詞語群の場合は、それが単なる動詞語群であるかどうか判明しにくいので、一応、前方の名詞語群は未充足語群に含めるが、原形不定詞を要求しているブロックがあれば、そのブロックと結合される。また依存関係が交差した場合は当該語群対を棄却し、新たな対を探す。

例 23 に対する処理を具体的に述べると以下のようになる。

[I] Case I の場合

i) (Step 1) の結果

第 3 ブロックで動詞語群より前方の名詞語群が未充足であることがわかる。

ii) (Step 2) の結果

The man // who made Tom // go //
came back to England.

iii) (Step 1) の結果

第 4 ブロックが未充足語群をもつ最後のブロックである。

iv) (Step 2) の結果

第 1 ブロックの名詞語群は既に governor を持っているので第 4 ブロックと結合するブロックは見つからない。従って必須パターンの選択が正しくなかったと判定される。

[II] Case II の場合

i) (Step 1) の結果

第 2 ブロックで原形動詞語群が未充足であることが分る。

ii) (Step 2) の結果

(Step 3) 以下に述べたように第 2 ブロックと第 3 ブロックとが結合される。

The man // who made Tom // go //
came back to England.

iii) (Step 1) の結果

第 4 ブロックで名詞語群が未充足である。

iv) (Step 2) の結果

第 4 ブロックは名詞語群である第 1 ブロックと結合される。その結果未充足語群をもつブロックはなくなり、ブロック結合が完成する。

The man // who made Tom // go // came back
to England.

次に関係詞や that 節の that 等が省略された場合を考えてみよう。例 14 においてブロック内で必須依存構造が生成された結果は次のようになる。

This is the book I // like best.

ここで第 1 ブロックは未充足語群がないが第 2 ブロックの方は、like の必須パターンを参照することにより、2 つの名詞語群が欠落していることがわかる⁺。従って第 1 ブロックで governor を持たずに孤立している “I” を “like” の主語（名詞語群（-1））として結合させブロック境界を修正するが、まだ目的語（名詞語群（+1））が未充足である。従って関係代名詞 that を第 2 ブロックの先頭に挿入して次のものを得る。

This is the book // that I like best.

最後に語群間任意依存規則（Table 6）より that の先行詞が求められ、次の依存構造が結果として生じる。

This is the book // that I like best.

3.6 節間依存構造の生成

英語文はそれを構成する節の結合関係により、次の

⁺like は必須パターン DP-1 の構造になるので、この場合、動詞の前方と後方の名詞語群（主語と目的語のこと）が未充足である。

ように分類される¹¹⁾.

- 1) 単文 (Simple sentence)
- 2) 重文 (Compound sentence)
- 3) 複文 (Complex sentence)
- 4) 混文 (Mixed sentence)

ここで問題にすべきものは2)~4)であるが、4)は2), 3)の複合であると考えられるので除外する。

重文とは、2つ以上の等位節（独立節）を含む文のことである。これらの節をつなぐものは等位接続詞であるが、継続的用法の関係代名詞、関係副詞も等位接続詞の機能をもつと考えてよい。また複文とは従属節を含んでいる文のことで、主節と従属節をつなぐ役割を果たすものは従属接続詞と呼ばれる。that節のthat, 制限的用法の関係詞はそれぞれ、名詞節、形容詞節を作る従属接続詞と考えられる¹¹⁾。しかしながら、等位接続詞とか従属接続詞とかいう区別は形式上の分類であって、論理の意味とは無関係のことが多い¹²⁾。このことは次の2つの例より明白である。

(例24)

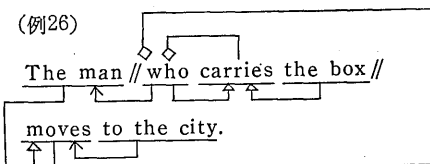
Hurry up or you will be late for school.
=If you do not hurry, you will be late for school.

(例25)

Move an inch, and you will be dead.
=If you move an inch, you will be dead.

すなわち、これらの例において、意味的には左辺の等位接続詞 or, and は右辺の従属接続詞によって書き換えが可能である。従ってここではこれらを区別することなく接続詞として扱い、2つの節を支配するとし以下のように依存構造を表示する。

(例26)



3.7 多品詞語処理

単語ごとの“品詞割当て”処理では各単語にはその取りうる品詞の全てが割り付けられる。従って多品詞語が多い英語の場合、一つの入力文に対してこの処理で与えられる品詞の組合わせ（品詞系列と呼ぶ）の数はかなり大きなものになりうる。しかしながら文法的に正しい組合わせはその内の極少数であり文法的に正

しくないものは語群形成または依存構造生成時に棄却される。例27における“to”の品詞に関する多義は語群形成過程において、語群形成規則（Table 4）を参照することにより解消することができる。

(例27) ...to the big city.

$$\left\{ \begin{matrix} (PR) \\ (TO) \end{matrix} \right\} \times (AJ) \times (AJ) \times (NU)$$

また例28における“rose”は動詞と名詞の2つの品詞を持っている。この例の場合、2つの品詞系列とも語群形成規則には違反していないので語群形成時においてはその曖昧さは解消できない。しかしながら次の依存構造生成過程で、roseを名詞とした場合は語群間任意依存規則（Table 6）に違反し依存構造が生成できないとして棄却される。

(例28) The man often rose from the bed.

$$(NUGP)(ADGP) \left\{ \begin{matrix} (NUGP) \\ (VBGP) \end{matrix} \right\} (PRGP)$$

4. 結 論

自然言語理解システム IMAGES-I の構文処理部について述べた。従来、自然語文の意味表現は大部分が深層格を用いる深層依存構造（deep case dependency）に基づいているため、依存構造そのものが単語のもつ概念間の関係を直接反映している必要があり、かなり個々の単語の意味に影響を受けた余り体系的でない格の抽出や構文情報と意味情報が混然一体となった変形操作が必要であった。これに対して、IMAGES-I の構文処理部が出力する表層依存構造はあくまでも自然語文のもつ意味の第一次近似であり、次の意味構造生成過程において、体系的に意味構造に変換されるまでの過渡的存在である。従って、この構文処理部で使用している規則やそれを構成している基本概念は全て単語のもつ文法的性質や機能にだけ基づいたものであり、従来のものよりかなり単純なものになっている。またこのことにより構文処理部と意味処理部¹³⁾は完全に分離され、改変や異種言語処理への適用が容易である。なお、プログラムの大きさはIMAGES-I 全体が260 KB、構文処理部は195 KBであった。計算機は本学大型計算機センターのFA-COM-M200、プログラム言語はPL/Iを使用した。構文処理部の入出力結果例は付録を参照されたい。

本研究の一部は昭和54年度文部省科学研究費（奨励研究（A））によった。

参 考 文 献

- 1) Winograd, T.: "Understanding natural language," Academic Press (1972).
- 2) Schank, R. C.: "Conceptual Information Processing," North-Holland (1975).
- 3) 長尾 真: "自然言語の理解", 情報処理, 19-10, p. 961(1978).
- 4) 横田, 田町: "視覚化された概念モデルに基づく自然語の意味解釈について", 信学論 (D), 63-D, 5, p. 424(1980).
- 5) 横田, 田町: "言語・図形の対応構造に基づく気象通報の意味解釈について", 信学論 (D), 63-D, 6, p. 530(1980).
- 6) 続木健二: "自然語理解システムにおける構文および意味処理", 九大修士論文 (1980).
- 7) 首藤友喜: "依存構造論による英日機械翻訳の実験", 九大修士論文 (1972).
- 8) 石原, 田町: "D-tree モデルとそれに基づく英日機械翻訳のための言語分析について", 信学論 (D), 57-D, 7(1974).
- 9) Langendoen, D. T.: "Essentials of English Grammar," Holt, Rinehart & Winston (1971).
- 10) 岡田, 田町: "自然語および図形解釈のための単純事象概念の分析および分類", 信学論 (D), 56-D, 10(1973).
- 11) 木村 明: "英文法精解", 培風館 (1973).
- 12) 毛利不可信: "意味論から見た英文法", 大修館書店 (1972).
- 13) 横田, 続木, 田町: "自然言語理解システム IMAGES-I における意味処理について", 九大総理工研究科報告, 2, 2, pp. 107-113 (1981).
- 14) Katz, J. J. and Fodor, J. A.: "The structure of a semantic theory", Language, 39, 2, pp. 170-210 (1963)

付 録

Fig. 5 は IMAGES-I の構文処理部に関する入出力結果例である。例 29 の英語文入力に対して、語群処理、依存構造処理が施され "RESULT OF DEPENDENCY STRUCTURE GENERATION" 以下に示す依存構造が出力される。RELDEP (1), RELDEP (2) はそれぞれ例 29 における D1, D2 と対応している。

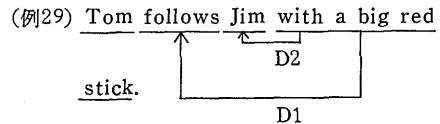


Fig. 5 Example of syntactic processing of IMAGES-I.

```

THE NUMBER OF WORDS=      8
INPUT SENTENCE=TOM FOLLOWS JIM WITH A BIG RED STICK .
RESULT OF 'QIPRODC' AFTER 'CONDENS'
RELDEP (1)=<(TOM)NU >NUGP <(FOLLOWS)VB >VBGP <(JIM)NU >NUGP <(WITH)PR
(A)AJ <(BIG)AJ <(RED)AJ <(STICK)NU >PRGP ' ;
ENTERED IN WGPING
RELDEP (1)=<(TOM)NU >NUGP <(FOLLOWS)VB >VBGP <(JIM)NU >NUGP <(WITH)PR
(A)AJ <(BIG)AJ <(RED)AJ <(STICK)NU >PRGP ' ;
RELDEP (1)=<(TOM)NU1 >NUGP1 <(FOLLOWS)VB1 >VBGP1 <(JIM)NU2 >NUGP2 <(WITH)
PR1 (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3 >PRGP1 ' ;
RELDEP (1)=<(TOM)NU1 >NUGP1 <(FOLLOWS)VB1 >VBGP1 <(JIM)NU2 >NUGP2 <(WITH)
PR1 (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3 >PRGP1 <(JIM)NU2
>NUGP2 <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:AJ1+:AJ2+:AJ3
+>PRGP1:0 / ' ;
RELDEP (2)=<(TOM)NU1 >NUGP1:0 <(FOLLOWS)VB1 >VBGP1:NUGP1* <(JIM)NU2 >NUGP2
PRGP1* <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:AJ1+:AJ2+:AJ3+
>PRGP1:0 / ' ;
RELDEP (1)=<(TOM)NU1 >NUGP1:0 <(FOLLOWS)VB1 >VBGP1:NUGP1:PRGP1*:NUGP2# <
(JIM)NU2 >NUGP2:0 <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:AJ1+
:AJ2+:AJ3+>PRGP1:0 / ' ;
RELDEP (2)=<(TOM)NU1 >NUGP1:0 <(FOLLOWS)VB1 >VBGP1:NUGP1*:NUGP2# <(JIM)NU2
>NUGP2:PRGP1*:0 <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:AJ1+:
AJ3+>PRGP1:0 / ' ;

RESULT OF DEPENDENCY STRUCTURE GENERATION
RELDEP (1)=<(TOM)NU1 >NUGP1:0 <(FOLLOWS)VB1 >VBGP1:NUGP1*:NUGP2#
<(JIM)NU2 >NUGP2:0 <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:
AJ1+:AJ2+:AJ3+>PRGP1:0 / ' ;
RELVP (1)= 1 / ' ;
RELDEP (2)=<(TOM)NU1 >NUGP1:0 <(FOLLOWS)VB1 >VBGP1:NUGP1*:NUGP2# <(JIM)NU2
>NUGP2:PRGP1*:0 <(WITH)PR1:NU3# (A)AJ1 <(BIG)AJ2 <(RED)AJ3 <(STICK)NU3:AJ1+:
AJ2+:AJ3+>PRGP1:0 / ' ;
RELVP (2)= 1 / ' ;

```

(+,0,*,and X mean respectively intra-phrase arbitrary, intra-phrase obligatory, inter-phrase arbitrary, and inter-phrase obligatory dependants.)