

## 機械翻訳システムの開発

電子技術総合研究所 西村 恕彦

この記事は、電気試験所研究報告705号に発表された  
上の標題の論文の一部である。

このほか関連した内容として、下記の2つの論文がある。

なぜFORTRANを止めたか。

数理科学 1968年12月号

記号処理用言語

電気通信学会誌 1967年4月号 50巻4号  
P 28 ~ 532

紙数の関係で、この論文の再録は割愛した。

5.3 接続詞のあいまいさ (しかし大きな)

例3. 次のような英文を解析し、日本語に翻訳します。

That's one small step for a man, but one giant leap for mankind.

これは実はたいへんめんどろな問題をふくんでいます。まず最初に一つの解を示します。その次にそれがどういう点でぐあいかわるいか検討します。

辞書を追加します。

- , but → 接続詞 CC (“、しかし”)
- giant → 名詞 NN (“巨人”)
- leap → 名詞 NN (“飛躍”)

それから That's の項をきちんとします。

- that's → that\_is
- \_is → 動詞be (“である”)

こういうやりかたを、文字列の書き換え (rewriting) といいます。

文法表では、第010500行の規則を変更します。

動詞be+名詞句+終止符 → 述語+終止符

```

*****
BEGINLEX -----DL DELIMITER NISIMURA
120700 A -----AA0000000000 NULL NISIMURA
ALL-----AA8G988L8U00 SUBETENO NISIMURA
+ BUT -----CC2)8F8+8F00 SIKASI NISIMURA
CAME-----VV0,8-000000 KITA 0 5 8 69-9-11
EARTH-----NNKOGH000000 TIKYUU 0 2 8 NISIMURA
E2-----ME MOON 0000 TUKI NISIMURA
FIRST-----DDJANF8/0000 SAISYO NI NISIMURA
FOOT-----NNO 00000000 ASI - 2 8 5 8 NISIMURA
120600 FOR-----PP8/8M948L00 NITOTTE NISIMURA
FROM-----PP8+9V8U0000 KARANO NISIMURA
GIANT-----NNGJPS000000 KYOZIN NISIMURA
HO-----ME SILVER 00 GIN - 2 8 NISIMURA
H-----ME GOLD 0000 KIN NISIMURA
HERE-----DD8D8D8/0000 KOKONI NISIMURA
IN -----PP8U*08/0000 NONAKANI - 2 8 NISIMURA
IS -----BF9 8 9 0000 DEARU 69-9-12
LEAP-----NNVM02000000 HIYAKU 0 2 8 NISIMURA
120800 MAN-----NNPS00000000 HITO NISIMURA
MANKIND-----NNPS(4000000 ZINRUI NISIMURA
MEN-----NNPS00000000 HITO NISIMURA
MOON-----NNE200000000 TUKI NISIMURA
120300 ONE-----AAA/8K8U0000 HITOTU NO NISIMURA
PEACE-----NNX9(W000000 HEIWA NISIMURA
PLANET-----NN(Z00000000 WAKUSEI + 2 8 NISIMURA
120200 'S -----BE9L829X0000 DEARU NISIMURA
SET-----VV*J8B0000000 OKU NISIMURA
120400 SMALL-----AANP8E860000 TIISAI NISIMURA
120500 STEP-----NNX08W000000 AYUMI NISIMURA
120100 THAT-----HH819Y000000 SORE NISIMURA
THAT'S-----RWTHAT IS - REWRITE THE CHARACTER STRING L69-9-12
THE-----AA00000000000 NISIMURA
IT8K-----ME TWO000000 HUTATU NISIMURA
UPON-----PP8UOR8/0000 NO UE NI NISIMURA
WE -----HH859Y859Y00 WAREWARE NISIMURA
69-9-11

```

5.3.1 辞書

又

00000000111111112222222233333333444444445555555566666666777777778  
1234567890123456789012345678901234567890123456789012345678901234567890

\* APOLLO 1969-7-21 ETL YAMATO NISIMURA  
CONTROL -----XXDLRWUN NISIMURA  
\*\*\*\*\*

5  
3  
2  
団  
文  
法  
表

010100	HH	NP	01P	PRONOUN	NISIMURA
010200	AANN	NP	02P	NCUN PHRASE	NISIMURA
010240	NPNN	NP	02P	NCUN PHRASE	NISIMURA
010260	NPCCNPXX	NP	03P	CONJUNCTION	NISIMURA
010300	AAAA	AA	02P	ADJECTIVES	NISIMURA
010400	PPNP	DD	02R	PREPOSITIONAL PHRASE	NISIMURA
010500	HENPXX	PR	02R	BE PREDICATE	NISIMURA
010600	PRDD	PR	02R	ADVERB	NISIMURA
010700	HPPRXX	CA8NOG00	10I	CASE PARTICLE ('HA')	NISIMURA
010800	NPCAPR	SE	03P	SENTENCE	NISIMURA
010900	SEXX	SX	02P	COMPLETE SENTENCE	NISIMURA
013100	NPPNP	NP	03R	PREPOSITIONAL PHRASE	NISIMURA
013200	NPPNPNN	NP	22P	NOUN PHRASE	NISIMURA
013300	NN	NP	01P	NOUN	NISIMURA
013400	VVNPXX	CA830000	10I	CASE PARTICLE ('WO')	NISIMURA
013500	VVCANPXX	PR	03R	TRANSITIVE VERB PREDICATE	NISIMURA
013600	DDVV	VV	02P	ADVERB	NISIMURA
013700	DDSE	SE	02P	ADVERB	NISIMURA
013800	VVDD	VV	02R	ADVERB	NISIMURA
013900	VVXX	PR	01P	VERB PREDICATE	NISIMURA

010500 BENPXX P R 02R

そのほかに、規則を追加します。

名詞句+名詞 → 名詞句

名詞句+接続詞+名詞句+終止符 → 名詞句+終止符

NPNN NP 02P

NPCCNPXX NP 03P

これらの記述表を、第5.3.1図と第5.3.2図に示します。

これで実験してみます。第5.3.3図に示すとおり訳文は、「これは人にとって一つの小さい歩み、しかし人類にとって一つの巨人飛躍である。」となりました。問題は接続詞 but のところです。

THAT'S ONE SMALL STEP FOR A MAN, BUT ONE GIANT LEAP FOR MANKIND.

それは人にとって一つの小さい歩み、しかし人類にとって一つの巨人飛躍である。

5.3.3 図 入出力文

一般的にいって、代名詞や接続詞の処理は文法的翻訳では非常に困難とされています(前報告80, 95, 100, 125ページ参照)。この場合には正しい答をわれわれが知っておりますので、次のような正しい解析をシステムにやらせました。

one small step for a man , but one giant leap for mankind

人にとって一つの小さい歩み , しかし 人類にとって一つの巨人飛躍

しかし接続のしかたはこれ以外にもあります。いずれも二つの名詞句をつないでおります。

one small step for a man , but one giant leap for mankind

人類にとって 人, しかし一つの巨人飛躍 にとって一つの小さい歩み

one small step for a man , but one giant leap for mankind

人類にとって 人にとって一つの小さい歩み , しかし 一つの巨人飛躍

one small step for a man , but one giant leap for mankind

人 , しかし 人類にとって一つの巨人飛躍 にとって一つの小さい歩み

これらはいずれも句構造文法の機械的適用としては正しい解析です。しかしちょっと常識のある人間ならば、あとの三つの解析、翻訳をまったくなんせんすなものとするでしょう。人間にとってあきらかに常識的なことが、計算機にできないわけはありません。

人間の頭のなかでおこっている解析の過程を計算機で模倣させるようにして、ヤマトシステムの能力を拡張したいと筆者は考えています。ただし、生体の機能や動作をそっくり真似することは意味がありません。計算機向きの、別のやりかたをとります。しかしヤマトシステムにはさしあたりこういう能力はありませんから、さきのようないんちきなやりかたでがまんするしかありません。

## 5. 4 和文英訳

例4. 次のような日本語を解析し、英語に翻訳します。

月の砂漠を、はるばると旅の駱駝がゆきました。

金と銀との鞍置いて、二つならんでゆきました。

漢字テレタイプライタの漢字はだいたい当用漢字にしたがっていますので、上の文はそのままでは計算機には入れられません。それで、一部の文字の表現を変えます(翻字)。

月の砂ばくを、はるばると旅のラクダがゆきました。

金と銀とのくら置いて、二つならんでゆきました。

漢字テレタイプライタの鍵盤をこのとおりに打鍵すれば、穿孔された紙テープが得られます。

その紙テープを計算機に入れることにします。もとの文を単語単位に分かち書きして入れてやる必要はありません。ヤマトシステムは自動分かち書き (automatic segmentation) の機能をもっているからです。つまり、辞書に語を登録しておけば、その登録された語にしたがって原文の文字列が切断、識別されるのです。テキストの語のあいだには何も区切り記号がなくてもよいのです。

辞書を書きましょう。

月	→ 名詞	ME (moon)
の	→ 所有格助詞	NO (of)
砂ばく	→ 名詞	ME (desert)
を	→ 目的格助詞	WO (空)
,	→ 空	DL
はるばると	→ 副詞	HU (far away)
旅	→ 名詞	ME (travel)
ラクダ	→ 名詞	ME (camel)
が	→ 主格助詞	GA (空)
ゆき	→ 動詞	ZI (go)
ました	→ 助動詞	ZY (did)
。	→ 終止符	XX (. )
金	→ 名詞	ME (gold)
と	→ 接続助詞	TS (and)
銀	→ 名詞	ME (silver)
くら	→ 名詞	ME (saddle)
置いて	→ 他動詞	TA (set)
二つ	→ 名詞	ME (two)
ならんで	→ 動詞	ZI (queue)

和文英訳の文法を次のように与えます。まず名詞に関して。

74

名詞+所有格助詞+名詞 → 名詞 (訳は逆順)

名詞<sub>1</sub>+接続助詞<sub>2</sub>+名詞<sub>3</sub>+接続助詞<sub>4</sub>

→名詞<sub>1</sub>+接続助詞<sub>2</sub>+名詞<sub>3</sub>+挿入詞 (空)

→名詞

名詞+主格助詞 → 主格

名詞+目的格助詞 → 目的格

MENOME	ME	03R
METSWETS	**000000	31W
METSME**	ME	04P
MEGA	SK	02P
MEWO	MK	02P

次に、動詞の前の主格、目的格、副詞などの語順が日本語では自由で、英語では一定なのを調整します。

目的格+主格 → 主格+目的格

目的格+副詞 → 副詞+目的格

主格 +副詞 → 副詞+主格

MKSK	SKMK	02X
MKHU	HUMK	02X
SKHU	HUSK	02X

これらの文法によって、もとの日本語はほぼ次のような語順に規格化されます。

(はるばると) (旅のラクダが) (月の砂ぼくを) (ゆき) (ました) (。)

この語順から英文の語順を作る文法は次のとおりです。

目的格+動詞 → 動詞 (訳は逆順)

動詞+助動詞 → 動詞 (訳は逆順)

主格+動詞+終止符 → 文+終止符

副詞+文 → 文

MKZI	Z I	02R
ZIZY	Z I	02R
RKZIXX	BU	02P
HUBU	BU	02P

助動詞のところが、

× (月の砂ぼくを) (ゆきました)

× 目的格+ (動詞+助動詞)

ではなくて、

○ (月の砂ぼくをゆき) (ました)

○ (目的格+動詞)+助動詞

となっていることに注意してください。ふつうの国文法や英文法では、こういう解析はしないと思いますが、右方向最長一致で、英文和訳をするときには、これで良さそうです。この方がいくぶん合理的な面もあるのです。

それから、主格動詞を文とする場合に、無条件にするのではなく、右側に終止符がきたときだけ、句構造で結合しています。こうしないと、終止符ではなく、助動詞がきたときに、英語の語順に展開できなくなります。

あとの文はずっとやっかいです。「くら置いて」と「二つならんで」とはまったく似た形式ですが、前者の名詞は目的格、後者の名詞は主格としないと、英文にできません。これらの格の判定は、動詞が自動詞か他動詞かでつけることにします。

名詞+他動詞 → 目的格(名詞)+他動詞

名詞+動詞 → 主格(名詞)+動詞

META MK 01 P

MEZI SK 01 P

そして、

目的格+他動詞 → 動詞(訳は逆順)

MKTA Z I 02 R

これらによってもとの文はだいたい次の形になります。

動詞+主格+動詞+動詞+助動詞

これを英文らしい形にするには、次のように変換するのがよいでしょう。

主格+助動詞+動詞+接続詞+動詞+接続詞+動詞

主格と接続詞を処理します。

動詞+主格 → 主格+動詞

動詞+動詞 → 動詞+接続詞+動詞

→ 動詞

ZISK SKZI 02 X

ZIZI SZ\_and00 10 I

ZISZZI Z I 03 P

これで次のようになります。

主格+動詞+助動詞+終止符

あとは前の文にあった助動詞の文法で処理できます。

なんだか、われながらいんちきくさい気がしないでもありませんが、実験してみますと、第5.4.1図、第5.4.2図のとおり、和文英訳ができます。確認しますと、入力原文と出力訳文とは次のとおりです。

入力原文：

月の砂ばくを、はるばると旅のラクダがゆきました。

金と銀とのくら置いて、二つならんでゆきました。





です。

```
E 2      → 月
8 U      → の
)Y 9 N 8 B → 砂ぼく
8 3      → を
2)       → ,
```

追跡印刷の上半分の左側が、自動分かち書きによる辞書引きの結果です。

```
E 2      ME moon
8 U      NO of
)Y 9 N 8 B ME desert
```

などと表示されています。追跡印刷の下半分が、文法解析の過程です。右の方に文の品詞列が順次に印刷されます。

この例は、それほど真面目に和文英訳をやろうというものではありません。活用語尾の処理ひとつをとっても、まったく省略したままです。しかし、(1) ヤマトシステムが特定の国語に依存しないメタ文法を基礎において作られていること、(2) 自動分かち書きの機能をもっていること、(3) 句構造文法などのメタ文法が、英語の解析にも日本語の解析にも同じように使えること、などは示せたと思います。

もちろん、ちゃんとした和文英訳の実験をするつもりでしたら、文法の体系にせよ、辞書にせよ、もっと慎重に考えて、全体的に構成しなければなりません。

## 5. 5 数式翻訳 (ポーランド変換)

ヤマトシステムは、英文和訳、和文英訳などのいろいろな自然語の処理に使えるばかりでなく、人工語の処理に使うこともできます。人工語にもいろいろありますが、ここでは多くのプログラミング言語のなかに使われている表現をとりあげてみます。

例5. 算術代入文を読んで、次のようにポーランド記法に変換して出力する。ただし、演算子は四則演算だけ、オペランドは単純変数と定数だけとする。括弧は任意に使われる。

入力文：

```
delta=beta*beta-4.0*alpha*gamma;
y=q=(a*c+b*d)/(c*c+d*d)+i*
((a*d-b*c)/(c*c+d*d));
```

出力文：

```
beta_beta*_4.0_alpha*_gamma*_delta=;
a_c*_b_d*_+_c_c*_d_d*_+_i_a_d*_
_b_c*_+_c_c*_d_d*_+_y=q=;
```

これは、記号処理あるいは記号計算と呼ばれる分野での、いわば標準問題です。大駒, 中西 (19

69) では KLISP, Bobrow (1968, 27~52ページ) では IPL-V, LISP1.5, DYSTAL, LISP2, L<sup>o</sup> CORAL, AMBIT, COMIT, SNOBOL, PANON, CONVERT, FLIP, COGENT での実行例が報告されています (AXLE は良い言語ですが, 処理ルーチンが開発されていないので, 報告されていません)。

さてヤマト語でこの問題を解くと, 第 5.5.1 図のようになります。第 020200~02100 行が文法表, beginlex 行~endlex 行が辞書です。これによって, 入力文がどのように解析されていくかをみましょう。

* CONTROL	POLISH SYNTAX	CONVERSION	XXDLRWID	581-0441-342	NISIMURA	69-6-4	NISIMURA
*****	*****	*****	*****	*****	*****	*****	*****
020200	IDMU	PR	12R	ID * ID	REVERSE		NISIMURA
020300	IDAD	PR	12R	ID + ID	REVERSE		NISIMURA
020400	IDADIDMU		20S	ID + ID *	SKIP		NISIMURA
020500	IDADIDPR		20S	ID + ID *	SKIP		NISIMURA
020600	IDEQ	ST	02P	ID =	PHRASE		NISIMURA
020700	STIDEQ	ST	03P	ID = ID =	PHRASE		NISIMURA
020800	STIDXX	AS	02R	ID = ID **,	REVERSE		NISIMURA
020900	LEIDRI	ID	03P	( ID )	PHRASE		NISIMURA
021000	IDPR	ID	02P	ID ID +	PHRASE		NISIMURA
*****	*****	*****	*****	*****	*****	*****	*****
BEGINLEX		DL		SPACES ARE SIMPLE DELIMITER			NISIMURA
044200	+	AD4+00000000	+	ADDING OPERATOR			NISIMURA
044300	-	AD4-00000000	-	ADDING OPERATOR			NISIMURA
044400	*	MU4*00000000	*	MULTIPLYING OPERATOR			NISIMURA
044500	/	MU4/00000000	/	MULTIPLYING OPERATOR			NISIMURA
044600	**,	XX5**00000000	**,	STATEMENT SEPARATOR			NISIMURA
044700	(	LE0000000000	(	LEFT PARENTHESIS			NISIMURA
044800	)	RI0000000000	)	RIGHT PARENTHESIS			NISIMURA
044900	=	EQ4=00000000	=	ASSIGNMENT OPERATOR			NISIMURA
ENDLEX		DL		SPACES ARE SIMPLE DELIMITER			NISIMURA
*****	*****	*****	*****	*****	*****	*****	*****
BEGIN-TEXT		*****		*****			*****

### 5.5.1 図 ポーランド変換

辞書引き 入力テキストの文字列は, 辞書の見出しと照合されて, 切断され, 品詞の列に書き換えられます。ここで辞書には, 演算子 (特殊記号) の類だけが登録されていて, 英字や数字や小数点の類は登録されていません。したがってこれらの英字などの列は, すべて未知語になります。

未知語の文字列は, 前後を登録語の文字列で区切られることを思い出してください。だから, 二つの演算子記号ではさまれた英字などの列は, 一つの未知語とされ, パラメータ行で宣言された未知語品詞 ID を与えます。ID は Identifier (変数名) のつもりです。

入力文字列から文の品詞列への書き換えは, 次のようになります。セミコロンにたいして, 文末品詞が辞書で与えられていることに注意してください。

入力文字列:

$$\text{delta} = \text{beta} * \text{beta} - 4.0 * \text{alpha} * \text{gamma};$$

$$y = q = (\text{a} * \text{c} + \text{b} * \text{d}) / (\text{c} * \text{c} + \text{d} * \text{d}) + i * ((\text{a} * \text{d} - \text{b} * \text{c}) / (\text{c} * \text{c} + \text{d} * \text{d}));$$

辞書による切断:

$$|\text{delta}| = |\text{beta}| * |\text{beta}| - |4.0| * |\text{alpha}| * |\text{gamma}|;$$

$$|y| = |q| = |(\text{a} * \text{c} + \text{b} * \text{d})| / (|\text{c}| * |\text{c}| + |\text{d}| * |\text{d}|) + |i|$$

$$|*| (|(\text{a} * \text{d} - \text{b} * \text{c})| / (|\text{c}| * |\text{c}| + |\text{d}| * |\text{d}|))|;$$

品詞列:

ID-EQ-ID-MU-ID-AD-ID-MU-ID-MU-ID-XX

a = b \* b - 4 \* a \* g ;

ID-EQ-ID-EQ-LE-ID-MU-ID-AD-ID-MU-ID-RI-

y = q = ( a \* c + b \* d )

MU-LE-ID-MU-ID-AD-ID-MU-ID-RI-AD-

/ ( c \* c + d \* d ) +

ID-MU-LE-LE-ID-MU-ID-AD-ID-MU-ID-RI-MU-

i \* ( ( a \* d - b \* c ) /

LE-ID-MU-ID-AD-ID-MU-ID-RI-RI-XX

( c \* c + d \* d ) ) ;

文法処理 文の品詞列は文法表の見出しと順々に照合され、書き換えられます。たとえば最初の算術代入文で、はじめに照合、書き換えされる数ステップを示すと、次のとおりです。第5.5.2図を参照してください。

(1) “delta=”に相当する ID-EQ が照合されて、一つの句 ST に書き換えられます。ここでは、書き換え部分は照合部分と一致します。

(2) “beta\*beta”に相当する ID-MU-ID が照合されて、1つの ID に書き換えられます。ここで、

beta\*beta → ID(beta beta \*)

1 2 3            1 3 2

という語順にしたいのです。ヤマト語では、1-3-2というような語順の句を1回で作ることはできませんから、次のように、2回に分けます。

beta\*beta → beta PR(beta \*)

→ ID(beta beta \*)

(3) “ID<sup>(4.0)</sup>\*”に相当する ID-AD-ID-MU が照合されます。“4.0”の右側の演算子のほうが結合の強さがあるので、左側の部分を「飛び越し」ます。そして、次の“4.0\*alpha”に相当する ID-MU-ID の照合にすすみます。このあたりのことは、[表2.8.3の項にも説明があります。

(4) ID-MU-ID の処理は、020200行と021000行との2段で記述されていました。

ID-MU-ID → ID-PR(ID-MU)

ID-PR → ID(ID-ID-MU)

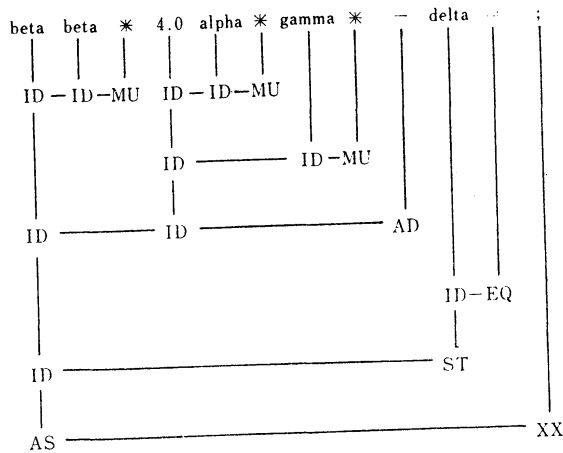
ところが(3)のようなステップの次には、これらの規則がそのまま適用されることはありません。といいますのは次のような品詞列と書き換えとをみてください。

ID-AD-ID-MU-ID

↓  
ID-AD-ID-PR

このときに、左側の ID-AD-ID という結合よりも、右側の ID-PR の結合のほうが強いのです。このことを忘れて、021000行の規則で安心していきますと、





5.5.4図 訳文のリスト表現

文字列の展開 解析のおわった文のリスト構造は、上から下へ、左から右へ、順次に語をたどって訳語文字列が展開されます。辞書に登録されていた語（演算子の類）は、辞書で指定してある文字列が出ます。

原語	品詞	訳語 (以部符号)	訳語 (印字)
+	AD	4+	+
*	MU	4*	*
,	XX	5,	;
(	LE	(空)	(空)

括弧は、文法解析にあたっては本質的に重要なので、固有の品詞を与えられますが、訳語文字列では、空となって、まったく表現されません。数式をポーランド記法で表現することの利益の一つは、括弧を使わない (Parentheses free) で、演算の優先順序を表わせることだとされています。

辞書になかった語（変数名）については、その文字が漢字テレタイプ符号に変換されて、出力されます。いちいちの語の先頭には、1字の空白が添えられて、語と語とのあいだを区切るようになっています。

注意 日本語の語順は、基本的には逆ポーランド記法にしたがっている部分が多い (水谷, 1968) から、次のような、数式→日本語の翻訳も可能であろう。

例題: 左から右へ順次に計算可能な式を、次の例のように日本語に直すプログラムを書け、  
 :  $Y = (A * X + B) * X + C;$   
 : A, X, B, X, Cを足してYとする。

あるいは、英文→数式という翻訳も可能であろう。

例題: 次のような英文を連立方程式に直すプログラムを書け (清水ほか, 1962)。

: There are two numbers. One-fifth of their sum is 12, and one-quarter of their difference is 2. Find the numbers.

記号微分で代表されるような記号計算については、FORMACをはじめとして、実用的な処理例もかなり蓄積されているようである。ヤマト語でこれらができるかどうか試みてみたが、結論は、現状のままではまず不可能だということになった。おもな難点は次の三つである。

- (1) 分配律が表現できない。
- (2) 一般的な値が扱えない。
- (3) 四則演算ができない。

まず、(1) 分配律についてであるが、記号計算でしばしばお目に掛かる

$$\begin{aligned} a*(x+y) &\rightarrow a*x+a*y \\ (f*g)' &\rightarrow f'*g+f*g' \end{aligned}$$

などの式は、オペランドや演算子が、書き換えの結果、重複して現われる性質をもっている。これらは、記号計算における分配律と呼べよう。一般の英文和訳などでは、こういう記号の分配の機能はほとんど必要なかったので、ヤマト語からは除外したが、これを用いないで上のような式を展開することは、非常にめんどろである。

次に、(2) 一般的な値については、文法表の見出しとの照合にも関係する。ヤマト語では、語や句の品詞をある一つの句の品詞で書き換えると、以後はその新しい品詞の値だけしか照合できなくなる。もとの語や句の品詞の値を、こっそりのぞいて見るということは不可能である。ところが記号計算では、

$$a*a$$

という文字列を品詞の列に書き換えて、

$$\text{ID}\{a\}*\text{ID}\{a\}$$

として、かつその二つの変数名が同じものだから、

$$\text{ID}\{a\} \uparrow 2$$

と二乗の形に書き換えるというような、書き換えの層のレベルを上がったり、下りたりする機能が必要である。

もちろん、記号計算の途中で数値計算（とくに定数だけでも）を行なう場合にも、句の品詞のなかをのぞいて、語の文字列の値を参照しなければならない。

この機能と関係があるが、(3) 四則演算もまた、記号計算の一部で必要とされる。ここでたとえば、

$$\begin{aligned} 1^+ &\rightarrow 1 \\ 2^+ &\rightarrow 3 \\ 3^+ &\rightarrow 4 \\ 9^+ &\rightarrow +0 \end{aligned}$$

というような書き換え規則で、整数の計算は記述できるだろうが、まったく実用的でない。

これらの機能をヤマト語に追加すれば、記号計算の一応用として、定理の証明なども可能であろう。