自然言語処理 76－2
（1990．3．9）

句構造文法に対する効率的文解析手法

林達也* 宮俊司** 坂巻利哉** 吉田健一**

＊ 富士通研究所
＊＊ 富士通ソーシアルサイエンスラボラトリ

本論文では，一般の句構造文法に対する効率的な文解析手法について述べる．
この方法は，限定文脈依存文法（ＲＣＳＧ）を受け入れるＹＡＰＸＲ文解析システムを拡張する事によって得られる．従って本方式は，ＹＡＰＸＲと同様に横型トップダウン手法に基づいており，且つ又，ＹＡＰＸＲの特性を継承して効率の良い文解析を行うことが出来る．

日本語のような語順の自由な言語に対して，文脈自由文法ベースの記述形式は必ずしもしっくりしない面がある．そして，文節間の係り受け関係も含めて考えると，むしろ句構造文法の方が文脈自由文法ベースよりも素直に日本語の規則を記述出来る事が指摘されている．

一方ギャップ文法も，語順の自由な言語を扱えるが，性能に問題がある．

本方式はこれに対して，文法が与えられた時点で予め全ての位置集合が求められれば，句構造文法に対しても効率的な文解析を行う事が出来るので，此の種の言語に有効な手法であると考えられる．

An Efficient Sentence Analysis Method for General Phrase Structure Grammers

Tatsuya Hayashi * Syunji miya ** Toshiya Sakamaki ** Kenichi Yoshida**

* Fujitsu Laboratories LTD.
1015 Kamikodanaka Nakaharaku Kawasaki, 211, JAPAN

** Fujitsu Social Science Laboratory LTD.

This paper describes an efficient sentence analysis method for conventional phrase structure grammars.

The expressions based on context free grammars are not particularly suited for languages like Japanese where free word order is allowed. Moreover, it is pointed out elsewhere that when we consider dependency relationships among "Bunsetsus", phrase structure grammars are more suitable for expressing Japanese.

Our method is appropriate for such languages, since it provides the power of phrase structure grammars with improved efficiency, provided all the "position sets" could be calculated in advance.

1．まえがき

筆者等は先に，文脈自由文法を主体にしそれに解析効率に影響を及ぼさない範囲で文脈依存規則を導入したＲＣＳＧ（Restricted Context Sensitive Grammar）に対する文解析システムＹＡＰＸＲについて述べた[1)-5)]．

ＲＣＳＧは基本的には，文脈自由文法の枠組みから離れたものではない．そのため，日本語のように語順の自由な言語の場合は，必ずしもＲＣＳＧが適切な文法であるとは言えない．

ギャップ文法は語順の自由な言語を簡潔に記述できる特色を持っているが[6)-8)]，ｇａｐが０を含む任意の長さの部分導出列を代表出来るので，逆に性能上問題が生じる事が予想される．

一方日本語は，語順は自由だが，無条件に自由な訳ではなく，文節を単位として係り受け関係が右向きでかつ非交叉でなければならないと言う条件が存在する．そして，この係り受け関係に対応する非終端記号を導入すると，日本語は一般的な句構造文法（ＰＳＧ）を用いて素直に記述出来る事が示されている[9)]．

そこで，句構造文法に対する効率的な解析手法が課題となるが，これに対しては上述したＹＡＰＸＲを多少拡張する事で，横型トップダウン方式の句構造文法向けの実用的な解析システムを得る事が出来る．

本論文ではこれについて述べる事にする．以下では，2.で文法記述形式，3.で実現方法，4.で動作例について述べる．

2. 文法記述形式

ここで許される文法は，従来のＹＡＰＸＲで用いられている機能を全て包含した句構造文法である．基本形式は次に示す通りである．

左辺 → 右辺.

ここで，

左辺 ：左辺項…

左辺項：構文要素［(内部引数，…) ］

［(/外部引数，…/)］

右辺 ：ε又は右辺項

右辺項：構文要素［(内部引数，…) ］

［(/外部引数，…/)］

［スラッシュカテゴリ］

［｛文脈依存制約｝］

［｛基本制約｝］

但し，外部引数や文脈依存制約，スラッシュカテゴリは本論とは直接関係ないので省略する．（外部引数等については文献２） ４） ５）を参照されたい．

3. 実現方法

記述形式がＲＣＳＧに合致する場合は，従来の方法で実現できるため，ここでは，一般の句構造文法規則を対象にして実現方法を述べる．

ＲＣＳＧを対象とした場合は文法が与えられた時点で，規則右辺の左端要素に対応する核位置を全て容易に求める事が出来た．

しかし句構造文法を対象とする場合は，核位置を求めるのが容易でない場合がある．そこで，ここでは先ず核位置が決定したものとして述べる．尚，文法規則を

$P_1\ P_2 \cdots P_m \rightarrow {}^{n1}Q_1\ {}^{n2}Q_2 \cdots {}^{nn}Q_n$

$(P_i, Q_j \in V,\ nk$ ～位置id)

とする．簡単の為引数や制約は除外する．又，P_i, Q_j に対応する述語をそれぞれ p_i, q_j とする．

引数はＹＡＰＸＲでは以下の通りであった．

n ：解析パスの先頭要素

Ai ：解析スタック（入力）

（解析パスの残りの要素）

Ao, Aol ：解析スタック（出力，差分リスト）

Oi ：省略スタック（入力）

Oo, Ool ：省略スタック（出力，差分リスト）

Li ：引数スタック（入力）

Lo, Lol ：引数スタック（出力，差分リスト）

Ti ：解析木スタック（入力）

To, Tol ：解析木スタック（出力，差分リスト）

しかしここでは，本論に関係ない省略スタックは除外する．又，解析木は句構造文法の場合複雑になるので，解析木スタックには代わりに係り受け関係(W1,W2) を格納する事にする．

(W1,W2) は文節W1が文節W2に係る事を示す．

3.1 左端型

構文要素Q_1に対応するホーン節は，

```
q1(l1, Ai, [ [n2, l1 ¦ Ai] ¦ Ao] , Ao,
       Li, [ [Li] ¦ Lo] , Lo,
       Ti, [ [Ti] ¦ To] , To).
q1(l2, Ai, [ [n2, l2 ¦ Ai] ¦ Ao] , Ao,
       Li, [ [Li] ¦ Lo] , Lo,
       Ti, [ [Ti] ¦ To] , To ) .
         ・・・
         ・・・
q1(lk  Ai, [ [n2, lk ¦ Ai] ¦ Ao] , Ao,
       Li, [ [Li] ¦ Lo] , Lo,
       Ti, [ [Ti] ¦ To] , To ) .
```

となる．l1～lkはn1に対応する核位置である．

ljを左位置として持つ要素は一般に，非終端記号に限らないのが句構造文法の場合の特色である．

前述したように一般には，n1に対応する核位置を求める事が容易でない場合がある．但しその場合でも以下の様に2通りの方法のいずれかを用いることによって実現することができる．

先ず第1の方法は，解析パスの先頭要素が何であってもあたかも核位置であるかのように見なして，解析パスをn2に進めるという方法で，時間・空間的に非能率ではあるが，ともかく解析を正しく行う事が出来るのである．唯，その時の解析は完全なボトムアップになるので，ε規則は禁止される事になる．

これに対し第2の方法は，ε規則を許し且つ現実的な方法である．それは次の通りである．

先ず最初に，句構造規則の左辺は第1項のみを考える．すると，見かけ上ＹＡＰＸ́Ｒの場合と同様に左端要素に対応する核位置は全て求める事が出来る．これを仮核位置と呼ぶ事にしよう．

容易に分かるように，本来の核位置は同時に又仮核位置でもある．

こうしておいて，左端型のアクションを仮核位置に基づいて作成すれば良いのである．

3.2 中間型

Q_2に対応するホーン節は，

```
q2(n2, Ai, [ [n3 ¦ Ai] ¦ Ao] , Ao,
       Li, [ [Li] ¦ Lo] , Lo,
       Ti, [ [Ti] ¦ To] , To ) .
```

で，これは従来と全く同じである．

3.3 右端型

Q_m に対応するホーン節を次のように定める事がポイントである．

```
qn (nn, [ N ¦ Ai] , Ao, Ao1,
        Li, Lo, Lo1, Ti, To, To1) :-
    p1(N, Ai, A1, Ao1,
          Li, L1, Lo1, Ti, T1, To1),
    p200(A1, A2, L1, L2, T1, T2),
          ・・・
          ・・・
    pm 00(Am-1, Ao, Lm-1, Lo, Tm-1, To).
```

つまり，左辺の要素全体に対応する（と思われる）部分入力列が検出されたと見なして，記述された順序で連続して述語呼出を行う点が句構造文法の場合の特色である．

これを出発記号S_0からの導出と関連づけて見ると，右端型のアクションは，

$S_0 \doteqdot> \alpha P_1P_2\cdots P_m \tau \Rightarrow \alpha Q_1Q_2\cdots Q_n \tau$

に於いて，αの最後の要素の右位置からτの最初の要素の左位置に解析パスを進めることに他ならない．

p_200 ～ p_m 00は入力文に対応する述語と同様に，解析スタック（入力）は一般に複数の解析パスから成る（但し，リストの最後が変数である点が異なる）．従って，以下のホーン節を設けて，インターフェイスを合わせるものとする．

```
pj 00(  [ [ N ¦ Ai] ¦ Ao] , Ao1,
        [Li ¦ Lo] , Lo1,
        [Ti ¦ To] , To1) :-
    pj (N, Ai, Ao, Ao1,
          Li, Lo, Lo1,
          Ti, To, To1) ,
    !,
```

```
p』00(Ao, Aol, Lo, Lol, To, Tol).
p』00(Aol, Aol, Lol, Lol, Tol, Tol) :- !.
```

仮核位置方式の場合も右端型のアクションは同様に，左辺の要素に対応する述語を逐次的に呼び出せばよい．

但しこの場合の相違点は，核位置方式では最終的な解析パスが常に存在するが，仮核位置方式では逐次的な述語呼出の過程で，解析パスが消滅してしまう場合があると言う事である．

つまり，仮核位置方式は核位置方式よりは効率が良くないが，単純なボトムアップ方式に比べると効率的であり，両者の中間に位置する方法であるといえる．

4. 動作例

本節では簡単な日本語の文法を例にとり上げて，パーサの動きを説明しよう．

使用する文法例を図1 に示す．これは文献9)を参考にして少し手を加えたものである．

〔e 〕はピリオドを表す．〔p 〕は文節を示す終端記号で以下の引数を持つものとする．

W ：自立語＋付属語，

C1：自立語の品詞区分，

C2：付属語の品詞区分又は活用形

relは係り受け関係を示す非終端記号である．但し，relのみを左辺に持つ規則はなく，従ってそれから導出される終端記号は存在しない点を指摘しておく．

簡単の為，s や prseqには引数を持たせない事にする．規則1,4,5,6 の checkは構文的側面から係り受けの可否を調べる簡単な述語である．又，putは係り受け関係(W1,W2) を解析木スタックに格納する命令である．規則1 の基本制約は最後の文節が終止形でなければならない事を指定している．

この例では，左端要素 ^{9}[p]，^{12}[p]，^{16}[p] に対応する核位置には，b,8,10,14,11,15,17の７個が存在する．

この文法に基づくprologのプログラム（Quintus版）を付録に示す．

入力文を「tarouga jirouni renrakusuru kotowo yakusokusuru ．」としよう．

又，本システムへの入力述語は，

```
yopen(A1,L1,T1),
p0 (A1,A2,L1,L2,T1,T2, [tarouga,n,ga] ) ,
p0 (A2,A3,L2,L3,T2,T3, [jirouni,n,ni] ) ,
p0 (A3,A4,L3,L4,T3,T4, [renrakusru,vt,cn] ),
p0 (A4,A5,L4,L5,T4,T5, [kotowo,n,wo ] ),
p0 (A5,A6,L5,L6,T5,T6,
                    [yakusokusuru,vt,end ] ),
e0 (A6,A7,L6,L7,T6,T7, ['.',end,end ] ),
yclose(A7,L7,T7).
```

としよう．ここで，

n　　：名詞，

vt　　：他動詞，

cn　　：連体形，

end　：終止形，

yopen ：スタックの初期設定を行う述語，

yclose：結果（係り受け構造）を出力する述語，

である．入力述語のホーン節も付録に示しておく．

解析チャートの数はこの場合22個であるが，係り受け構造で見ると異なり数は図２に示すように５通りとなる．

(1)に対応する解析チャートの一つを示すと図３のようになる．余談であるが，この例は文脈処理迄行わなければ曖昧性が解消しない．

本方式はＬＲ手法をベースにしているので，従来のＹＡＰＸＲと同様に，解析チャートはボトムアップに作成される．

図３の場合，先ず規則６により（kotowo, yakusokusuru）に対応する部分チャートが生成され，述語p,rel,p が呼び出される．先頭の pはやはり規則６により右端型アクションを行って(renrakusuru, kotowo) に対応する部分チャートを生成し，再びp,rel,p を呼び出す．以下同様の手順を経て図のような解析チャートが生成される訳である．

ここで，終端要素^{17}[p] が左端要素^{16}[p] の核位置になっている所が従来のＲＣＳＧとは異なる点である．

5. あとがき

本論文では，横型トップダウン文解析システムＹ

ＡＰＸＲを拡張して一般の句構造文法に対する効率的な文解析システムが得られる事を示した．

日本語のように語順が自由な言語の場合，文脈自由文法の枠組みで記述する事は必ずしもしっくりしない所がある．ギャップ文法では，記述力は一応あるとしても，能率の良い解析が期待出来ない．

日本語は係り受け関係も含めて考えると，句構造文法の方が素直に表現出来るので，本方式の有効性が期待出来る．

つまり，ここに述べた方法は，従来のＹＡＰＸＲと同様に，

(1) 左端要素に対応する核位置あるいは仮核位置を全て，文法が与えられた時点で求める事が出来るので，実行時のトップダウン予測が不要である．

(2) prolog 処理系の特性を生かして，第１引数をアトムにしてダブルハッシュを可能にしている．

(3) 実行時にはバックトラックを生じない．

等により効率の良い解析システムになっているからである．本文では基本手法とその有効性の片鱗を簡単な例で示したに過ぎないので，機械翻訳や自然言語インターフェイス等の実用規模の日本語への適用が今後の課題と考えている．

参考文献

1)林達也：論理型言語による構文解析法ＹＡＰについて，情報処理学会論文誌， Vol. 29, No. 9, pp. 835-842(1988) .

2)林達也：拡張ＣＦＧとその構文解析法ＹＡＰＸについて，情報処理学会論文誌，Vol. 29, No. 5, pp.480-487 (1988) .

3)林達也：ＹＡＰＸの効率的実現法，情報処理学会論文誌，Vol. 30, No.10, pp. 1354-1356(1989).

4)林達也：横型トップダウン文解析システムの実現と評価，情報処理学会，自然言語処理研究会 Vol. 74-9, pp. 65-72(1989) .

5)林達也：文解析システムYAPXR の実現と評価，情報処理学会論文誌（投稿中）(1989).

6) Dahl,V. & Abramson,H.：On Gapping Grammars, Proc. 2nd International Conference on Logic Programming, pp. 77-88(1984).

7) Dahl,V.：More on Gapping Grammars, Proc. FGCS, PP. 669-677(1984) .

8) Popowich,F.：Unrestricted Gapping Grammars, Proc. IJCAI 85, pp. 765-768(1985) .

9) Sugimura,R.：Logical Dependency Grammar and Its Constraint Analysis , ICOT technical memorandum TM-0679, pp. 10 (1989).

```
0.   S0                          -->   [b] S [*] ┤
1.   S                           -->   [1] prseq        [2] [p ]  (w,c1,c2) {c2 = end}
                                        [3] [e ] .
2.   prseq                       -->   [4] [p ] (w,c1,c2)      [5] rel .
3.   prseq                       -->   [6] [p ] (w,c1,c2)      [7] rel        [8] prseq .
4.   [p ] (w1,c11,c12)  rel  [p ] (w2,c21,c22) -->
                                        [9] [p ] (w1,c11,c12)  [10] prseq
                                       [11] [p ] (w2,c21,c22) {chech(c11,c12,c21),put(w1,w2) } .
5.   [p ] (w1,c11,c12)  rel  [p ] (w2,c21,c22) -->
                                       [12] [p ] (w1,c11,c12)  [13] rel   [14] prseq
                                       [15] [p ] (w2,c21,c22) {chech(c11,c12,c21),put(w1,w2) } .
6.   [p ] (w1,c11,c12)  rel  [p ] (w2,c21,c22) -->
                                       [16] [p ] (w1,c11,c12)
                                       [17] [p ] (w2,c21,c22) {chech(c11,c12,c21),put(w1,w2) } .
```

図1　句構造文法例

```
(1)  (tarouga, yakusokusuru)   (jirouni, yakusokusuru)  (jirouni, renrakusuru)
     (renrakusuru, kotowo)     (kotowo, yakusokusuru)
(2)  (tarouga, yakusokusuru)   (jirouni, yakusokusuru)
     (renrakusuru, kotowo)     (kotowo, yakusokusuru)
(3)  (tarouga, yakusokusuru)   (jirouni, renrakusuru)
     (renrakusuru, kotowo)     (kotowo, yakusokusuru)
(4)  (tarouga, renrakusuru)    (jirouni, renrakusuru)
     (renrakusuru, kotowo)     (kotowo, yakusokusuru)
(5)  (tarouga, yakusokusuru)   (tarouga, renrakusuru)  (jirouni, renrakusuru)
     (renrakusuru, kotowo)     (kotowo, yakusokusuru)
```

図2　例文の解析結果

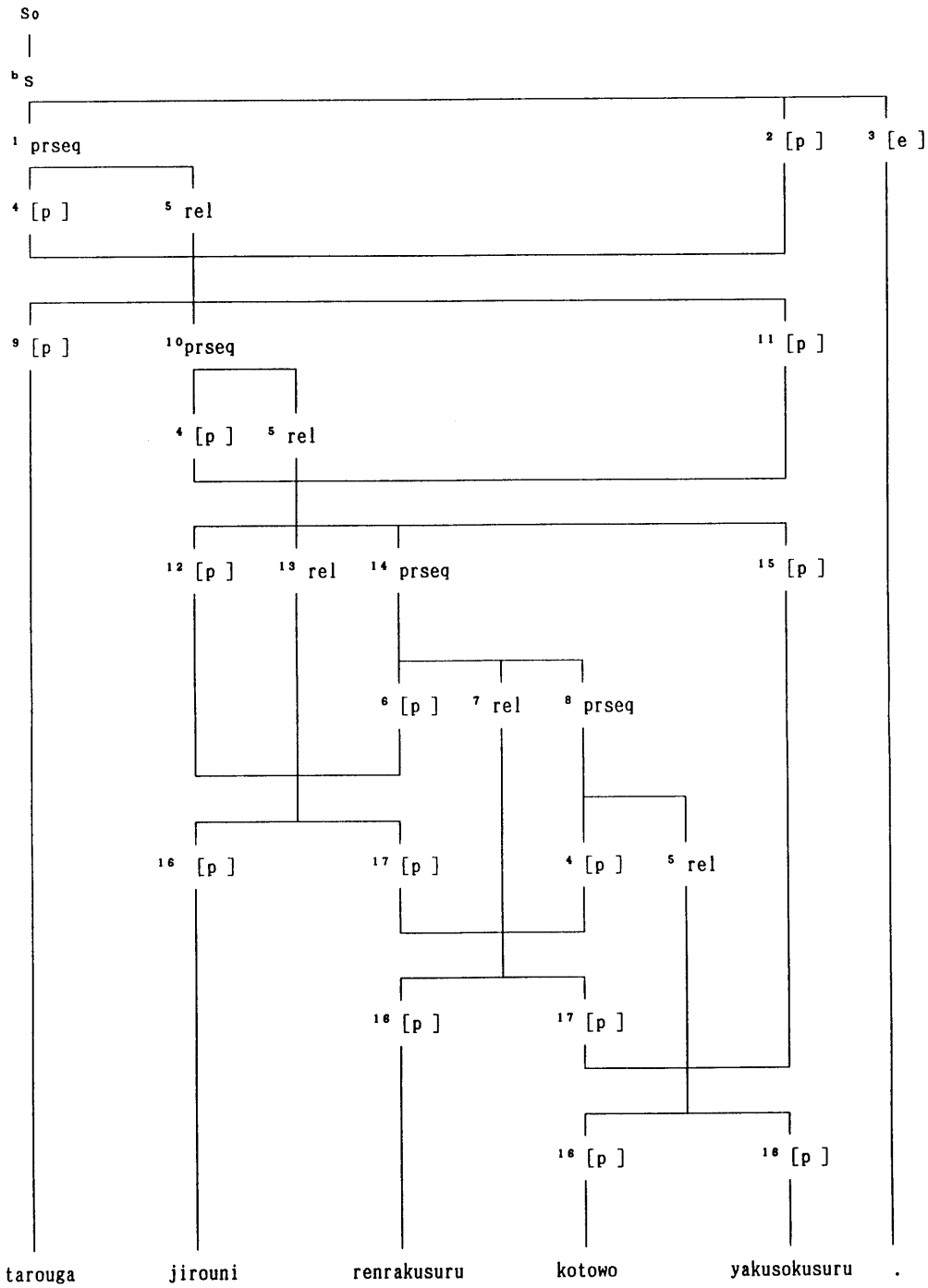

図3　例文の解析チャート

```
yapxr_2(Outfile):-
    tell(Outfile),
    yopen(Ai1,Li1,Ti1),!,
    p0(Ai1,Ai2,Li1,Li2,Ti1,Ti2,[tarouga,n,ga]),!,
    p0(Ai2,Ai3,Li2,Li3,Ti2,Ti3,[jirouni,n,ni]),!,
    p0(Ai3,Ai4,Li3,Li4,Ti3,Ti4,[renrakusuru,vt,cn]),!,
    p0(Ai4,Ai5,Li4,Li5,Ti4,Ti5,[kotowo,n,wo]),!,
    p0(Ai5,Ai6,Li5,Li6,Ti5,Ti6,[yakusokusuru,vt,end]),!,
    e0(Ai6,Ai7,Li6,Li7,Ti6,Ti7,['.',end,end]),!,
    yclose(Ai7,Li7,Ti7),
    close(Outfile).


yopen([[b]],[[]],[[]]):-!.

p0([[N|Ah]|At],Ao,[Lh|Lt],Lo,[Th|Tt],To,[W,C1,C2]):- !,
    p(N,Ah,Ao,Ao1,[W,C1,C2|Lh],Lo,Lo1,Th,To,To1),
    p0(At,Ao1,Lt,Lo1,Tt,To1,[W,C1,C2]).
p0([],[],_,[],_,[],_):- !.
e0([[N|Ah]|At],Ao,[Lh|Lt],Lo,[Th|Tt],To,[W,C1,C2]):- !,
    e(N,Ah,Ao,Ao1,[W,C1,C2|Lh],Lo,Lo1,Th,To,To1),
    e0(At,Ao1,Lt,Lo1,Tt,To1,[W,C1,C2]).
e0([],[],_,[],_,[],_):- !.

s(b,Ai,[[e|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
s(_,_,Ai,Ai,_,Li,Li,_,Ti,Ti):- !.

prseq(b,Ai,[[2,b|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
prseq(8,[N|Ai],Ao,Ao1,Li,Lo,Lo1,Ti,To,To1):- !,
    prseq(N,Ai,Ao,Ao1,Li,Lo,Lo1,Ti,To,To1).
prseq(10,Ai,[[11|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
prseq(14,Ai,[[15|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
prseq(_,_,Ai,Ai,_,Li,Li,_,Ti,Ti):- !.

rel(5,[N|Ai],Ao,Ao1,Li,Lo,Lo1,Ti,To,To1):- !,
    prseq(N,Ai,Ao,Ao1,Li,Lo,Lo1,Ti,To,To1).
rel(7,Ai,[[8|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
rel(13,Ai,[[14|Ai]|Ao],Ao,Li,[Li|Lo],Lo,Ti,[Ti|To],To):- !.
rel(_,_,Ai,Ai,_,Li,Li,_,Ti,Ti):- !.

e(3,[N|Ai],Ao,Ao1,[W,C1,C2|Li],Lo,Lo1,Ti,To,To1):- !,
    s(N,Ai,Ao,Ao1,Li,Lo,Lo1,Ti,To,To1).
e(_,_,Ai,Ai,_,Li,Li,_,Ti,Ti):- !.

p(2,Ai,Ao,Ao1,[W,C1,C2|Li],Lo,Lo1,Ti,To,To1):-
    ((C2 = end,!,
      Ao = [[3|Ai]|Ao1],
      Lo = [Li|Lo1],
      To = [Ti|To1]);
     (Ao = Ao1, Lo = Lo1, To = To1)).
p(b,Ai,[[5,b|Ai],[7,b|Ai],[10,b|Ai],[13,b|Ai],[17,b|Ai]|Ao],Ao,
    [W,C1,C2|Li],
    [Li,Li,[W,C1,C2|Li],[W,C1,C2|Li],[W,C1,C2|Li]|Lo],Lo,
    Ti,[Ti,Ti,Ti,Ti,Ti|To],To):- !.
p(8,Ai,[[5,8|Ai],[7,8|Ai],[10,8|Ai],[13,8|Ai],[17,8|Ai]|Ao],Ao,
    [W,C1,C2|Li],
    [Li,Li,[W,C1,C2|Li],[W,C1,C2|Li],[W,C1,C2|Li]|Lo],Lo,
    Ti,[Ti,Ti,Ti,Ti,Ti|To],To):- !.
p(10,Ai,[[5,10|Ai],[7,10|Ai],[10,10|Ai],[13,10|Ai],[17,10|Ai]|Ao],Ao,
    [W,C1,C2|Li],
    [Li,Li,[W,C1,C2|Li],[W,C1,C2|Li],[W,C1,C2|Li]|Lo],Lo,
    Ti,[Ti,Ti,Ti,Ti,Ti|To],To):- !.
p(14,Ai,[[5,14|Ai],[7,14|Ai],[10,14|Ai],[13,14|Ai],[17,14|Ai]|Ao],Ao,
    [W,C1,C2|Li],
    [Li,Li,[W,C1,C2|Li],[W,C1,C2|Li],[W,C1,C2|Li]|Lo],Lo,
    Ti,[Ti,Ti,Ti,Ti,Ti|To],To):- !.
p(11,[N|Ai],Ao,Ao2,
     [W2,C21,C22,W1,C11,C12|Li],Lo,Lo2,
     Ti,To,To2):-
    ((check(C11,C12,C21),!,
      Ao = [[10,11,N|Ai],[13,11,N|Ai],[17,11,N|Ai]|Ao4],
      Lo = [[W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li]|Lo4],
      To = [Ti,Ti,Ti|To4],
      p(N,Ai,Ao1,Ao2,[W1,C11,C12|Li],Lo1,Lo2,
        [(W1,W2)|T1],To1,To2),
      rel00(Ao1,Ao3,Lo1,Lo3,To1,To3,[]),
      p00(Ao3,Ao4,Lo3,Lo4,To3,To4,[W2,C21,C22]));
     (Ao = Ao2, Lo = Lo2, To = To2)).

p(15,[N|Ai],Ao,Ao2,
     [W2,C21,C22,W1,C11,C12|Li],Lo,Lo2,
     Ti,To,To2):-
    ((check(C11,C12,C21),!,
      Ao = [[10,15,N|Ai],[13,15,N|Ai],[17,15,N|Ai]|Ao4],
      Lo = [[W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li]|Lo4],
      To = [Ti,Ti,Ti|To4],
      p(N,Ai,Ao1,Ao2,[W1,C11,C12|Li],Lo1,Lo2,
        [(W1,W2)|T1],To1,To2),
      rel00(Ao1,Ao3,Lo1,Lo3,To1,To3,[]),
      p00(Ao3,Ao4,Lo3,Lo4,To3,To4,[W2,C21,C22]));
     (Ao = Ao2, Lo = Lo2, To = To2)).

p(17,[N|Ai],Ao,Ao2,
     [W2,C21,C22,W1,C11,C12|Li],Lo,Lo2,
     Ti,To,To2):-
    ((check(C11,C12,C21),!,
      Ao = [[10,17,N|Ai],[13,17,N|Ai],[17,17,N|Ai]|Ao4],
      Lo = [[W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li],
            [W2,C21,C22,W1,C11,C12|Li]|Lo4],
      To = [Ti,Ti,Ti|To4],
      p(N,Ai,Ao1,Ao2,[W1,C11,C12|Li],Lo1,Lo2,
        [(W1,W2)|T1],To1,To2),
      rel00(Ao1,Ao3,Lo1,Lo3,To1,To3,[]),
      p00(Ao3,Ao4,Lo3,Lo4,To3,To4,[W2,C21,C22]));
     (Ao = Ao2, Lo = Lo2, To = To2)).
p(_,_,Ai,Ai,_,Li,Li,_,Ti,Ti):- !.

rel00(Ai,Ai,Li,Li,Ti,Ti,):-
    var(Ai),!.
rel00([[N|Ah]|At],Ao,[Lh|Lt],Lo,
      [Th|Tt],To,[]):- !,
    rel(N,Ah,Ao,Ao1,Lh,Lo,Lo1,Th,To,To1),
    !,
    rel00(At,Ao1,Lt,Lo1,Tt,To1,[]).
p00(Ai,Ai,Li,Li,Ti,Ti,):-
    var(Ai),!.
p00([[N|Ah]|At],Ao,[Lh|Lt],Lo,
      [Th|Tt],To,[W1,C11,C12]):- !,
    p(N,Ah,Ao,Ao1,[W1,C11,C12|Lh],Lo,Lo1,Th,To,To1),
    !,
    p00(At,Ao1,Lt,Lo1,Tt,To1,[W1,C11,C12]).


check(n,ga,vt):- !.
check(n,ni,vt):- !.
check(n,wo,vt):- !.
check(vt,cn,n):- !.


yclose([],[],[]):- !.
yclose([Acar|Acdr],[Lcar|Lcdr],[Tcar|Tcdr]):- !,
    print('Ai='),print(Acar),nl,
    print('Li='),print(Lcar),nl,
    print('Ti='),print(Tcar),nl,
    yclose(Acdr,Lcdr,Tcdr).
```