179　統語情報を用いて文のわかち書きを行なうプログラム。

田中穂積
（電子技術総合研究所）

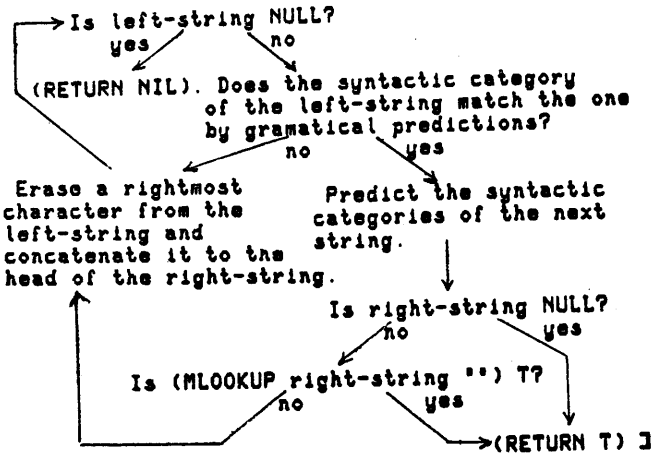
§1. はじめに　西欧語の場合には，単語ごとに区切って分かち書きすることが，長い間の慣習として固定し，それが正書法として確立されている。日本語の場合には，漢字と「かな」の使用により，語が連続していても区切りがつけ易く文の理解にさほど大きな影響を及ぼさないということもあり，分かち書きは，あまり行なわれなかった。計算機により日本語を処理する場合には，辞書の記載項目は単語単位であるから，連続した文字列のなかから，いくつかの辞書記載項目を識別しなければならない。これが，分かち書きの問題である。分かち書きの最も一般的な手法は，最長一致により辞書記載項目を識別してゆくことである。この手法のみでは，分かち書きを自動化するプログラムとして不十分であることはよく知られている。（これについては，§3で考察する。），漢字かなまじり文の表記を許すことにより，問題が解消することもあるが，それでもうまくゆかない例もある（たとえば文献(1)参照）。

本稿では，最長一致の原則だけでなく，文法規則から得られる統語情報を用いながら，分かち書きと同時に解析木をも作り出すプログラムを作成したので，その概要と動作例とを示す。統語情報の利用には，MITのPratt[2][3]の開発したプログラムを使用した。なお，このプログラムは，Augmented Contest Free Parser である。

§2. 分かち書きを自動化するプログラムの必要性　計算機に日本語を入力する場合，人に分かち書きさせて入力させるシステムが考えられる。しかし，その場合であっても，用言の活用変化に関し，用言を語幹と語尾に分離する操作が必要になる。これを避けるために，各用言毎に辞書記載項目の一部として活用変化表をもたせるシステムもあるが，メモリの使用量は増す。一方，わかち書きの形式自体，単語で切る（あまり実用的でないといわれている），文節で切る（調辞の分離が必要になる），両者の折衷法などあるが，単語や文節自体，学説によりまちまちで定説にない。我々が日本語文を書くときには，分かち書きしないことが普通であり，計算機に日本語を入力するとき，分かち書きの特定の形式を強制することは，必ずしも自然であるとはいえない

§3. 統語情報の利用について　統語情報の利用により，従来不可能であった分かち書き

Fig. 1. Main flow of the program for the separation of words.

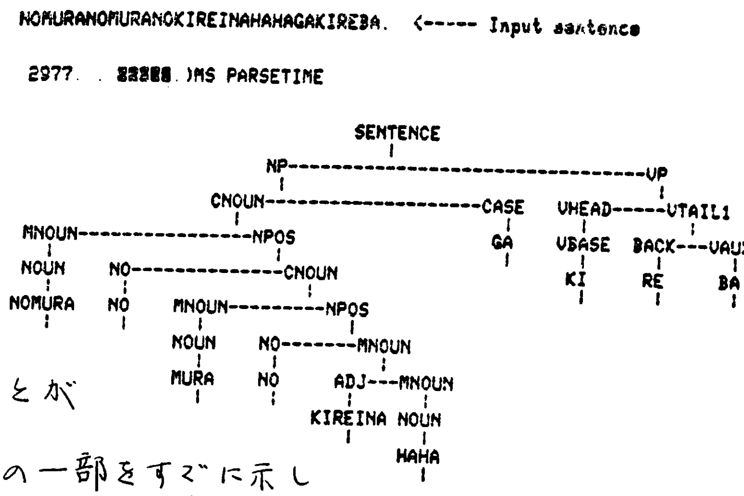


が計算機により可能になることをFig. 2の例につき説明する。　プログラムは，左から右へと最長一致により辞書記載項目を探す。これにより最左端にNOMURAを認識する。ここで，同時に，文法規則から，次の単語の文法的カテゴリーを予測することができる。Fig. 2. の例の文法規則からは，それはCaseであることが予測される。　オ２の辞書記載項目を最長一致で探すと，これもNOMURAであることがわかる。　しかし，NOMURAの文法的カテゴリーはNounであり予測と一致しないので，第２のNOMURAのサブパートのなかを最長一致で認識する。　このBack upにより，オ２のNOMURAのなかのNOがそれに合致することを見出す。　以下順に，このようにして正しい分かち書きを自動的に得ると同時に，解剖木をも導出することができる。　プログラムの概略をFig. 1. に示す。　これから明らかなように，本プログラムは，統語情報を用いながら左から右へと文字列をスキャンしてゆくのが特長である。　比較的文末に位置することの多い動詞を認識して，右から左へスキャンしてゆくシステム(4)もあるが，そうする必要はない。　人間の行なう文の理解の仕組みは，むしろ，本稿で述べたような，左から右へ，統語情報を用いながら分解してゆく方式に近いものと想像される。心理学的な根拠を考察してみることも有意義であろうと思われる。

§4 プログラムの動作例

プログラムの動作例をFig. 2に示す。　これは，計算機からの出力である。　ここで特に注意してほしいことは，辞書には，「着る」(又は「切る」)の語幹KIと，「切れる」の語幹KIREが記載されていることである。　したがって，この例では，２ヶ所で最長一致による誤りが生じ，統語情報によりrecoverされていることがわかる。

Fig. 2. An example.



§5 おわりに

Fig. 2に，その一部をすでに示したように，本稿で述べたプログラムを，用言の活用処理用として利用できる。　また音声認識用として，音素系列から文を推測する場面に応用することも考えられる。

（謝辞）　このプログラムは，Prattの開発したLINGOL(2)(3)をベースにしている。快く筆者に，プログラムのリストを与えられたPratt教授に深謝する。また，常日頃，有益な助言を与えて下さる渕一博推論研究室長と，電総研のLISP作成者でプログラムのデバッグとスピード・アップに関し，協力して下さった東芝総研黒川利明氏，LINGOLプログラムの解読に協力していただいた後藤典孝氏に深謝する。

参考文献

(1) 西村恕彦：“日本語構文解析やまと”，情報処理学会 CL2-2 (1975, 7. 25)
(2) V. R. Pratt: “A Linguistic Oriented Programming Language”, 3IJCAI, 372-381 (1973)
(3) V. R. Pratt: “LINGOL − A Progress Report”, 4IJCAI, 422-428, (1975)
(4) 長尾真他：“自然言語のためのプログラミング言語PLATON，情報処理，15, 9, 654-661 (1974)

137

文法解析システムにおける予測制御機構について.

田中穂積, 佐藤泰介, 元吉文男
(電子技術総合研究所)

§1. はじめに. より大きな言語理解システムを作成するためには, 柔軟で効率的なパージング・システムが必要になる. 我々がベースにしたシステムは, Prattの作成したLINGOLである.(1) その使用経験から, パージング・システムとして, 特に機能拡張をはかる必要性が感ぜられたものは, つぎの2点である. (1) 形態素分析の強化, (2) 直接的な予測制御.

拡張にあたり, LINGOLの文法記述形式をそのまま保存し, しかも, Prattのパーズ・アルゴリズムの高速性を生かす, ということを前提にした. (1)については前回,わかち書きを自動化するプログラムについて報告したが,(2) このプログラムが形態素分析にそのまま応用できる. 今回報告する(2)の拡張とともに, CF(文脈自由)文法規則的な記述により, かなり複雑な形態素分析が行なえる.(3)

§2. 予測制御の必要性. 予測制御の必要性を理解するために, つぎのような規則(R1)を, CF文法規則で記述してみる. (R1)---"格助詞「は」,「が」には, いかなる格助詞も後続しない." LINGOLのパーザは, 全てのCF文法規則は無条件に適用可能であると考えている.(4) したがって規則(R1)はつぎのように記述される. 格助詞$_1$ = {が, は}, 格助詞$_2$ = {を, に, へ, ……}として,

格助詞 → 格助詞$_1$, 格助詞 → 格助詞$_2$
格助詞$_2$ → 格助詞$_2$ + 格助詞$_2$, 格助詞 → 格助詞$_2$ + 格助詞$_1$ } ---(R1)'

これから, 格助詞に2つの文法カテゴリを設けたためにCF文法規則数が増大することが理解できる. 助詞の相互承接をさらに精密に規則化しようとすれば, 多数の(作為的な)文法カテゴリと, 多数の文法規則が必要になる.

これを避けるためには, 助詞の相互承接関係をチェックするプログラムを起動して, パーザがCF文法規則の適用の可否を決めればよい. すなわち, CF文法規則を条件付きにするのである. このような条件付きのCF文法規則により, 規則(R1)は, ただ1つの規則(R1)''にすることができる. (最終的な形式は§4の(R1)'''に示す.) 格助詞 = {が, は, を, に, へ, ……} として.

格助詞 → 格助詞 + 格助詞 [ただし「が」,「は」を除き適用可] ---(R1)''

[ ]内は, 左のCF文法規則の適用の可否を決める条件である. このように, CF文法規則に対し, 規則適用禁止条件を付加できるように拡張することにより,

(1) 不必要な部分文法解析木の生成を禁止して, それに伴なう不必要な文法カテゴリの予測を排除できる. これにより適切な予測が可能になる.

(2) このような予測制御により, CF文法規則数を制限し, パージングの高速化をはかることができる. また適切なわかち書きの自動化が行なえる.

非公式な説明を終り, 以下に実際の予測制御機構について述べる.

§3. 予測制御機構. 条件付きCF文法規則の記述形式はつぎのようである.

CF文法規則$_i$ [e_i]; ここでe_iは任意のS式である. ----(1)

パーザの基本動作をLISP的に記述するとつぎのようである.

(COND ((NOT (EVAL e_i)) <CF文法規則$_i$を適用>)
(T <CF文法規則$_i$の適用を禁止>)) } ----(2)

e_i が NIL なら，無条件に CF 文法規則 $_i$ が適用されることになる．

§4 予測制御用基本関数

(i) pass-message : (pm <message>)

適切な予測制御を行なうために，パージング制御のメッセージを送る．関数値は常に NIL．

(例1) A → B [(pm 'x)]

ノード B が成立すれば，無条件にノード A が成立する．この時メッセージ x が B より A に送られる．

(ii) query-message : (qm <message>)

関数 pm により送信されたメッセージが，そのノードにきているかを検査する．きていれば関数値は T，きていなければ NIL．

(例2) A → B [(qm 'x)]

ノード B が成立し，ノード B にメッセージ x が ない 時に限り，ノード A が成立する．

(例3) A → B + C [(qm 'x)]

ノード B が成立し，ノード B にメッセージ x が ない ときに限り，C を予測し，予測文法規則を A → B + C に．

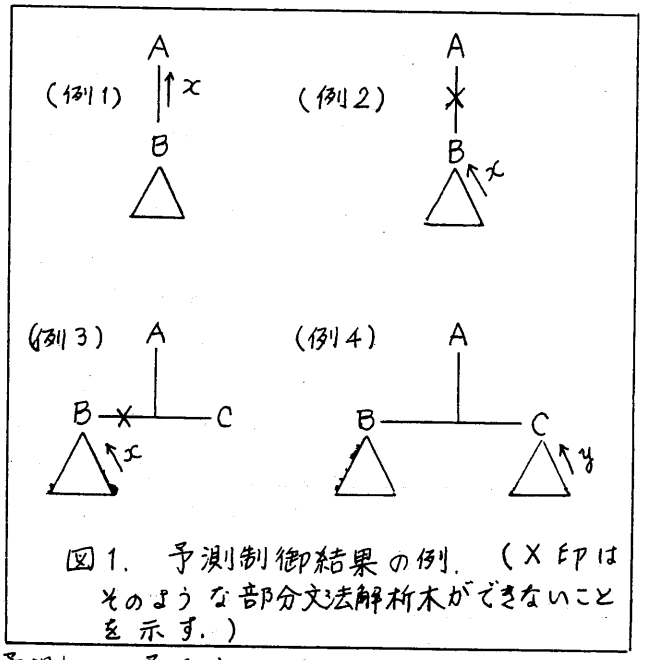


図1. 予測制御結果の例．(X印はそのような部分文法解析木ができないことを示す．)

(iii) memo-advice : (ma <S式>)，パーザに対する忠告を S 式としてメモしておく．メモされた S 式は，予測カテゴリが満足された時点で再び評価される．関数値は常に NIL．

(例4) A → B + C [(ma (NOT (qm 'y)))]，ノード B が成立すれば，無条件に C を予測カテゴリとし，予測文法規則を A → B + C にする．パージングが進み C が成立した時，ノード C にメッセージ y があるときに限り，A が成立する．

これらにより，(R1) はつぎのように書ける．辞書に，「を」,「に」,「へ」… ← 格助詞，「が」,「は」← 格助詞 [(～格)] ([ ] 内は，メッセージ～格を，格助詞というノードに送ることを意味．)，として，

格助詞 → 格助詞 + 格助詞 [(qm '～格)] ---- (R1)'''

§5. おわりに

予測制御関数を組合せることにより複雑な予測制御を行なうことができる．(R1) に関していえば，(1) の e_i は任意の S 式が書けるから，助詞の相互承接を記述した表の索引プログラムを書くことにより，CF 文法規則を増やさずに規則の精密化をはかることができる．パーザの制御用として，意味に近い情報も利用できよう．

§6. 謝辞．日頃有益な示唆をいただく渕一博推論機構研究室長に深謝する．

(1) Pratt, V. R.: "LINGOL--A Progress Report", Proc. IJCAI4, pp. 422-428 (1975).
(2) 田中穂積："構文情報を用いた自動わかち書きプログラム"，情報処理学会16回大会予稿 179，(1975)
(3) 田中穂積，横山晶一："活用語尾を処理するプログラム"，本大会予稿，(1976)
(4) 渕一博："定理証明としてみた Earley/Pratt のパージングアルゴリズム"，本大会予稿，(1976)
(5) Heidorn, G. E.: "Augmented Phrase Structure Grammar" in Schank and Nash-Webber ed.: "Theoretical Issues in Natural Language Processing", Cambridge, Mass., pp. 1-5, June, 1975.