

# あゝ囲碁練習材のデータ構造とその記述言語

佐々木 彬夫(フリー), 谷 信幸(熊本県立大野中学校), 富田 真次(ソニー), 塚村 善弘(ソニー),  
多田 碩佳(ネオログ電子), 松沢 孝(ネオログ電子)

## 1. はじめに

囲碁の論文が小<sup>(1)~(20)</sup>, ゲームの研究のサーベイでもそれらに触れようになった<sup>(1)</sup>。これまでは、コンピュータに碁を打たせようとする研究が多いが「初学の幼児の域を出し強さのプログラムは存在しない<sup>(1),(10),(7),(5)</sup>。

囲碁では対局者が白石と黒石を交互に置いて行くが、図1に示した様な標準手順が数多く存在<sup>(18)</sup>、その学習なしでは上達は望めない。ここで報告するシステムは、このような標準手順を教科書で理解/記憶するための助けとなるためのCAIシステムであり、コンピュータに碁を打たせようというものはない。

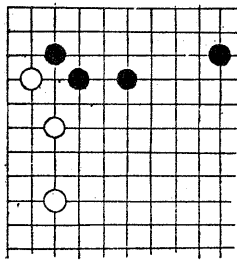


図1 囲碁の標準手順の一例

本システムでは、問題毎にデータをロードして学習を行なうがその問題に因る教科書の記述内容をコンピュータ向きに転記する記述能力の高い言語GOLOGを得意な事にしていよ。GOLOGで転記した教科書の内容をGOLOGのプロセッサで処理して上記のデータを得る訳であるが、試作材の経緯から実用材ではGOLOGが必須の工場のツールと考へられ。

標準手順記憶用のCAIと、対局の終り手順を再生表示して行く機能を持つシステムは既に2,3市販されている。表示にはLEDや液晶パネルが用いられシステムの処理内容が公表されているものもある<sup>(16)</sup>。これらのシステムにおけるCAIは何れも標準手順の座標の並びを土通りの約束の形でROMに記憶して置き、ユーザが着手をたどる際に一致/不一致を表示するのみである。これに対し、本システムでは

着手に因り教科書に記載してある全ての枝分れに対応して種々の表示/応答を行なうようになっている。また、付帯的な機能として「待った」ができる他、標準手順を碁盤上の任意の隅と向きで学習できるようにしている。隅と向きを選択は最初の着手する着からコンピュータがユーザの希望を自動的に識別する。

碁がコンピュータにとって扱いにくい根本的理由は、場合の数が非常に多いという事である。例えは図2は黒から打ち出して白を取るといふ詰碁で、初段程度の機材(最も悪いハンデキャップを望むはプロの高段者とさうして対等に対局できる程度)があれば、10分間位眺めてこれは最後まで分る問題で、X印は第1着手として可能性のあるところである。この例では第1着手は11通り、第2着手は10通り、...となり場合の数は11! ≒ 4000万通りとなる。これを図3のようにツリーとして表示すると問題の深刻さが良く分る。

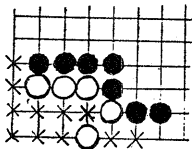


図2 詰碁の例

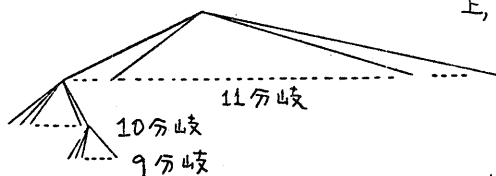


図3 着手のツリー

問題は深刻なのであるが、人間が考へる時には各ノードにおいて高々2,3のブランチしか考慮の対象としない。すなわち、人間は図3のような巨大なツリーの極めて微小なしかし根から葉まで届くサブツリーを抜き出して考へ、唯一通りの<sup>(17)</sup> 答えを短時間で見出す卓越した直観力を持っている。囲碁の教科書を置く場合にもこの事が前提であり、執筆者は自分の直観に従って根から葉まで届く微小なサブツリーを撰んだ

上、そのサブツリーの各ノードにおかれ高々2,3のブランチにつき主観的な説明をすよに過ぎない。従ってサブツリーの撰択の仕方と、各ノードにおいて説明するブランチの撰択が教科書のノードと、各ブランチに因る説明の内容がその並(り易さ)という事となり、教科書の書き方は無限の事となる。

本システムでは教科書の著者が取り上げたサブツリーを記憶

すよと共に、各ノードにおける複数のブランチに対する著者のコメントを①あまじき悪手、②向題あり、③別の話、④正解に分類して記憶している。

ユーザは、盤面の各交点にスイッチを埋め込んだ「基盤」上で自分の着手は任意の所に、コンピュータの応着はコンピュータが盤面のLEDに表示した所に置いて行く。コンピュータはユーザの打着を読み取った上、①の場合はアラー-4、②と③の場合にはそれぞれ別のアラー-4と教科書中でそれに関する著者の説明がどのページを表示した上、応着を盤面のLEDに出かす。④の場合は単に応着を表示し、①の場合には応着を表示しない。

教科書の著者が取り上げたサブツリーに含まれない打着は全て④として扱う。ユーザはアラー-4と教科書のページの表示を見ながら各ブランチの意味を学習して行くが、著者の選択したサブツリーのどのブランチに踏み込んでコンピュータは教科書の内容を交映した応着とアラー-4を表示出かす。現在市販されているシステムは全て図3のツリーのうち、根から葉に達する唯一のパスを記憶したものごと、サブツリーの形状も、各ブランチに関する情報も記憶していない。また、上記の付帯的な機能も持っていない。

## 2. ハードウェア

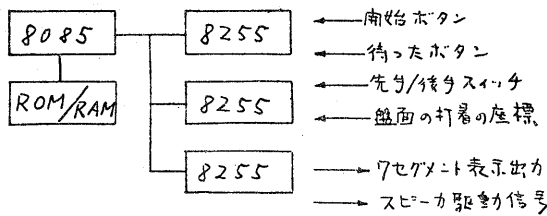


図4に試作機のハードウェア構成を示す。コンピュータ・システムとしては8085が8255経由で接点情報の入出力を行なうたけで、別に変わった所はない。メモリーはROMを2KB、RAMを1KB実装した。

図4 ハードウェア構成

## 3. 教科書の論理構造と試作機のデータ構造

図4の教科書は執筆者が撰んだサブツリーの各ノードにおける個々のブランチに関する説明を集積したものであから、その論理的な構造を図示すると図5のようになる。

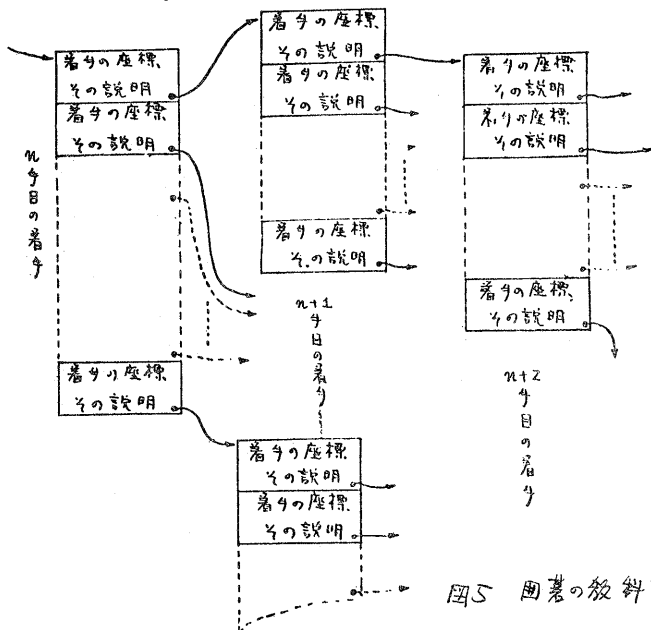
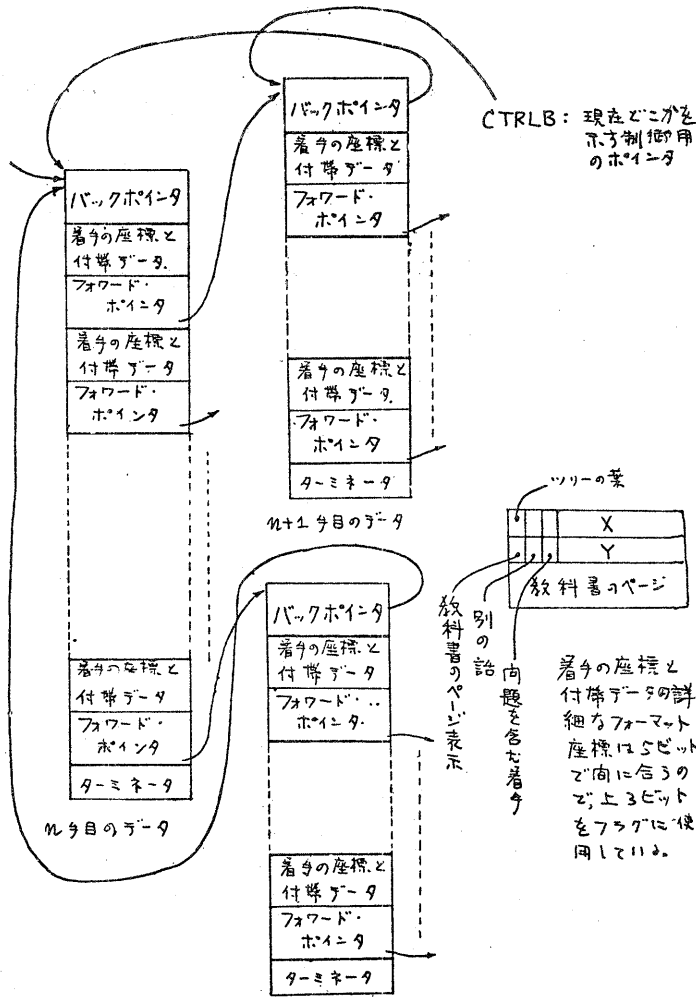


図5の各ブランチの説明は、サブツリー自体の撰択と同様に人間の高度な直観を前提にした(コンピュータにとっては極めて難解な)文章であり、多くの場合1枚の板の図でそのブランチの付近について「以下悪くない」とか「これで厚い」といった調子である。そしてその結論は、「向題を含む着手」、「それは別の話」、「正着」の何れかである。執筆者が随わていない(大多數の)ブランチは全て「あまじき、考慮に値しない悪手」という事になる。

本システムではこのような観点から、図6のようなデータ構造

図5 図4の教科書の論理的な構造

で教科書の執筆者が探んだサブツリーを表現している。そして、ユーザの着字に対して者とセグメント表示により、①「あまじき悪字」、②「向題を含む着字、教科書のページ」、③「別の標準字順、教科書のページ」、④「正着」を表示する事とした。



ちなみに、コンピュータにとって極めて難解な説明は教科書の執筆者に任せ、ユーザの着字に関して説明を要する場合には教科書のページを表示するのみである。

ユーザの着字が正着の場合、コンピュータはポインタをたどって次のノードの最初のブランチの着字を盤倒のLEDに表示出かし、ユーザがその場所にコンピュータの竹りに石を置いて行く。その時に石を正しく置いたかどうかもチェックしている。

図6のデータ構造は、N文字目とN+1文字目の着字の論理的な関係を示すものであが、現在どににしているのは制御用のポインタ CTRLB が指している。

次の着字に行く時には、フォワードポインタの値を CTRLB に転送して行く。

電源投入リセット

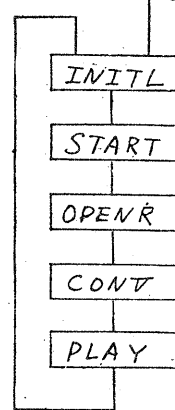


図6 試作機の詳細データ構造

#### 4. 学習の処理

図7で INITL は初期処理、START は開始ボタン待ち、OPENR はユーザの先字/後字の判定、CONT は学習を行なう際と向きの自動判定、PLAY がユーザに学習をさせようとする時である。

学習が終了すると(教科書の執筆者が探んだサブツリーの葉に達すると) INITL に戻り、ユーザは先字/後字を設定し直した上で再び任意の隅と向きで学習を開始できる。

プログラムの全体のサイズは約 1.5KB である。

図7 学習の処理

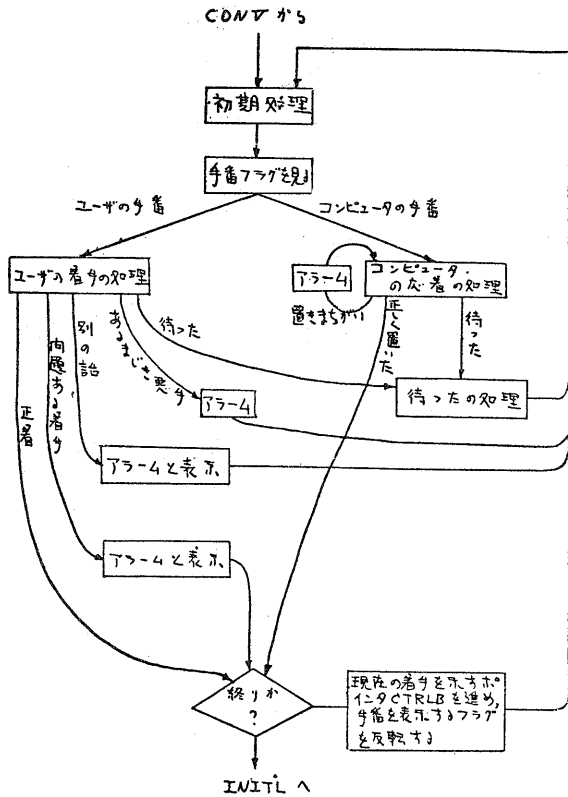


図8 PLAYのフロー

図8に学習の処理を行なうPLAYの部分のフローを示す。PLAYでは先ずユーザーの置いた石がコンピュータの手番かユーザーの手番かを、手番フラグで調べて分岐する。コンピュータの手番の場合には盤側のLEDの表示を力通り正しく置いたかどうかチェックし、正しければその着手が学習の終りかどうか見よ。終りならINITIALに戻り、終りでない場合は次の着手にあつてコンピュータの始まりを指すポインタCTRLB(図6参照)を更新した上、手番フラグを反転してPLAYの始めに戻る。

ユーザーの手番の場合には、「あまじき着手」と「別の話」の場合には、必要なアラー4の表示をした後PLAYの始めに戻り、「向題ある着手」と「正着」の場合には学習の終りとなれば上記CTRLBを更新した上、やはり手番表示フラグを反転してPLAYの始めに戻つてよい。

何れの手番の場合にも、盤上の着手を読み取りのと併せて待つボタンを読み取り、ユーザーが待つボタンを押した時には後述の「待った」の処理へ

分岐してよい。 盤上の石で生ずるパターンには必要がないので記憶してない。

### 5. 学習を行なう隅と向き of 自動機

図1に1例を示した標準手順は、実際では教科に記載の図とは異なる隅と向き、色で打たれ事が多く、そのために混乱して法を覚える事になりがちである。

図9に隅、向き、色が異なる同形の標準手順の例を示す。一般にあつた標準手順と同形の手順は8通りあり(4隅と左右)、任意の向き、隅、色の組み合わせで練習を重ねる事が重要である。試作機では、ユーザーがどこに石を置き始めよかにより自動的に隅と向きを機が選択、追従するようにしてよい。

左上隅で右向き of 標準手順を基準とすれば、容易に検証できるよりに次の何れかの座標変換で他の隅と向き of 標準手順に写像できる。

$$\begin{aligned}
 T_1: x' = x, y' = y; & \quad T_2: x' = y, y' = x; \\
 T_3: x' = 20 - x, y' = y; & \quad T_4: x' = y, y' = 20 - x; \\
 T_5: x' = x, y' = 20 - y; & \quad T_6: x' = 20 - y, y' = x
 \end{aligned}$$

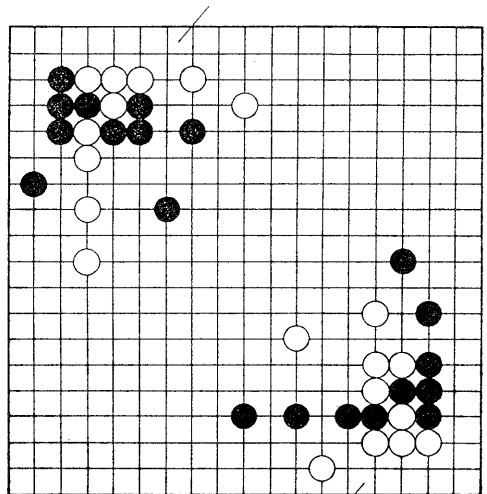


図9 隅と向きが異なる同形の手順の例

$$T_7: x' = 20-x, y' = 20-y; \quad T_8: x' = 20-y, y' = 20-x.$$

また,  $T_1 \sim T_8$  の逆変換は次の  $T_1^{-1} \sim T_8^{-1}$  であり事も容易に検証できる。  $T_K^{-1} = T_K$  ( $K \neq 4, 6$ ) であり所が面白い。

$$T_1^{-1}: x' = x, y' = y; \quad T_2^{-1}: x' = y, y' = x; \quad T_3^{-1}: x' = 20-x, y' = y; \quad T_4^{-1}: x' = 20-y, y' = x; \\ T_5^{-1}: x' = x, y' = 20-y; \quad T_6^{-1}: x' = y, y' = 20-x; \quad T_7^{-1}: x' = 20-x, y' = 20-y; \quad T_8^{-1}: x' = 20-x, y' = 20-x.$$

人間は教科書記載の隅と向きで手順を記憶していて、実験では自分の着方には  $T_K$ , 相手の着方には  $T_K^{-1}$  を施しながら考えているのである(  $K$  は  $1 \sim 8$  の何れかの値で終始変りなり)。 読作材でもそのようにしており、最初回7のCONVでその回の学習に対してはどの座標変換の対を使用すべきかを決定している。

すなわち、コンピュータは左上隅、右向き標準手順を記憶(して)いて、コンピュータが先手学習を行なう場合でも第1着はユーザーに置かせて  $T_1, T_2, \dots, T_8$  を施して見よ。 ユーザーの置いた第1着が  $T_K^{-1}$  でメモ

リー内の第1着の座標に良れば、ユーザーは  $T_K$  と  $T_K^{-1}$  の対を使用するのは適当だということになるので、PLAY中の処理でそのようにする。 標準手順の第1着に関して、大抵の場合には上記のようにPLAY中で用いべき座標変換の対が決定できるが、容易に検証できるように、第1着が基盤の対角線上の場合には上記の変換は一意的でない。

変換が一意的でない場合には、第2着もユーザーに置かせて  $T_1, T_2, \dots, T_8$  を施して見なければならぬ。 第2着も対角線上で座標変換が一意的でない場合には、更に第3着もユーザーに置かせて、ユーザーが終始一貫頭の中で行なっている座標変換は何れであるかを判定しなければならぬ。

第3着も対角線上にくる標準手順は存在しないから、 $T_1 \sim T_8$  の上記の特殊性にも拘らず、実用材を作らば事が可能である。 ユーザーは終始一貫同じ座標変換を頭の中で行なっているから、速くも第3着までにはそれがどの座標変換であるかが決定でき、それ以降の着方については座標変換が一意的かどうか対角線上の着方の場合にも気にする必要はない。

## 6. 待ったの処理

本システムではユーザーが自分の着方とコンピュータの応着を交互に基盤上に置いて行くが、先生(コンピュータ)の応着を見て自分の着方が誤りであったと思つたならば、「待った」をして学習効果を上げる様になつていく。 基盤上の着方はそれを読み取つた上で図6の制御用のポイントCTRLBの指すデータ群と比較対照して処理(7)なので、CTRLBを元に戻せば任意の手数だけ以前の状態に戻る事ができる。 読作材では、「待った」の処理の基本となる1手前へ戻す操作は、今の着方に関するデータ群に付属(している)バック・ポイントの指す所へCTRLBを戻して行く。 (しかし、1手進む時には現在のデータ群のト・ア・アドレスをスタックに入れてから進む事にすれば図6のデータ構造からバック・ポイントを指略する事ができる)

## 7. 手抜き処理

「手抜き」とは、標準手順の進行途中でその手順と全く関係のない所に打たれること(致命的損害につながるかどうか)と云う事である。 少数の標準手順では、途中で1回ないし数回手抜きが事がその根幹になつていく。 読作材では手抜きの処理はしてないが、「手抜きボタン」と「手抜き表示灯」を設ければ、例之は  $x=20, y=20$  といった実在しない着点(基盤上の着点の座標は  $x=1 \sim 19, y=1 \sim 19$  である)に打たれる扱いで、実在する着点への打着と同様に処理が事ができる。 手抜きに対する教科書執筆者のコメントは、普通の着方に対するのと同じであるから、「石を盤上に置かないという着方」または「石を盤外に置くという着方」を上記のサブツリーに含めても、普通の着方と全く同様に処理を進めて行く事ができる。

## 8. コンピュータの擬人性の向上

読作料ではコンピュータの応着はその多量にあげた複数のプログラムのうち、図6のデータ構造で一番著しい蓄地に入って、その蓄地を表現出来るようにして、どのプログラムの蓄地を応着とすよが、その都度乱数で選択すればコンピュータの擬人性を上げることが出来る。このようにする事により、ユーザは先生が時にまちがえたり、ハメチを打ってきたりした場合にそれを正しくとがめるといふ学習をする事が出来る。

## 9. 教科書内容の記述言語 GOLOG

本システムでは、学習しようとする標準手順毎に教科書の記述内容に合わせて図6のような構造のデータをメモリーにロードしなければならぬ。 (しかし、読作料の経験では熟練した若し、蓄が強いプログラマでも教科書の文章から誤りなくデータをDB(Define Byte)やDW(Define Word)で蓄けよのは30着分位が限界であった。

従って、実用性を生産するに当っては教科書の内容を転記するためのプログラミング言語と、その言語で蓄いたソースデータを処理して上記のデータを生成するプロセッサが必要である。本システムでは、教科書の内容を表わす例として図10の様なツリー構造を図11のように書き表わすプログラミング言語 GOLOG とそのプロセッサを用意する事にしている。

図10は“大斜”という名前の標準手順を表わすツリーであるが、最初の2個の数字は学習しようとする着々の基本形にあつて座標である。

その次の文字は、R: 正着(省略可), Q: 内題ある着々, D: 別の話である。 Pに続く数字は表示すべき教科書のページであり、Pと数字は書いても書かなくても良い。 あゝノードで2以上のプログラムの正着の場合にも記述に支障は生じない。

SKIPは分岐を意味し、STOPはその着々が教科書の執筆者が撰んだ“サブツリーの葉である”事を意味する。(STOP)は、その葉であるが教科書に「標準手順以後」の解説があり、それをコンピュータに入れよるを意味する。読作料では「標準手順以後」の処理はしていない。処理をするには図6のデータ構造に(ツリーの葉)フラグをもう1ビット付加すれば良い。

図10のツリー中、太線のパスが正着のシーケンスであり、これが実用に教科書の説明と共に頭に刻み込まべき“大斜”の左上隅右向き形である。

図11は図10のツリーをGOLOGで記述したものである。 図11中のBRA: とNCH:

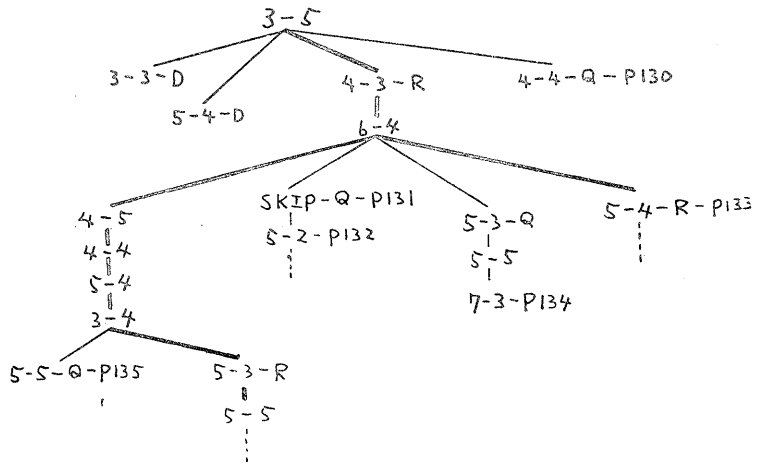


図10 “大斜”という名前の標準手順を表わすツリー

は、図10中のマル4・7・3・4を表記する括弧の役割をしよう。

/\* TAISHA \*/

3-5, /\* MOKU HAZUSHI \*/

BRA: 3-3-D, STOP /\* SAN SAN NO KAKARI \*/

5-4-D, STOP /\* TAKA GAKARI \*/

4-3-R, 6-4, /\* FUTSU NO KAKARI, TAISHA GAKE \*/

BRA: 4-5, 4-4, 5-4, 3-4, /\* TAISHA HYAPPEN \*/

BRA: 5-5-Q-P135, STOP /\* UE TSUGI \*/

5-3-R, 5-5, ..... /\* FUTSU \*/

NCH:

SKIP-Q-P131, 5-2-P132, (STOP), ..... /\* TAISHA TENUKI \*/

5-3-Q, 5-5, 7-3-P134, STOP /\* TAISHA WA IYA (A) \*/

5-4-R-P133, ..... /\* TAISHA WA IYA (B) \*/

⋮

NCH:

4-4-Q, STOP /\* HOSHI NO KAKARI WA DAME \*/

NCH:

END,

図11 図10のツリーをGOLOGで表記したものを

図10のようなツリーを図11のようにGOLOGで表記する場合にSTOPは書かない事にしてGOLOGのプロセッサの処理に支障はないが、書く事にした方がソースデータが読み易くなる。また、BRA: と NCH: の向で INSERT SUBTREE などと書いて、別の所に記述してあるより小さなサブツリーをはめ込むようにすれば、教科書の内容の転記と転記後の目視によるチェックが非常に楽になる。

GOLOGのプロセッサについては別の機会に報告する。

## 10. まとめ

囲碁の研究にはコンピュータに碁を打たせようという方向と、碁の練習にコンピュータを使おうという方向があるが、前者の方向の研究は進んでいないし、かなり難しいであろう。差し当り、実用性の高いシステムを作るという事と後者の方向という事になるが、これなどは充分な材能を持ったシステムが存在しなかった。

ここで報告した試作材は後者の方向のものであり、現在の材能と材能拡張の可能性は報告者の5, 3級-3段の棋力を有する3名が碁を遊ぶ議論を糧として積み上げたものである。ここで述べた他に横議(対局の録音)の再生と採録の材能を追加すれば、それ以上の材能は事実上不要であろう。

また、概要を述べたようにとめた記述言語GOLOGは誰かのように書いた教科書でも、その内容をコンピュータ向きに転記するのには充分な記述能力を持ってほしいと思われ、かつそのプロセッサの作成は容易である。初段程度の棋力があれば、図11の例がGOLOGの威力を良く表わしている事を汲み取れるであろう。

GOLOGが充分な記述能力を持ってほしいので、本システムの例から教科書の執筆に対して要求すべき執筆上の注意事項はないが、執筆者が本システムの存在を想定して従来よりも良い教科書を書く事は可能である。

図6に示したデータ構造は冗長度が高いので改良の余地があり、GOLOGのプログラマもそれを考慮して作成しなくてはならない。

ここで述べた試作料とGOLOGの検討を通じて得られた成果のうち最大のものは、囲碁の教科書と囲碁について何をどのように書いてあるものかが明確になった事かも知れり。

## 参考文献

- (1) 実近憲昭, 「ゲーム・プログラミング・プログラマの近年の成果」, 情報処理, 第20巻第7号, pp.601-611, 昭和54年7月。
- (2) Soule S., "The Implementation of a GO Board", Inf. Sci., 16, pp.31-40, 1978.
- (3) Thorp E. O., Walden W. E., "A Computer Assisted Study of GO on MxN Boards", Inf. Sci., 4, pp. 1-33, 1972.
- (4) Lichtenstein D., Sipser M., "GO is P-space Hard", Proc. 19th Ann. Symp. on Foundation of Computer Science, pp. 48-54, Oct. 1978.
- (5) Reitman W., Wilcox B., "Pattern Recognition and Pattern Directed Inference in a Program for Playing GO", Pattern-Directed Inference Systems Ed. Waterman D., Hayes H. Roth, pp. 503-523, Academic Press, 1978.
- (6) Thorp E. O., Walden W. E., "A partial analysis of GO", The Computer Journal, Vol. 7, No. 3, pp. 203-207, 1964.
- (7) Zobrist A. L., "A model of visual organization for the game of GO", Proc. SJCC '69, pp. 103-112.
- (8) Remus H., "Simulation of a Learning Machine for Playing GO", Proc. IFIP Cong. '69, pp. 428-432.
- (9) Benson D. B., "Life in the Game of GO", Inf. Sci., 10, pp. 17-29, 1976.
- (10) Reitman W., Wilcox B., "The structure and performance of INTERIM. 2 GO program", Proc. 6th IJCAI pp. 711-719, 1979.
- (11) Reitman W., Wilcox B., "Modeling tactical analysis and problem solving in GO", Proc. 10th Ann. Pittsburgh Conf. on Modeling and Simulation.
- (12) Reitman W., Wilcox B., "Perception and Representation of Spatial Relations in a Program for Playing GO", Proc. ACM Ann. Conf. pp. 37-41, 1975.
- (13) Starkey J. D., "The X-O-O Heuristic in Game Tree Analysis", ?
- (14) Benson D. B., Bruce R., "Hilditch J., Starkey J. D., "Tree Analysis Techniques in Tsunego", ?
- (15) Brown D. J. H., "Hierarchical reasoning in the game of GO", ?
- (16) 野原 孝, 「基トレナとステア教育について」, CAI学会の発表資料
- (17) 大竹 英雄, 「囲碁の基礎訓練上巻」, 産報, 昭和50年。
- (18) 鈴木 爲次郎, 「囲碁大辞典」7版, 誠文堂新光社, 昭和50年

## 謝辞

この報告をまとめるに当り, 電子技術総合研究所の大塚氏, 実近氏, 島井氏には御討論を頂いた上, 資料を提供頂いた。またソニー株式会社の高崎氏, 日本電子販売株式会社野口氏並びに第一家庭電器株式会社の鈴木氏には御援助頂いた。