

HIPSを用いた新聞記事割付け方式

富田 雅己 溝口理一郎 角所 収

(大阪大学 産業科学研究所)

1. まえがき

知識工学の分野において、プロダクションシステム⁽¹⁾⁽²⁾⁽³⁾ (以下、PSと略す)は有望な知識表現法の一つと考えられており、DENDRAL⁽⁴⁾やMYCIN⁽⁵⁾などの多くの実用する知識ベースシステムにおいて用いられている。

PSの基本動作はあるWorking Memory (WM) に対して、その条件部がマッチするルールを探し、起動可能なルールが複数個存在する場合には、ある種の戦略に従って、ルール競合の解消 (conflict resolution) を行ない、一つのルールを選択し、実行するという繰り返しである。

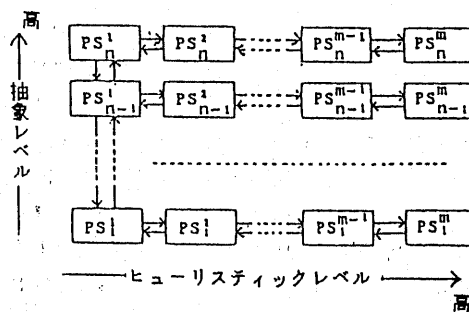
このような基本動作を行うPSは、Modularity (ルールの追加、削除、変更が容易)、Readability (ルールが一片の完結した知識であるため、その意味の理解が容易)、Self-Explanatory (結論までの処理の全過程の理解が容易) という3つの特徴を持つので、全体の制御の構造が不明確な分野、或いはそれを事前に決定することが極めて困難な分野に適している。しかし、PSには効率の悪さ、制御構造の一樣性に起因するプログラミングの困難さ等の欠点がある。そこで、タグ・マーカー等の特殊記号の導入に代表されるad-hocな改良がなされてきたが、PSの優れた特性が犠牲にされた。我々は、PSの3つの特徴を保存しつつ、さらにPSの能力を向上させるために、基本PSを2方向の階層に配列した階層的プロダクションシステム (Hierarchical Production System) を開発し、ブロック問題などでは良好な結果が得られることを報告した。⁽⁶⁾ HIPSは、問題固有の知識を抽象度別に階層的に組み込むことにより、人間の階層的思考法により近い処理をすることができ、ルール競合解消のためのヒューリスティックスを直接ユーザが組み込むことにより、従来のPSよりも効率の良い処理をすることが可能となった。本稿では、このような特徴をもつHIPSが現実的レベルの問題解決に対して、有力な手段となり得ることを確認するために、人間が階層的思考法を行っていると考えられる新聞記事割付け問題を取り上げる。以下にHIPS設計の基本思想とシステムの動作の概略、及びHIPSを用いた2種類の新聞記事割付け方式について述べる。

2. 階層的プロダクションシステムHIPSの概略

2.1 HIPSの階層性と特徴

PSを用いた知識表現に基づく問題解決システムを設計する際には、問題領域をほとんど独立した状態に分解し、問

題固有の知識をその使われ方とは無関係にシステムに組み込まなければならない。HIPSでは、Hearsay-II⁽⁷⁾⁽⁸⁾やABSTRIPS⁽⁶⁾に用いられているように問題領域の知識が抽象度の異なる階層的な知識に分解して表現される。また、ルール競合解消のためのヒューリスティックな知識についても階層性を導入し、ユーザが定義したヒューリスティックスを他の知識同様、ルール形式で組み込むことを可能にした。このような2種類の階層的知識を反映させるためにHIPSは図1のような基本PSを2方向の階層構造に配列した構造を持つ。

図1 階層的プロダクションシステムHIPS_n^m

また、HIPSは既存のPSを自然な方向に拡張するように設計されており、ルール競合解消のためのad-hocな手法は何も使われていない。さらに、縦の抽象レベルの設定により人間の階層的思考法と同様に、抽象度の高い順に (大まかなプランニングから具体的なプランニング) 問題解決が行なわれていくこともHIPSの特徴の一つである。

2.2 HIPSの動作の概略

最初ユーザが、問題記述として初期状態 (WMとなる) と最終 (ゴール) 状態、及び各階層に対応したルールすなわち階層的な問題固有の知識とヒューリスティックな知識のリスト (PM) を与えることによりHIPSは作動する。HIPSの縦横のレベルはユーザが作成するこのPM内のルールの階層性の数に従って実行時に自動的に生成される。図1のような階層構造を持つHIPSは最初最も高い抽象レベルのルールを使って基本PSであるPS_n¹を動かして抽象度の高い解を得る。しかし、その途中の各繰り返しサイクルにおいて競合が生じれば、ヒューリスティックレベルで一つ上のPS_n²

を起動し、ルールの競合の解消を行う。その際にヒューリスティック用のルールが競合を起せば、さらに上位の PS_n^3 、 PS_n^4 、----、 PS_n^m と競合が完全に解消されるまで或いはヒューリスティックレベルのルールがなくなるまで同様の操作が繰り返され、その後もとの PS_n^1 での繰り返しサイクルに戻り、処理を再開する。そして最終的に解の系列が求まればそれをもって一つ抽象レベルの低い PS_{n-1}^1 に降り、そのレベルでのルールを使って PS_n^1 で得られた解系列をサブゴール系列とみなし各サブゴール間を埋めて行く。同様に横のヒューリスティックレベルのPSをも必要な際には、それを起動しながら抽象レベルを $PS_{n-2}^1, PS_{n-3}^1, \dots, PS_1^1$ へと徐々に降りていき、プランをより詳細なものへと具体化していく。

以上がHIPSの全体の動作の概略であるが、この動作を制御する機構として、WMとルールとのマッチングの際のバックトラック、ある繰り返しサイクルにおいて処理を進めることができなくなった際のバックトラック、及びある抽象レベルで得た解の中のサブゴールが適当でないことが下位のレベルに降りて初めてわかった場合、上位レベルにバックトラックしてそのサブゴールを修正し直すレベル間のバックトラックという3種類のバックトラック機構がある。さらにHIPSでは、ルールのLHSにはマッチング要素のほか*で始まる名前を持つLISP関数を書くことができ、RHSには任意のLISP関数を書くことができるように拡張されている。このためHIPSはLISPを含んだ複合的プログラミング言語となっておりルール形式では表現が困難な部分をLISP関数で表現することでルール表現する場合よりも自然で容易なプログラミングが可能となっている。

3. 新聞記事の割付け方式

3.1 基本的方針

ここでは新聞記事割付け問題を解くにあたり考慮した抽象レベルの設定、処理手順などについて2種類の割付け方式を対照させながら簡単に述べる。

1) 抽象レベルを、割付け方式1では3つに、割付け方式2では5つに設定した。方式1における3つの抽象レベルは、抽象度の高い順にそれぞれ、レベル1：与えられた記事の配置可能性の判定、レベル2：重要度による記事の大きな割付け、レベル3：見出し、写真の大きさ、形状などの条件を考慮した詳細な割付けを行なう。方式2では各抽象レベルごとに適当な大きさのブロック*に分け、レベル1～レベル4では、記事の使用ブロックとその使用面積を決める。レベル5では、レベル4での記事の使用ブロックリストを参照し、見出し、写真の配置も考慮した行単位

での詳細な割付けを行なう。下位レベルに行くほど紙面を小さなブロック単位に分割し、人間の階層的思考法に対応した処理が行なわれる。紙面の大きさは、どちらの方式とも横100行、縦15段とした。表1に方式2の各抽象レベルにおけるブロックの数と大きさを示す。

	抽象レベル1	抽象レベル2	抽象レベル3	抽象レベル4
ブロックの数	4	16	64	256
ブロックの大きさ (行数×段数) 四捨五入	56×8	28×4	14×2	7×1
	448	112	28	7

表1 抽象レベルにおけるブロック

2) 記事にはあらかじめ重要度**を与え、方式1、2とも重要度の順に記事を割付ける。広告など配置すべき位置の決められた記事は先に割付けるが、割付け場所の決められていない記事の割付けは方式1では、step1：記事の最も右上位置（原点と呼ぶ）となる行の決定、step2：記事の形状の決定、step3：記事内の写真の位置など詳細な条件を考慮した記事の割付け、という3段階、方式2では、step1：記事の原点となるブロックの決定、step2：各抽象レベルでの使用ブロックの決定という2段階で行なわれる点が異なる。方式1の記事の形状の決定は、あらかじめWM内に用意された25種類の形状を指定したリストとルールのLHSとのマッチングによって行なわれる。方式2では使用ブロックの決定の際に発火するルールによってブロック単位の大まかな形状が決める。 (レベル5では、行単位の最終的な形状が決定)

3) 方式1では、抽象レベル3のみ、ヒューリスティックレベルを設け、記事の形状と残り領域の形状を考慮した評価関数***を定義した。方式2では、抽象レベル1～4においてヒューリスティックレベルを設け、4つの評価基準によりルールの競合解消を行なった。

3.2 割付け方式1

3.2.1 状態記述 (WM) とルール (PM)

WMの要素は、次の3種類がある。

* 紙面の最も右上のブロックを(1 1)とし、ブロック(X Y)は右からX番目で上からY番目の位置のブロックを示す。

** 重要度はP0～P3の4段階とした。重要度P0の記事は、あらかじめ配置される位置の決められている広告などで、各レベルとも最初に割付けられる。

*** 文献[10]参照。

①記事の形状を指定したリスト

(FORM 2 2 ((E F A C) (1 > 2)))

この例では、図2のように1つの記事を2つの長方形に分割し、X座標が $E < F < A < C$ の順、すなわち①の長方形が②の長方形よりも左にあり①が②より大きいことを示している。

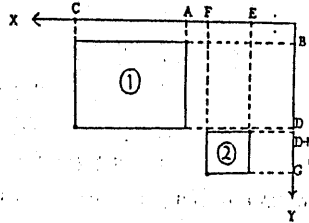


図2 記事の形状

②記事のデータ

(EC 260, P2 206, (L 3) (HEAD (W 10) (L 3) HD1) (PHOTO (W 12) (L 2) PT1))

これは、記事名がECで全体の大きさが260行分、重要度P2記事部分の大きさが206行、横10行、縦3段分の大きさのHD1という名の見出しと、横12行、縦2段分の大きさのPT1という写真があることを示し、(L 3)はこの記事では最小限縦に3段分のスペースが必要であることを示す。

③記事の割付け結果

(LAYOUT AV (1 <= X <= 100.) (12. <= Y <= 15))

これは、図3に示すようにAVという名の記事が第1行～100行、第12段～15段の領域に割付けられていることを示す。

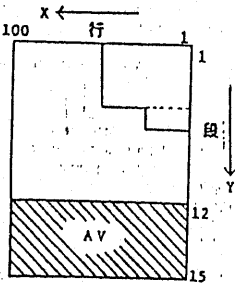


図3 記事の割付け

次に、図4にルールのリスト (PM) を示す。ここでA1 B1～B5, C1～C7は各々抽象レベル1, 抽象レベル2, 抽

象レベル3用のルールである。CR1, CR2は抽象レベル3の競合解消のためのヒューリスティックレベルのルールである。ルールLHSの第1要素が*で始まらないリストはWMとマッチングを取るリストであり、第1要素が*で始まるリストはマッチングを取った後、そのマッチング要素が適当であるかどうかを判断する関数である。RHSはLHSでマッチした記事に関して、見出しの割付け、形状の決定、写

```
((((AI ((POSSIBILITY-LAYOUT 100, 15, WM) ---)
      ((SETV WM NIL) (DEPOSIT (LAYOUT OK!)) (SAY))))))
((BI ((=NAME =SIZE P0 %COND) (%COND %COND) ---)
      ((ZDEL =NAME) (LAYOUT =NAME %COND) (SAY)))
(B2 ((=NAME =SIZE P1 =ARSIZE %COND) ---)
      ((ZDEL =NAME) (%ZDISTRIBUTE 1 =NAME) (SAY)))
(B3 ((=NAME =SIZE P2 =ARSIZE %COND) ---)
      ((ZDEL =NAME) (%ZDISTRIBUTE 2 =NAME) (SAY)))
(B4 ((=NAME =SIZE P3 =ARSIZE %COND) ---)
      ((ZDEL =NAME) (%ZDISTRIBUTE 3 =NAME) (SAY)))
(B5 ((NOTARTICLE WM) ---) ((DEPOSIT (LAYOUT OK!)) (SAY))))
((CI ((=NAME =SIZE P0 %COND) (%COND %COND) ---)
      ((ZDEL =NAME) (LAYOUT =NAME %COND) (SAY)))
(C2 ((=NAME =SIZE P1 =ARSIZE %COND) (ORG =ORG)
      (%POINTSELECT 'P1 =ORG) (%HEADCHECK =ORG %COND) ---)
      ((ZDEL 'ORG) (%DEPOSIT (=NAME ORIGIN =ORG)
        (%ASSIGNHEAD =NAME =ORG %COND) (SAY))))
(C3 ((=NAME =SIZE P2 =ARSIZE %COND) (ORG =ORG)
      (%POINTSELECT 'P2 =ORG) (%HEADCHECK =ORG %COND) ---)
      ((ZDEL 'ORG) (%DEPOSIT (=NAME ORIGIN =ORG)
        (%ASSIGNHEAD =NAME =ORG %COND) (SAY))))
(C4 ((=NAME =SIZE P3 =ARSIZE %COND) (ORG =ORG)
      (%POINTSELECT 'P3 =ORG) (%HEADCHECK =ORG %COND) ---)
      ((ZDEL 'ORG) (%DEPOSIT (=NAME ORIGIN =ORG)
        (%ASSIGNHEAD =NAME =ORG %COND) (SAY))))
(C5 ((=NAME =SIZE =P =ARSIZE %COND) (=NAME ORIGIN =ORG)
      (%FORM =HD =D !FORM) (=NAME HEAD =HEAD)
      (%FORMCHECK =SIZE !ORG %COND !FORM =HEAD =NO) ---)
      ((!FORM =NAME =HEAD !FORM !ORG)
        (%DELETE (=NAME ORIGIN !ORG) (=NAME HEAD =HEAD)) (SAY)))
(C6 ((=NAME =SIZE =P =ARSIZE %COND) (FORML =NAME %LAYOUT)
      (%ELSECONDCHECK %COND %LAYOUT) ---)
      ((ZDEL =NAME) (LAYOUTALL =NAME %COND %LAYOUT =P) (SAY)))
(C7 ((NOTARTICLE WM) ---) ((DEPOSIT (LAYOUT OK!)) (SAY))))
(CR1 ((JUDGEORIGIN) (%ORIGINDIFFERENCE)) ---) ((REPLACEMIN) (STOP)))
(CR2 ((JUDGEFORM) (%FORMDIFFERENCE)) ---) ((REPLACEMIN) (STOP))))
```

図4 ルール集合 (PM) (割付け方式1)

```
((FORM 1 1 ((P1 (A B)) (P3 (C D))))
(FORM 2 2 ((E F A C) (1 > 2)))
(FORM 3 2 ((E F A C) (1 = 2)))
(FORM 4 2 ((E F A C) (1 < 2)))
(FORM 5 2 ((A = F) (E A C) (1 > 2)))
(FORM 6 2 ((A = F) (E A C) (1 = 2)))
(FORM 7 2 ((A = F) (E A C) (1 < 2)))
(FORM 8. 2 ((E A F C) (1 > 2)))
(FORM 9. 2 ((E A F C) (1 = 2)))
(FORM 10. 2 ((E A F C) (1 < 2)))
(FORM 11. 2 ((A = E) (A F C)))
(FORM 12. 2 ((A E F C)))
(FORM 13. 2 ((C = F) (A E C)))
(FORM 14. 2 ((A = E) (A C F)))
(FORM 15. 2 ((E A C F)))
(FORM 16. 2 ((C = F) (E A C)))
(FORM 17. 2 ((A E C F) (1 > 2)))
(FORM 18. 2 ((A E C F) (1 = 2)))
(FORM 19. 2 ((A E C F) (1 < 2)))
(FORM 20. 2 ((C = E) (A C F) (1 > 2)))
(FORM 21. 2 ((C = E) (A C F) (1 = 2)))
(FORM 22. 2 ((C = E) (A C F) (1 < 2)))
(FORM 23. 2 ((A C E F) (1 > 2)))
(FORM 24. 2 ((A C E F) (1 = 2)))
(FORM 25. 2 ((A C E F) (1 < 2)))
(AC1 440, P1 318, (W 50.)
      (UPPERHEAD (W 50.) (L 1) UH1)
      (HEAD (W 8.) (L 4) HD1)
      (PHOTO (W 20.) (L 2) PT1))
(AD1 400, P0 ((12.) (100, 15.)))
(AD2 50, P0 ((76, 10.) (100, 11.)))
(AD3 40, P0 ((91, 1) (100, 4)))
(EC1 260, P2 206, (L 3) (HEAD (W 10.) (L 3) HD1)
      (PHOTO (W 12.) (L 2) PT1))
(FD1 180, P2 150, (L 2) (HEAD (W 15.) (L 2) HD1))
(EC2 85, P3 85, (W 1) (L 1))
(AC2 45, P3 45, (W 1) (L 1)))
```

図5 初期状態 (WM) (割付け方式1)

真の割付けを行なう関数とWMの内容を変更する関数から成る。

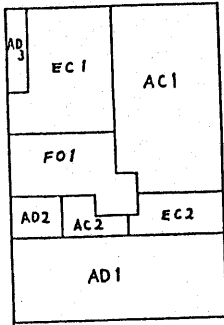


図6 記事の割付け結果
(割付け方式1)

3.2.2. 動作と結果の考察

記事データとして図5、PMとして図4を与え、HIPSを動作させた結果を図6に示す。記事の割付けは、HIPSの動作と同様、各抽象レベルごとに行なわれ、抽象度の高い順に処理が進む。各レベルの割付けが終れば(LAYOUT OK!)というリストがWMにつけ加えられ、レベル3の実行が終れば割付けが完了したことになる。この実行例を見て判断すれば、良い割付けを行なったと言えるが、LHSの*つきLISP関数内部では、かなり複雑な記号処理が行なわれており、ルールを見ただけでは処理の内容を理解することはできず、HIPSの利点であるReadabilityが失われてしまった。

また、割付けを2段階の抽象レベルに分けて、人間の階層的思考法に対応した考え方で処理を行なったが、2つの抽象レベルでは、問題固有の知識の分解が不十分であり、人間の持つ知識の階層性をうまく反映しているとは思えない。

3.3 割付け方式2

3.3.1 状態記述(WM)とルール(PM)

WM中には、次の2種類の情報が記述される。

①記事のデータ

```
(DATA AA 430. P1 (UPPERHEAD (W 30) (L 1) UH1)
  (PHOTO (W 20) (L 2) PT1 LEFT))
(DATA FF 280. PD ((1 14.) (100. 15.)))
```

これはDATAというアトムで始まるリストで、上例のように記事名、大きさ、重要度、見出し写真の情報などが記述されている。また重要度PDの記事については配置すべき位置の情報が書かれている。

②記事の割付け結果

次のリストはレベル2における記事AAの割付け結果を示したものである。

```
(LEVEL2 (1. 2.) (FULL 112.) ((AA 112.)))
(LEVEL2 (2. 2.) 84. ((AA 84.)))
(ASSIGN LEVEL2 AA ((2. 2.)(1. 2.)(2. 1.)
  (1. 1.)))
```

最初のリストはレベル2のブロック(1.2.)がその面積112行分すべてが使用され、AAという名の記事が112行分使っていることを示す。3番目のリストは、レベル2において、AAという名の記事が、ブロック(2.2.)、(1.2.)、(2.1.)、(1.1.)を使用することを示す。記事の割付けが終るごとに①のリストがWMから除かれ、②のリストがWMにつけ加えられる。

図7に、ルール(PM)の一部を示す。PMは、実際の記事の割付けに関係したルールが99個、ルール競合解消用のルールが15個、合計114個用意されている。

```
((L1R0 ((DATA =NAME =SIZE PO $LAYOUT) (COND1 $LAYOUT))
  ---)((DELETE (DATA =NAME =SIZE PO $LAYOUT))
  (LAYOUTPO =NAME =SIZE LAYOUT 'LEVEL1) (SAY)))
((L1R1 ((DATA =NAME =SIZE PO $LAYOUT) (COND1 $LAYOUT))
  ---)((DELETE (DATA =NAME =SIZE PO $LAYOUT))
  (LAYOUTPO =NAME =SIZE LAYOUT 'LEVEL2) (STOP)))
((L2R2 ((DATA =NAME =SIZE =PR $COND) (-DATA =N =S PO $L) (ORGBLOCK =ORGB)
  (ORGBLOCKSELECT =ORGB 'LEVEL2))
  ---)((DEL 'ORGBLOCK) (ASSIGN2 =NAME =ORGB 'LEVEL2) (SAY)))
((L2R3 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (LEFTFAST =NAME =SIZE !ORGB 'LEVEL2))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL2) (SAY) (STOP)))
((L2R4 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (LEFTFAST-DOWNRIGHT =NAME =SIZE !ORGB 'LEVEL2))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL2) (SAY) (STOP)))
((L2R5 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (FROM-DOWN 2 =NAME =SIZE !ORGB 'LEVEL2))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL2) (SAY) (STOP)))
((L2R6 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (FROM-DOWNRIGHT 2 =NAME =SIZE !ORGB 'LEVEL2))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL2) (SAY) (STOP)))
((L2R10 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (RIGHT-DOWN =NAME =SIZE !ORGB 'LEVEL2))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL2) (SAY) (STOP)))
((L2END ((!NOTARTICLE WH) ---)((DEPOSIT (LAYOUT OK!)) (SAY) (STOP)))
((L2HR1 ((!NAMESORT0B) ---)((REPLACENAME) (STOP)))
((L2HR2 ((!NAMESORT1) ---)((REPLACENAME) (NEXTRULE)))
((L2HR3 ((!LEVELDIFFERENCE 'LEVEL2) ---)((REPLACERULE) (STOP)))
((L2HR4 ((!)) ---)((STOP)))
((L3R1 ((DATA =NAME =SIZE PO $LAYOUT) (COND1 $LAYOUT))
  ---)((DELETE (DATA =NAME =SIZE PO $LAYOUT))
  (LAYOUTPO =NAME =SIZE LAYOUT 'LEVEL3) (STOP)))
((L3R2 ((DATA =NAME =SIZE =PR $COND) (-DATA =N =S PO $L) (ORGBLOCK =ORGB)
  (ORGBLOCKSELECT =ORGB 'LEVEL3))
  ---)((DEL 'ORGBLOCK) (ASSIGN2 =NAME =ORGB 'LEVEL3) (SAY) (STOP)))
((L3R3 ((DATA =NAME =SIZE =PR $COND) (=NAME ORGBLOCK !ORGB)
  (LEFTFAST =NAME =SIZE !ORGB 'LEVEL3))
  ---)((ASSIGN4 =NAME =SIZE 'LEVEL3) (SAY) (STOP)))
((L3R20 ((DATA =NAME =SIZE =PR $COND) (=NAME ORIGIN !ORGB) (-DATA =N =S PO $L))
  ---)((DELETE (DATA =NAME =SIZE =PR $COND) (=NAME ORIGIN !ORGB)
  (LAYOUT) (=NAME) (SAY) (STOP)))
((L3END (-DATA =NAME =SIZE =PR $COND) ---)((DEPOSIT (LAYOUT OK!)) (SAY) (STOP)))
```

図7 ルールの集合(PM)の一部(割付け方式2)

各ルールの動作は次のように分類される。

- ①重要度P0の記事の割付けを行なう。(5個)
- ②レベル1における重要度と記事の大きさを考慮した大まかな割付けを行う。(14個)
- ③原点となるブロックまたは行を決定する。(4個)
- ④使用ブロックを決定する。(53個)
- ⑤レベル5での各段ごとの記事の割付けを行う。(15個)
- ⑥見出しまたは写真の割付けを行う。(2個)
- ⑦各レベルにおける記事の割付けが完了した時に発火し処理を終了する。(6個)

特に②, ③, ④における原点の位置決定, 使用ブロックの決定などはLHSのLISP関数によって行なわれる。

3.3.2 評価関数

方式2では, できるだけ人間が見て良い配置であると考えられる割付けを行なうルールが選択されるように, ルール競合解消のために次の4つの評価関数(基準)を用意した。

- ①上位の抽象レベルで求めた使用ブロックの領域内に含まれる下位の抽象レベルで求められた使用ブロック数。
- ②記事に見出しがある場合, 見出しが入らないような大きさを生成したルールはコンフリクトセットから取り除き, 見出しが入る場合は10点を与える。
- ③未割付け領域の形状の良さを考えた次式で表わされる評価値A。

$$A = -\sum_i \left(\sum_j \frac{1}{C_{ij}} \right)$$

ただし

$$C_{ij} = \sum P, \quad P = \begin{cases} 0 & (a=5) \\ \frac{a}{5} & (0 < a < 5) \\ 1 & (a=0) \end{cases}$$

i: 段の番号, j: 数える方向

(1: 上方向, 2下, 3左, 4右)

S: あるブロックの面積

a: 面積Sのブロックの使用面積

C_{ij} は割付ける記事を各段ごとに分け各段の最も右側及び左側のブロックから, 上下左右の方向に向かってカウントした未割付けブロックの数である。このカウントはその方向に未割付けブロックがなくなった時, あるいは, 今考えている記事が使用しているブロックに出会った場合に終了する。

④記事の形状がなるべく正方形に近いものが選択されるような次式の評価値B。

$$B = \begin{cases} \left(\frac{W_x}{W_A} + \frac{L_x}{L_A} + \frac{L_A}{W_A} \right) \times 10 & (W_A \geq L_A) \\ \left(\frac{W_x}{W_A} + \frac{L_x}{L_A} + \frac{W_A}{L_A} \right) \times 10 & (W_A < L_A) \end{cases}$$

ここで, 記事の横方向のブロック数の最大を W_A , 最小を W_x ; 縦方向のブロック数の最大を L_A , 最小を L_x とする。

以上の①~④の評価値の合計点が, コンフリクトセット内の1つのルールの得点であり, この得点の高い順にルールが並べ換えられる。

3.3.3 動作と結果の考察

WMの初期状態として図8を与え, PMとして図7のようなルールを与えて, HIPSを動作させた抽象レベル3の割付け結果と最終的な結果を図9に示す。

```
(DATA AA 430, P1 (UPPERHEAD (W 30.) (L 1) UH1)
      (HEAD (W 8.) (L 4) HD1)
      (PHOTO (W 20.) (L 2) PT1 LEFT1))
(DATA BB 300, P1 (HEAD (W 10.) (L 4) HD21)
      (DATA CC 290, P2 (HEAD (W 8.) (L 3) HD31))
      (DATA DD 120, P3 (NONE))
      (DATA EE 80, P3 (NONE))
      (DATA FF 200, P0 ((1 14.) (100, 15.1))
      (DATA GG 40, ((11, 12.) (100, 13.1))
      (DATA HH 40, P0 ((91, 1) (100, 41))
      )
```

図8 初期状態(WM) (割付け方式2)

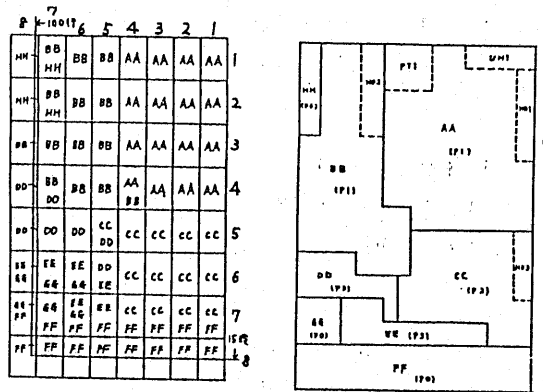


図9 記事の割付け結果 (割付け方式2)

各ブロック内の文字は, 記事名を表すが, たとえばブロック(4 4)は記事AAと記事BBが使用することを示す。

この実行例では前述した評価関数が比較的良好な評価値を示し, バックトラックも少なく良い割付けを行なった。方式2は, 知識の階層性に相当する抽象レベルを5つ設定したことで, 方式1よりも階層的思考法に近い処理が行なえ, HIPSの階層的構造の特徴を生かした問題解決を行なっており, ブロック単位の割付けによって, 方式1よりもルール作成が容易になり, ルールのReadabilityが改善された。

しかし5つの抽象レベルを設定したため、結局5回の割付け作業を行わねばならず解への探索路の短縮という意味では、効率の向上がなされたが、全体の処理効率が悪くなったかどうかは明らかではない。

4. 結論

階層のプロダクションシステムHIPSとHIPSの知識ベースシステムへの応用例として2種類の新聞記事割付け方式について述べた。HIPSは、基本PSを縦横2方向レベルに配列した構造を持ち、人間の持つ問題固有の階層的知識、及びそれらの知識の使い方に関するヒューリスティックな知識をプロダクションルールの形式で表現できる。方式2でのヒューリスティックルールの評価関数により、適切なルールが選択され、適当な解を避けることができ、この点においては従来のPSよりも問題解決の効率を上げることができた。

しかし、記事の割付けに関する処理の大部分がLISP関数で記述されており、ルールのReadability が犠牲にされ、抽象レベル設定による処理効率の向上が明らかでないことを含めて、全体の処理時間がかかりすぎ、実用的システムとして作動させることはこのままでは困難である。

謝辞

最後に新聞記事割付け方式を作成するにあたり、貴重な援助をいただいた大阪大学基礎工学部情報工学科豊田順一助教授に深く感謝致します。

参考文献

- (1) Davis, R. and J. King : " An overview of production systems ", Machine Intelligence, 8, pp.300-332 (1977) .
- (2) Erman, L.D. and V.R. Lesser : " A multilevel organization for problem solving using many, diverse, cooperating sources of knowledge " , Proc. of the 4th IJCAI, pp.483-490 (1975) .
- (3) Feigenbaum, E.A., B.G. Buchanan, and J. Lederberg : " On generality and problem solving---A case study involving the DENDRAL program " , Machine Intelligence, 6, pp.165-190 (1971) .
- (4) Hayes-Roth, F. and V.R. Lesser : " Focus of attention in the Hearsay- II " , Proc. of the 5th IJCAI, pp.27-35 (1977) .
- (5) 溝口, 大上, 冨田, 西山, 角所, 「階層のプロダクションシステム-HIPS」, 情報処理Vol.23, No.2 (1982) ;
- (6) Sacerdoti, E.D. : " Planning in a hierarchy of abstraction spaces " , Artificial Intelligence, 5, pp.115-135 (1974) .
- (7) 佐藤泰介, 「プロダクションシステムの試作とその使用経験」, 情報処理学会, 人工知能と対話技法研資3-1 (1978) .
- (8) Shortliffe, E.H. : " MYCIN : A rule-based computer program for advising physicians regarding antimicrobial therapy selection " , Stanford Univ. Computer Science Report, CS-74-465 (1974) .
- (9) 辻井潤一, 「プロダクションシステムとその応用」 情報処理 Vol.20, No.8, pp.735-743 (1979) .
- (10) 豊田順一, 伊藤哲郎, 篠原 守, 「新聞紙面等を対象とする二次元レイアウト問題」, 情報処理全国大会 (1978) .
- (11) Waterman, D.A., and Hayes-roth, F. : " Pattern-Directed Inference Systems " , Academic press New York (1978) .