九州大学学術情報リポジトリ
Kyushu University Institutional Repository

日英双方向機械翻訳システム BIJET の概要と英語処理について

吉武，春光
九州大学大学院総合理工学研究科情報システム学専攻

三池，誠司
株式会社東芝 | 九州大学大学院総合理工学研究科情報システム学専攻

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/10.15017/17616

出版情報：九州大学大学院総合理工学報告. 6 (2), pp.267-274, 1985-01-01. 九州大学大学院総合理工学研究科
バージョン：
権利関係：

日英双方向機械翻訳システム BIJET の概要と英語処理について

吉武春光*・三池誠司**・田町常夫***

(昭和59年9月29日　受理)

Bidirectional Japanese English machine translation system BIJET —The general performance and its processing part of English—

Harumitsu YOSHITAKE, Seiji MIIKE and Tuneo TAMATI

Recently researches in natural language processing are developed because of a rapid progress in software and hardware. Especially, the appearance of programming language Prolog makes remarkable improvements in the field of rule repetitive application. The authors have recently constructed an experimental machine translation system named BIJET (BIdirectional Japanese English Translation system). This system consists of three parts: a Japanese processor, a English processor and a transfer processor. Both Japanese and English processors are divided into a morphological analyser, a morphological synthesizer and a syntactic processor. A transfer processor is divided into a Japanese-English transfer processor and a English-Japanese one. BIJET is programmed in Prolog and Lisp. Especially the syntactic processors are written only by Prolog. These processors can be used as both analyser and synthesizer, because the input and output arguments of a Prolog statement do not have to be distinguished in advance. This paper discusses the general performance of BIJET and gives the details of its English processor.

1. 緒　　言

近年のソフトウェア，ハードウェア技術の進歩によりＡＩ関係の実験システムの開発・研究が容易になってきた．特にプログラミング言語 Prolog[1] の出現は，規則を繰り返し適用する分野においてシステム開発の容易さ，時間の節約などの点で見るべきものがある．自然言語処理においても構文解析に使用した例[2]が有名である．我々は従来より機械翻訳，自然言語理解の研究を進めているが，実験システムの記述言語としては PL/I や FORTRAN を使用しており，システムの処理内容が複雑になるにつれ，プログラミングに困難さが生じている．我々はこの点を考慮して日英双方向機械翻訳システム BIJET (BIdirectional Japanese English Translation system) の試作を行った．本システムは Prolog を主体に，補助的に Lisp を使用してシステムを記述したトランスファ方式による日英双方向の翻訳システムであり，Prolog のパラメータの入出力可逆性を生かしてモジュールの数を減らしている．

本システム開発の狙いは次の点にある．

(1) Prolog の種々の特徴を確認する．

(2) D-tree モデルを中間表現として採用したトランスファ方式による双方向翻訳がどの程度可能かを確認する．

なお翻訳の対象は中学校英語教科書程度の英文及びそれを人手により日本語に訳したものとしている．

本論文では BIJET の基本的機能と英語解析・生成部について述べる．

2. BIJET の基本的機能

BIJET の処理の概要を **Fig. 1** に示す．BIJET は大きく，英語処理部，トランスファ部，日本語処理部

* 情報システム学専攻博士課程
** 情報システム学専攻修士課程（現在(株)東芝）
*** 情報システム学専攻

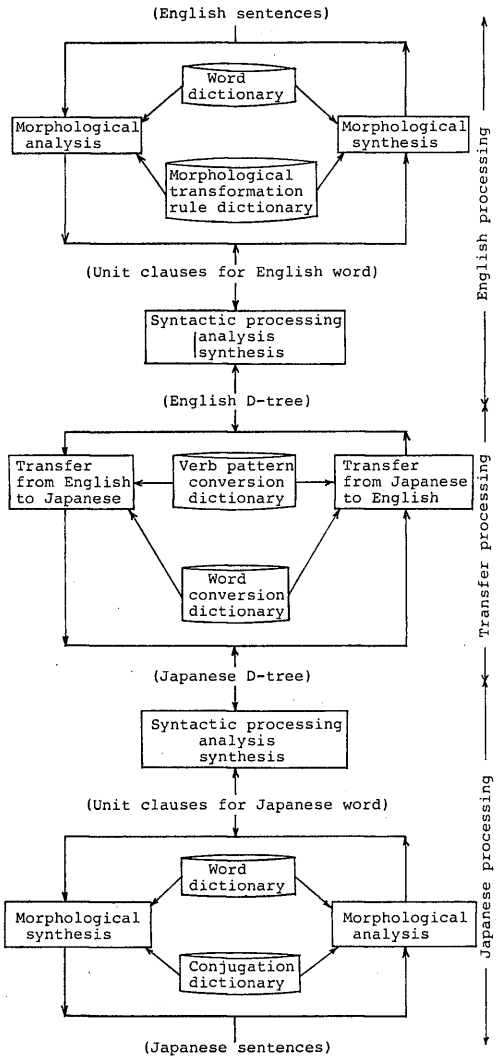


Fig. 1　Configuration of BIJET:

に分かれる．英語処理部と日本語処理部は更に各々形態素処理部と構文処理部に分かれる．BIJET は英→日翻訳と日→英翻訳という２つの動作モードを持っている．その際の処理の流れは **Fig. 1** の矢印のとおりであるが，各構文処理部は解析と生成で同一のものを使用している．以下，英→日翻訳の際の処理の概要を説明する．

BIJET は，まず入力された英文を英語形態素解析部において英単語辞書と英語形態素変形規則辞書を参照することにより単語に分解し，各英単語に対応した Prolog の宣言を出力する．そして，英語構文処理部を解析に使用して英語の中間表現である英語 D-tree を組み立てる．次に英→日のトランスファ処理部において述語機能パターン対応辞書と単語対応辞書を参照することにより日本語の中間表現である日本語 D-tree に変換する．そして，日本語構文処理部を生成に使用して，日本語単語の Prolog の宣言を出力する．これより日本語形態素生成部において日単語辞書と日本語形態素活用辞書を参照することにより日本語の文を組み立て出力する．日→英翻訳の場合はこれと逆の処理過程をとる．

3.　英語 D-tree モデルと英語依存文法

BIJET では翻訳の中間表現として英語と日本語の Dependency tree (D-tree) モデルを用いている．D-tree モデルは D. G. Hays の依存文法[3] によって組み立てた依存の木に，機械翻訳に必要な各種情報を付加したものである．本章では英語 D-tree モデルとそれを生成する依存文法について述べる．

3.1.　英語 D-tree モデル

本 D-tree モデルでは，単語の順序をその依存の木の中に保つ為に，単語・熟語を最小構文単位として依存構造を組み立てている．

(EDM1)　機械が識別すべき最小の文要素を単語・熟語の語幹，語尾とする．

(EDM2)　最小構文単位を単語・熟語とする．

(EDM3)　文の構文構造を D-tree で示す．これは機能要素又は品詞を節点 (node)，述語を根 (root)，機能要素又は品詞間の依存関係を枝 (branch) とする．

文を単語・熟語系列とみれば文脈にほとんど関係なく，近傍の単語・熟語の構文情報だけをみて文をある単位に分割できる．この単位は伝統文法の句にほぼ相当し，特別の場合を除き，構成要素の単語・熟語は連接する．この単位を語群と呼ぶ．

語群が述語に係るとき，述語語群に対するその語群の構文的役割を機能，その細分類を機能要素という，又これは必須と任意に分かれる．

文の基本的意味は，述語動詞のとる必須機能要素を並べた系列に表出される．この系列を述語機能パターンと呼ぶ。述語機能パターンは次のタイプＰ１とタイプＰ２に分けることができる．

タイプＰ１：第１種の依存関係のみを持つもの．

タイプＰ２：第１種のほかに，第２種の依存関係を持

つもの.

ただし,

依存関係　第1種：文脈に関係なく当該2語群だけをみて決定できるもの.

第2種：当該2語群が一定の文脈条件を満足しなければ決定できないもの.

なお D-tree は次の特徴を持っている.

(1) 根となる機能要素や品詞を除いて，常にただ一つの governor を持つ.

(2) 依存関係を示す枝は系列中で交差しない.

3.2. 英語依存文法

上記の D-tree を生成する依存文法 (Dependency Grammar) G_Eを形式的に次のように定義する.

$G_E=(V_N, V_T, R, A, P, S)$

V_N：非終端記号の有限集合

但し　$V_N=V_H \cup V_K$

ここで　V_H：品詞の有限集合

V_K：機能要素の有限集合

V_T：終端記号（単語）の有限集合

R：依存規則の有限集合

$R=\{X_i(X_{j1}, X_{j2}, \cdots, *, \cdots, X_{jn}) \mid X_k \in V_N\}$

ここで　X_i：governor

$X_{j1}, \cdots, X_{jn}$：dependent

*：governor の位置

カッコ内の要素はこの順序に並ぶ.

A：単語代入規則の有限集合

$A=\{X \rightarrow x \mid X \in V_H, x \in V_T\}$

P：機能要素割当規則の有限集合

$P=\{Y \rightarrow y \mid Y \in V_K, y \in V_H\}$

S：初期記号の有限集合

ただし，$S \in V_K$

この定義に従う英語依存文法 G_E が生成する D-tree を英語 D-tree と呼ぶ.

3.3. 英語の言語情報

本システムでは，前節で述べた D-tree に，更に機械翻訳に必要な情報を付加して使用している．言語情報をまとめると，次のようになる.

- 言語情報
  - D-tree
    - 単　語
    - 品　詞
    - 機能要素
    - 述語機能パターン
  - 単語情報

Table 1 Parts of speech in English

Symbol	A part of speech
DET	Determinor
PRON	Pronoun
NOUN	Noun
ADJ	Adjective
ADV	Adverb
VERB	Verb
CPLA	Copula
AUX	Auxiliary verb
PRE	Preposition
THAT	'that' of that-clause
SUBCONJ	Subordinate conjunction
COCONJ	Co-ordinate conjunction
RPRN	Relative pronoun
RADV	Relative adverb
IPRN	Interrogative pronoun
IADV	Interrogative adverb
IT	Formal 'it'
TO	'to' of to-infinitive
NOT	Not
COMMA	Comma
HERE	Introductory here
THERE	Introductory there

Table 2 Word information in English

Symbol	Word information
SIN	Singular number
PLU	Plural number
1	First person
2	Second person
3	Third person
NOM	Nominative case
POS	Possessive case
OBJ	Objective case
ROOT	Root-form
PRES	Present-form
PAST	Past-form
P.P.	Past participle form
ING	Ing-form
POSI	Positive degree
COMP	Comparative degree
SUPE	Superlative degree

これらの言語情報のうち単語，品詞，単語情報は入力文に対する形態素処理の結果により単語辞書から与えられるものである．又，動詞の場合には更に，その動詞が取りえる述語機能パターンの番号を与えている.

英語における品詞と単語情報の分類を **Table 1** と **Table 2** に示す．品詞の分類においては，語幹の構文的役割が基本的に共通か否かで大分類した．これにより動詞，名詞，代名詞，副詞及び形容詞などは伝統文法とほぼ同じ分類とした．しかし伝統文法の冠詞，人称代名詞所有格及び不定形容詞などは全て決定詞とした．更に not，コンマ及び形式主語の it 等は一語一分類とした．なお BIJET の品詞分類においては，構

Table 3 Obligatory function

Symbol	Function
SUB	Subject
SUB*	Formal subject
CLA	Copula
P	Predicator
PIV	Predicative
IO	Indirict object
DO	Direct object
AC	Adverbial complement
PC	Predicative complement
PO	Predicative object
NO	Nexus object
INO	Interrogative nexus object
NC	Nexus complement
INC	Interrogative nexus complement
CM	Comma

Table 4 Optional function

Symbol	Function
SC	Subordinate conjunction
CC	Co-ordinate conjunction
S-NM	Sentential noun modifier
P-PM	Phrasal predicative modifier
S-PM	Sentential predicative modifier

Table 5 Verb pattern type P1

Patten No.	Verb pattern	Example
P1-1	SUB+CLA+PIV	*Her name is Jane.*
P1-2	SUB+P	*Birds fly.*
P1-3	SUB+P+DO	*I read the book.*
P1-4	SUB+P+IO+DO	*I give him a book.*
P1-5	SUB+P+DO+AC	*I give a book to her.*
P1-6	SUB+P+AC	*I went to China.*
P1-7	SUB+P+DO+PIV	*I think it important.*
P1-8	P+SUB	*There is a book.*

文解析処理に必要最小限な分類としている．単語情報のうち，数，人称は文中の呼応関係を成立させるためのものである．機能要素と述語記能パターンの分類を **Table 3～6** に示す．これらの分類には KULTS[4]，QTRAN[5] のものをほぼそのまま使用している．

4. 英語解析・生成部

本章では，入力英語文に形態素解析処理・構文解析処理を施して英語 D-tree を得る英語解析過程，逆に英語 D-tree に構文生成処理・形態素生成処理を施して英語文を合成する英語生成過程の両過程について述べる．ただし，本システムでは，プログラミング言語 Prolog のパラメータの入出力可逆性を利用して，構文解析処理と構文生成処理は同一の手続きで行っているので，これをまとめて構文処理部として説明する．

4.1. 英語形態素処理部[6]

この処理部には入力英語文に形態素解析を行い，単語として識別し，形態素情報（品詞と単語情報）を付加して単語の宣言の形で出力する形態素解析処理と，単語の宣言より単語を取り出し，その形態素情報に応じて語形変形を行なって1つの文として出力する形態素生成処理が含まれる．なお，本処理部は Lisp によって記述している．

4.1.1. 入出力形態

英語形態素解析処理の入力及び英語形態素生成処理の出力は最後がピリオドである英語の平叙文である．英語形態素解析処理の出力及び英語形態素生成処理の入力は次の形式をとる単語の宣言（Unit clause）である．

Table 6 Verb pattern type P2

Pattern No.	Verb pattern	Example
P2- 1	SUB+P+NC	*I want to see you.*
P2- 2	SUB+P+NO	*I think that it is important.*
P2- 3	SUB+P+INC	*I know how to read it.*
P2- 4	SUB+P+INO	*I know where he is.*
P2- 5	SUB+P+IO+NC	*I wish him to go there.*
P2- 6	SUB+P+IO+NO	*I tell him that he must read it.*
P2- 7	SUB+P+IO+INC	*I tell him which to lean.*
P2- 8	SUB+P+IO+INO	*I tell him when I will go.*
P2- 9	SUB+P+IO+PC	*I stop him do it.*
P2-10	SUB+P+PO	*I stop swimming.*
P2-11	SUB+P+DO+PO	*I find him swimming.*
P2-12	SUB*+P+PIV+SUB (NO)	*It is important that you read it.*
P2-13	SUB*+P+PIV+SUB (NC)	*It is great to do it.*

Table 7　A part of word dictionary in English

((ALONG)	PRE	( )	( )	( ))	
((AM)	GOTOBE	( )	( )	( ))	
((ARE)	GOTOBE	( )	( )	( ))	
((BE)	CPLA	(P1-1 P2-12	P2-13)	( )	(TNB1 TNB2 ING1))
((BE DIFFERENT FROM)	CPLA	(P1-1 P2-12	P2-13)	( )	(TNB1 TNB2 ING1))
((DIFFERENT)	ADJ	( )	( )	(COM 0))	
((SHOULD	SHAL	( )	(PAST))	( ))	
((SMITH)	NOUN	( )	( )	(PUL1 GEN1))	
((SPRING)	NOUN	( )	( )	(PUL1))	
((SPRING)	VERB	(P1-2)	( )	(TN 15 ING1 TSP1))	

(ASSERT (WORD 開始位置　終了位置
　　(単語原形)　品詞　(単語情報)))

ただし，

ASSERT：PROLOG/KR の予約述語で，その引数を clause として登録することを示す．

WORT：PROLOG/KR の述語名

開始位置，終了位置：入力英語における単語の位置を示す．

(例1)　I　can　speak　English.
　　1　2　3　4　5

単語 'can' の場合　　開始位置……2
　　終了位置……3

(単語原形)：語形変化のない単語はそのままの形を，語形変化のある単語は原形を出力とする．又，連語及び熟語を1単語とみなして出力するためにリスト形式としている．

品詞：**Table 1** より品詞を1つ付与する．ただし，VERB 又は CPLA の場合には，品詞の代わりに取りうる述語機能パターンをリスト形式にしたものを与える．なお，多品詞語の場合には複数個の単語の宣言を出力している．

(単語情報)：**Table 2** より単語情報をリスト形式で与える．なお，同一品詞でも異義語の可能性がある場合には，複数個の単語の宣言を出力している．

4.1.2.　英語形態素解析処理

まず，入力英語文を単語に分割して1単語を取り出すが，英語の場合には単語間はスペースで区切ってあるので，すぐに単語として取り出すことができる．次に取り出した単語に対して，単語辞書 (Word dictionary：**Table 7**) の見出し語（以下，見出し語とだけ記述する）の検索を行う．この時，単語の頭文字を最初の比較要素とする．即ち，単語辞書の最初の項目から順に見出し語を検索し，その頭文字が一致する項目を探し出す．次に，その項目に変形規則 (**Table 8**) が与えてあれば，その規則に従って見出し語の変化形をすべて生成する．そして，検索する単語と見出し語の変化形とを比較し，一致すれば項目中の品詞及び単語情報をその検索する単語に対して付与して，単語の宣言の形で出力する．多義をつくすために，さらに項目を進めて，頭文字が一致する見出し語全部について同じ処理を行う．

Table 7 の単語辞書における各項目の構造は次のように表現できる．

((見出し語)　品詞　(述語機能パターン)
　　(単語情報)　(形態素変形規則名))

ただし，

(見出し語)：分離している語が熟語を構成する場合は，記号？を用いて次のように登録できる．

(例2)　ASK ? OF (ASK～OF の登録)

品詞：特に BE 動詞にかかわる熟語の登録の為に，GOTOBE なる単語サーチを行なう規則を設け，この項目に記述することにしている．これは，あらかじめ不定詞である be の項目中に変化形 is, am, are… などを語尾変形規則として設けておき，各 is, am, are, were… の品詞項目には GOTOBE という単語サーチのジャンプを行うルールを書いておき，無条件に be の見出し語にサーチを行なわせるものである．

(述語機能パターン)：動詞の場合だけ，その動詞が取りえる述語機能パターンの番号を与える．動詞以外の場合は（　）を与える．

(単語情報)：変形規則に従わない形態素変形をする単語に対して直接単語情報を与える．

(形態素変形規則名)：形態素変形規則辞書に登録してある変形規則名を与える．

Table 8 の形態素変形規則辞書 (Morphological transformation rule dictionary) における各項目の構造は，次のように表現できる．

Table 8　A part of morphological tranformation rule dictionary in English

(TN01	(((+ “”))	(ROOT PRES))
	(((+ “ED”))	(PAST P.P.)))
(TN02	(((+ “”))	(ROOT PRES))
	(((+ “D”))	(PAST P.P.)))
(TN04	(((+ “”))	(ROOT PRES))
	(((+LAST 1)　(+ “ED”))	(PAST P.P.)))
(TNB1	(((+ “”))	(ROOT))
	(((REMAIN 0)　(+ “IS”))	((3 SIN PRES)))
	(((REMAIN 0)　(+ “ARE”))	((2 PRES)))
	(((REMAIN 0)　(+ “AM”))	((1 SIN PRES))))
(ING1	(((+ “ING”))	(ING)))

（形態素変形規則名

　{（（形態素変形規則）　（単語情報））})

ただし，

（単語情報）：形態素変形規則によって変形した単語に付与する単語情報を登録する．

{ }：繰り返し．

（形態素変形規則）：形態素の変形規則を変形規則処理関数とパラメータを用いて表現する．

(1)　変形規則処理関数

+，−，REMAIN，+LAST，−LAST の5つの関数を用意している．

ⓐ　+

+は右側に1つのパラメータをとり，このパラメータを，与えた文字列すなわち単語辞書の見出し語もしくはそれに文字列処理関数を施した文字列の末尾に加える．

ⓑ　−

−は右側に1つのパラメータをとり，このパラメータを与えた文字列の語尾から除く．

ⓒ　(REMAIN x)

REMAIN は与えた文字列から，初めの x 個の文字を残して，その他は消去する．

ⓓ　(+LAST x)

+LASTは与えた文字列の最後の x 個の文字を与えた文字列の末尾に加える．

ⓔ　(−LAST x)

−LASTは与えた文字列の最後の x 個の文字を与えた文字列の末尾から取り除く．

(2)　関数用パラメータ

ⓐ　引用符で囲まれた任意の文字列

4.1.3.　英語形態素生成処理

ここでは，入力された単語の宣言中にある単語の原形と単語情報により単語辞書を検索し，各単語の変形規則を逆に見つける．この変形規則に従って単語の原形に変形を加えた後，一つの文としてまとめて出力する．

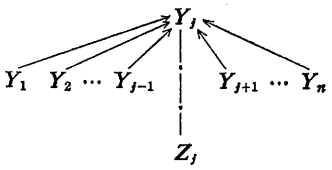
4.2.　英語構文処理部[7]

この節では，英語の単語の宣言より構文解析を行なって英語 D-tree を出力する英語構文処理過程と，逆に英語 D-tree に構文生成を行なって英単語の宣言を出力する英語構文生成過程について述べる．

4.2.1.　入出力形態

英語構文解析過程の入力と英語構文生成過程の出力は 4.1.1 項で述べた英語の単語の宣言である．英語構文解析過程の出力と英語構文生成過程の入力は第3章で定義した英語 D-tree であり，その表現は次のように記述している．

依存：$Y_j(Y_1\ \ Y_2\ \ \cdots\ \ Y_{j-1}\ \ (Z_j)\ \ Y_{j+1}\ \ \cdots\ \ Y_n)$



代入：A (B)　　割当：C (D)

この表現を使った実際の解析例を次に示す．

（例3）　I read the book.

解析結果

P3 (SUB (PRON (1)) (VERB (2)) DO (NOUN (DET (3) (4))))

(ASSERT (WORD 1 2 (I)　PRON (1 SIN NOM)))

(ASSERT (WORD 2 3 (READ) (P1-3) (1 SIN

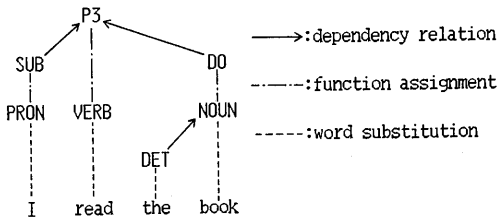


Fig. 2 The dependency tree of example 3.

PRES)))

(ASSERT (WORD 3 4 (THE) DET ()))

(ASSERT (WORD 4 5 (BOOK) NOUN(SIN)))

依存の木

なお D-tree 表現上の終端記号は，単語の開始位置で示している．又，述語 P の後に付いている番号は，述語機能パターン番号を示す．

4.2.2. 英語構文処理

英語構文処理部の処理方式は top-down, depth-first, left-to-right である．本処理部は PROLOG/KR で記述しており，これを構文解析と構文生成に用いている．構文解析処理では，単語の宣言を入力とし，品詞と単語情報によって D-tree を生成する．構文生成処理では，D-tree を入力とし，その D-tree を生成する規則をたどり，それらの規則の中で単語の宣言を出力していく．両処理とも構文規則（機能要素割当規則と依存規則）を top-down に適用するという処理過程においては全く同じである．

まず，英語依存文法 G_E の初期記号と文法規則を示す．

(1) 自然言語の文の中心は述語であると考え，初期記号の有限集合を

$$S=\{CLA,\ P\}$$

とする．

(2) 語群内では，その語群の構文的性質（機能要素）の中心となる品詞を持つ単語を governor とする．そこで機能要素割当規則には **Table 9** のようなものを用いている．

(3) 依存規則 R は次の 3 っに分けることができる．

依存規則R 1：述語機能パターンに従って機能要素の並びを検定する規則．**Table 10** に示す．

依存規則 R 2：R 1 以外の機能要素間の依存規則．**Table 11** に示す．

Table 9 A part of function assign rule in English

SUB→PRON	SUB*→PRON	PIV→NOUN
SUB→NOUN	CLA →CPLA	PIV→ADJ
SUB→IPRN	P →VERB	AC →PRE
SUB→RPRN	PIV →PRON	CM →COMA

Table 10 A part of dependency rule R1 in English

pattern 1	CLA (SUB, *, PIV)
pattern 4	P (SUB, *, IO, DO)
pattern 8	P (SUB, *, NC)
pattern 13	P (SUB, *, IO, NO)

Table 11 A part of dependency rule R2 in English

NO (*, P)
INO (*, P)
NC (*, P)
S-NM (*, P)

Table 12 A part of dependency rule R3 in English

NOUN (DET, *)
NOUN (ADJ, *)
PRE (*, NOUN)
VERB (AUX, *)

依存規則R 3：各機能要素（語群）を構成する単語の依存関係を調べる規則．**Table 12** に示す．

次に，構文処理過程は大きく次の 3 つの Step をたどる．

Step 1：依存規則 R 1 により機能要素間の並びを決定する．このとき，動詞の宣言中の述語機能パターンのリストにより，適用する述語機能パターンの依存規則を限定する．

Step 2：機能要素割当規則によって，依存規則 R 1 又は R 2 を適用する．

Step 3：依存規則R 3 によって，語群内の依存構造を検定する．語群内に埋め込み文などがあれば，機能要素 S-NM により再帰的に Step 1 以下を繰り返す．

5. 結　　論

日英双方向機械翻訳システム BIJET の概要と英語処理部について述べた．英語 D-tree の基本的な考え方は KULTS のものとほぼ同じであるが，語群を形成する単語間の依存関係を更に形成している．又，言語情報を用いて文の呼応関係を調べるなど，より細

かな処理を行なっている．BIJET の日本語処理部，トランスファ部，翻訳実験については別稿[8]で述べる．

なお，本研究の一部は昭和57～59年度文部省科学研究費　特定研究「言語の標準化」(57215006, 58207009, 59101006) の援助を受けたものである．

参考文献

1) W. F. Clocksin, C. S. Mellish: "Programming in Prolog", Springer-Verlag (1981).
2) F. C. N. Pereira, D. H. D. Warren: "Definite Clause Grammars for Language Analysis-A Survey of the Formalism and a Comparison with Augmented Transition Networks", AI 13 (1980).
3) D. G. Hays: "Dependency Theory: a formalism and some observations", LANGUAGE, Vol. 40, No. 4 (1964).
4) 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), J57-D, 7, pp. 435-442 (昭49-07).
5) 吉武，三池，田町："あいまい性指摘の能力を持つ英日機械翻訳システムQ-TRAN"，昭和57年電四九支連大 No. 731.
6) 末廣陽一："英日双方向翻訳システムの形態素処理部について"，昭和58年度，九大情報工学科卒業論文.
7) 三池誠司："PROLOG による英日双方向翻訳の試み"，昭和58年度，九大総理工情報システム学専攻修士論文.
8) 吉武，三池，田町："日英双方向機械翻訳システム BIJET の日本語処理と翻訳実験について"，九大総理工研究科報告，Vol. 6, No. 2 (昭和60年).