

# 知識と用語の相互獲得モデルとその詰碁への適用

小島 琢矢 ・ 吉川 厚

NTT コミュニケーション科学基礎研究所

kojima@rudolph.brl.ntt.co.jp

## 概要

従来、人間の初心者と熟達者では保持している知識量が異なることは知られていた。我々はこれまでに熟達度をより細かく調べ、知識を記述する囲碁用語の量やその意味する内容が熟達するにつれて大きく異なるという知見を得た。そこで、囲碁用語の獲得過程に注目して人の熟達化のモデルを提案する。本モデルは、知識を獲得し、その結果用語を獲得し、その用語を用いてさらに知識を獲得し、その知識からまた用語を獲得するという囲碁知識の獲得と囲碁用語の獲得が交互になされるモデルである。本モデルを検証するために詰碁に適用した。まず、盤面や石などの基本的な入力情報から知識を獲得する方式を提案し、実装した。次に、得られた知識から圧縮により単純な用語を獲得する方式を提案し、実装した。得られた知識を評価したところ適切であると判断された。更に、得られた用語を用いて知識を記述し直すことで用語を用いた知識を獲得する方式を提案した。最後に、その用語を含む知識を一般化することで、複雑な用語を獲得する方式を提案した。以上のように、本モデルの一部を実装し、また具体的なアルゴリズムを提案した。

## A Model of Inter-acquisition of Knowledge and Terms and its Application to Tsume-Go

Takuya KOJIMA      Atsushi YOSHIKAWA

NTT Communication Science Laboratories

kojima@rudolph.brl.ntt.co.jp

## Abstract

The purpose of this research is to build a model which simulates the processes by which human novices become experts. Our cognitive studies on Go have shown that the differences lie in the amount of Go terms they can explain and in how they ascribe meaning to the Go terms. Therefore we have proposed a model to acquire Go terms. In the model, knowledge without Go terms is initially acquired, and Go terms are acquired from this knowledge, and new knowledge is subsequently acquired by using the acquired terms, and so on. The model acquires Go knowledge and Go terms alternately. In this paper, we applied this model to Tsume-Go. We implemented a mechanism, compression, to acquire simple Go terms from pattern knowledge, and the acquired terms were evaluated as appropriate. We also proposed a mechanism of generalization.

# 1 はじめに

人間の初心者は、自律的に学習することで徐々に熟達者へと変わっていく。初心者と熟達者では持っている知識に量的な違いがあるという研究報告から [1], 我々は知識を獲得するアルゴリズムを提案してきた [7, 3, 4]。しかし、我々の囲碁を題材とした認知科学的な研究では、知識を記述する囲碁用語 (囲碁特有の概念を指し示す語彙) にも大きな違いがあることが分かってきた。つまり、円滑に使用できる囲碁用語の数や、その使い方、意味が大きく異なるのである [5]。このことから、熟達化の過程では、囲碁用語が獲得されると同時にその意味内容も変化すると考えられる。そして獲得した囲碁用語の意味の差が実力の差に大いに関係することも示唆されている。そこで本研究では、囲碁用語の獲得過程に着目して熟達化のモデル化を行なう。

# 2 知識と用語の相互獲得モデル

我々は、[8]において、知識獲得と用語獲得が交互に繰り返されながら、徐々に複雑な知識と用語が獲得されていくモデルを提案した。本論文ではこれを詰碁に適用する。

全体のイメージを図1に示す。まず、問題図として、石の配置とその答えが与えられる。ここでの用語は石そのものである。次に、石の配置を If 部と Then 部に持つ知識 (パターン知識) を獲得する。次に、その獲得した知識同士の共通部分を用語として抽出し、基本的な用語の獲得を行なう。次に、その獲得された用語を用いた知識を獲得する。この知識は石の配置と用語が両方含まれているので、ハイブリッド知識と呼ぶ。更に、このハイブリッド知識を元により複雑な用語を獲得する。これを繰り返し、徐々に複雑な知識と用語を獲得していく。

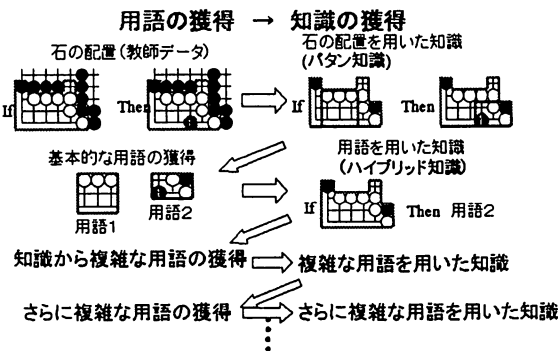


図 1: 囲碁知識と囲碁用語の相互獲得モデル

# 3 モデルの詰碁への適用

本モデルの有効性を検証するために実験を行なう。19路盤の本碁では着手の有効性の検証が難しいため、本研究では、正解・不正解が判定できる詰碁問題を対象領域とする。今回与えた詰碁とは図2のようなものである。番号のふっていない石が問題図で、番号がその回答手順である。

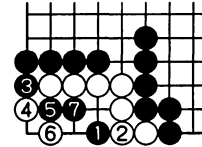


図 2: 詰碁の問題と回答の例

## 3.1 パターン知識の獲得

詰碁のパターン知識を獲得する手法は、[4, 3]で提案してきたが、これは遺伝的アルゴリズムに類似したヒューリスティック探索手法であった。今回は全探索をする手法を提案する。これは、用語獲得のためにパターン知識を網羅的に学習する必要があるからである。

### 3.1.1 知識表現

パターン知識は If-Then 形式のプロダクションルールで、下記のように記述できる。<sup>1</sup>

IF exist([0, 0], obj<sub>0</sub>) ∧ ([x<sub>1</sub>, y<sub>1</sub>], obj<sub>1</sub>) ∧ ... ∧ exist([x<sub>n</sub>, y<sub>n</sub>], obj<sub>n</sub>)  
THEN play([x<sub>a1</sub>, y<sub>a1</sub>] [x<sub>a2</sub>, y<sub>a2</sub>]...)

[x<sub>i</sub>, y<sub>i</sub>] は、知識の最初に記述された位置からの相対座標である。obj<sub>i</sub> は、SAME(着手する方と同じ色の石)、DIFF(相手側の石)、EDGE(盤端<sup>2</sup>)、VACANT(空点)のいずれかである。Then 部には、着手する手が n 手 (1 ≤ n) 並ぶ。すなわち、複数手を実行部を含むシーケンス (手順) 知識も表現できる。図3に例を示す。枠内が知識で、枠外は don't care である。

### 3.1.2 知識評価の基準

知識評価の基準として、頻度と正解率を用いる。頻度とは、全教師データに対し If 部が match する回数のことである。正解率とは、(If 部と Then 部の双方が match する回数)/(If 部が match する回数) のことであ

<sup>1</sup> 囲碁は回転・鏡像を考慮にいれると 8 個同様の知識があるので、そのうちの一つのみを獲得する。

<sup>2</sup> 盤のすぐ外側の点。盤上ではない。

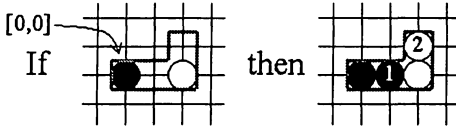


図 3: 知識の例: IF exist([0,0],SAME)  $\wedge$  exist([2,0],DIFF)  $\wedge$  exist([1,0],VACANT)  $\wedge$  exist([2,1],VACANT) THEN play([1,0] [2,1])

る。データマイニング [2] の分野でよく用いられる相関ルールでは、頻度を支持度 (support), 正解率を確信度 (confidence) と呼び、同様に評価の基準としている。

ここで良い知識とは、頻度が大きく、かつ、正解率の高い知識のことである。本研究では、頻度がある閾値 (例えば、全教師データの 1%) 以上で、かつ、正解率がある閾値 (例えば 0.1) 以上の知識を獲得することを目標とする。

### 3.1.3 知識獲得の方法

知識は以下の手順で獲得される。単純な知識から分裂という操作により、徐々に複雑な知識を獲得していく。

1. 最も単純な初期知識を与える
2. 知識の頻度の計算
3. 頻度が閾値を越えている知識は分裂し、少しだけ複雑になる
4. 2へ戻る

初期知識として、以下の最も基本的な 4 つの知識を与える (図 4)。

1. IF exist([0,0],EDGE)  $\wedge$  exist([1,0],VACANT) THEN play([1,0])
2. IF exist([0,0],SAME)  $\wedge$  exist([1,0],VACANT) THEN play([1,0])
3. IF exist([0,0],DIFF)  $\wedge$  exist([1,0],VACANT) THEN play([1,0])
4. IF exist([0,0],VACANT)  $\wedge$  exist([1,0],VACANT) THEN play([1,0])

分裂とは、If 部を 1 つ増やす (新たな条件を 1 つ加える), または、Then 部を 1 つ増やす (着手のシーケンスを 1 手延ばす) 操作である。If 部を増やす場合は、知識の If 部と Then 部が訓練例に match した状況で、If 部の周囲から 1 つの obj (石, 空点, 盤端) を条件部に加える。これは、図 3 で言えば、枠を一つ広げる操作に相当する。Then 部を増やす場合は、現在の最後の手を一手延

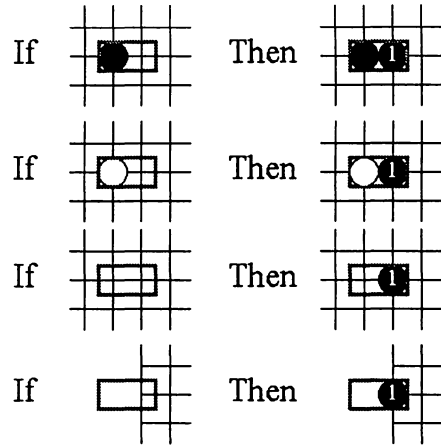


図 4: 初期知識

ばす。この操作は、各知識が訓練例に match した全ての場面について、可能な分裂を全て行なう。

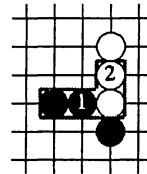


図 5: match した場面

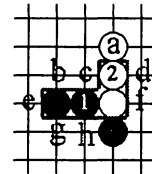


図 6: 分裂候補

図 3 の知識を例にとって説明する。この知識が、図 5 の場面に match したとする。この知識が分裂する場合、図 6 で a ~ i までの 9 箇所が加わる条件部の候補となる。その候補位置に例えば相手の石があれば、その座標に相手の石がある、という条件が加わった知識に分裂する。この図の例の場合、図 7 のような 9 個の知識に分裂する。

分裂は、頻度の閾値を越えたものについてのみ行なう。分裂は、知識の条件を増やすので、頻度は単調減少し、いったん頻度が閾値よりも小さくなると閾値以上になることがないからである。また、現時点で正解率が低くても分裂によって正解率は上がるので、正解率が閾値を越えていないものも分裂する。

### 3.1.4 結果

教師データとして、詰碁 1038 問 (全教師データ数 3993 手, 1 問あたり平均 3.8 手) を用いた。頻度の閾値は、全教師データ数の 5% (=200) 以上, 正解率の閾値を 10% とした。

閾値を越える知識が無くなるまで分裂させた結果得

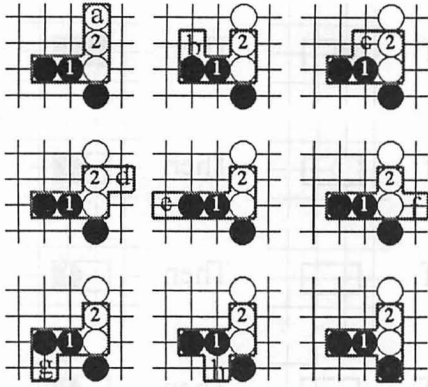


図 7: 分裂の例

られた知識数を図に示す。図 8 では、獲得された全ての知識数を示しており、図 9 では、そのなかで、頻度と正解率の閾値を越えていた知識の数である。合計、3,636 個の条件にあった知識が得られた。

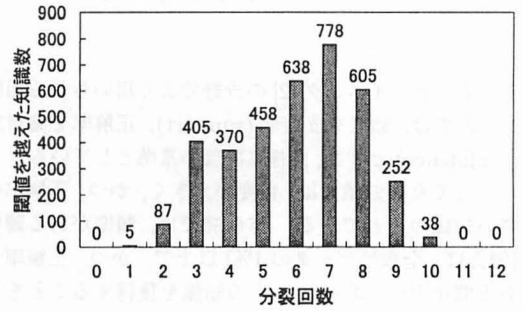


図 9: 分裂回数と閾値を越えた知識の数

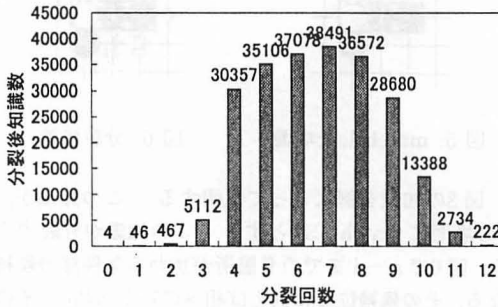


図 8: 分裂回数と獲得されたすべての知識の数

所要時間は、SUN Ultra SparcII での実時間で全体で約 31.5 時間かかった。獲得された知識の例を図 10 に示す。

初期状態では条件部が 2 つの知識であり、分裂するごとに条件が 1 つずつ増えていく。分裂の回数を増やす毎に分裂後に得られる知識数は増えていく。これは、条件部が増えることによって可能な知識のバリエーションが増えるためである。7 回の分裂時に分裂後知識数は最大になり、後は単調減少する。これは、条件が増えることによって頻度の閾値を越えられなくなる知識が増えるためである。

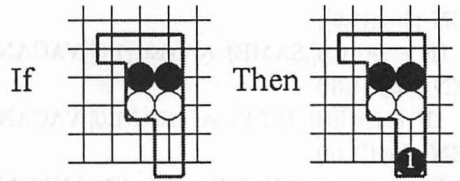


図 10: 獲得された知識の例。頻度 203, 正解率 21.6%

### 3.1.5 獲得された知識による詰碁の回答

獲得された知識により、訓練例には含まれない300問の詰碁を解いた。着手  $m$  の着手率を  $P_m (0 \leq P_m \leq 1)$  とすると、全ての可能な着手に関して以下の着手率  $P_m$  を計算し、最大のものを回答手とした。知識  $k$  の正解率を  $c_k$ 、着手  $m$  に知識  $k$  が match することを  $match(m, k)$  と書くと、

$$P_m = 1 - \prod_{k \in match(m, k)} (1 - c_k)$$

知識の正解率の閾値を変えた時の詰碁の正解率の変化を図11に示す。閾値が上がると用いる知識数は当然減ることになる。正解率20%以上の知識を用いたときのパフォーマンスが最大になっている。閾値をこれ以上上げると用いる知識数が減りすぎたために正解率が下がったものと考えられる。この問題を人間に解かせた場合[9]と比較すると、およそ人間の2級程度のパフォーマンスが得られた。

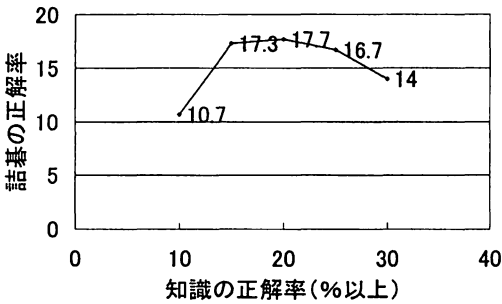


図 11: 詰碁の正解率

## 3.2 パターン知識からの用語獲得

次に、3.1節で獲得された知識から用語を獲得した。[8]では、圧縮、一般化、詳細化の3つの知識からの用語獲得方法を提案したが、ここでは圧縮による用語の獲得を行なう。

用語には「アキ三角」や「一間トビ」などの状態を表す用語と「ツケ」や「ハネ」などの行動を表す用語がある。状態を表す用語の圧縮では、知識のIf部に頻りに現れる盤面状況を用語に圧縮する。ここでの用語は、複数の石の配置を一つの用語に置き換えることに相当する。その結果、記述が長い知識が、より記述の短い知識に圧縮され、記憶容量の節約につながる。行動を

表す用語の圧縮では、知識のThen部に頻りに現れる着手を用語に圧縮する。この場合、着手がなされる盤面状況と着手が一つの用語に置き換えられる。

### 3.2.1 用語の表現

今回は、行動を表す用語の獲得を行なった。行動を表す用語は以下のように表現している。知識とはほぼ同様だが、相対座標の原点がThen部になっているところが異なる。

```
IF exist([x1, y1], obj1) ^ ... ^ exist([xn, yn], objn)
THEN play([0, 0])
```

例えば、図12の①の着手は「ハネ」と呼ばれるが、これは以下のように表現される。

```
IF exist([1, -1], SAME) ^ exist([0, -1], DIFF)
THEN play([0, 0])
```

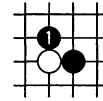


図 12: ハネ

### 3.2.2 用語評価の基準

用語にはこれを獲得すべきという正解がない。そのため、頻度のみを基準として用いる。頻度とは、全知識に対し用語のIf部とThen部がmatchする回数のことである。ここでは、頻度がある閾値(例えば、全知識の1%)以上の用語を獲得することを目標とする。

### 3.2.3 用語獲得の方法

知識獲得と類似の手法を使う。手順は以下のようになる。

1. 最も単純な初期用語を与える
2. 用語の頻度の計算
3. 頻度が閾値を越えている用語は分裂し、少しだけ複雑になる
4. 2へ戻る

初期状態では、以下の4つの用語がある。

1. exist([1, 0], EDGE) THEN play([0, 0])
2. exist([1, 0], DIFF) THEN play([0, 0])
3. exist([1, 0], SAME) THEN play([0, 0])

#### 4. exist([1,0],VACANT) THEN play([0,0])

分裂は知識獲得と同様の概念で、用語のIf部を1つ増やす操作である。用語のIf部とThen部が知識にmatchした状況で、用語のIf部、Then部の周囲から1つのobj(石,空点,盤端)を条件部に加える。この操作は、各用語がmatchするすべての知識について、可能な分裂をすべて行なう。

#### 3.2.4 結果

教師データとして与える知識は、3.1.4節で獲得した3,636個の知識である。頻度の閾値は、全教師データ数の1%(=37)以上とした。閾値を越える用語が得られなくなるまで実行した結果、図13に示すような用語数を得られ、合計412個の用語が得られた。その中には、図12で示したハネ(頻度465)など多くの囲碁で頻繁に用いられる用語が獲得された。用語数の推移は知識数の推移と似た傾向にある。各回の分裂後に新たに作られた用語の数が、分裂後用語数である。そのなかで、頻度の閾値を越えていた知識の数が、適切な用語数である。全体の所要時間はSUN Ultra SparcIIでの実時間で4分程度であった。

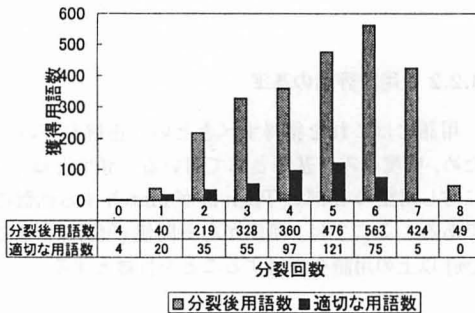


図13: 分裂回数と獲得された用語の数

#### 3.2.5 獲得された用語の主観評価

獲得された用語を筆者らが3段階で主観評価した。○は該当する用語があるもの。△は該当する用語はないものの、囲碁の手段として適切なもの。×は囲碁の手段として不適切なものとした。結果を図14に示す。頻度の閾値を変えたところ、頻度の高い方が適切な用語が多くなっている。また、頻度が5%以上の場合、半数以上が適切と判断され、本アルゴリズムの有用性がある程度示すことが出来た。

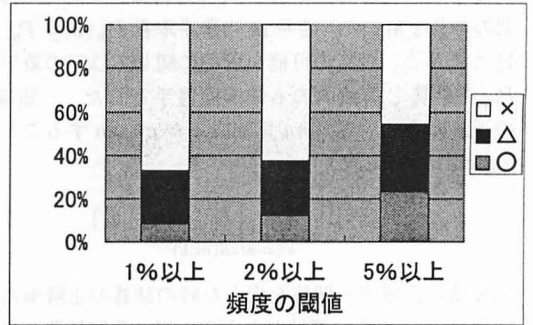


図14: 獲得された用語の主観評価

### 3.3 用語による知識の書き換えと一般化による用語獲得

3.2節で獲得した行動を表す用語は、知識の共通するThen部を抜き出したものである。そのため、獲得した用語を用いて知識のThen部を以下のように書き換えることができる。

```

IF exist([0,0],obj0) ∧ ([x1,y1],obj1) ∧ ... ∧
   exist([xn,yn],objn)
THEN play(ti)
    
```

$t_i$ は用語である。このように書き換えると、用語が曖昧な場合、複数箇所がmatchしてしまう場合も出てくる。これは用語のIf部が不十分であるためで、用語にとっての負の例とも考えられる。これを負の例と考えれば知識と同様に正解率が定義でき、より適切な用語を獲得する指標になる可能性がある。

上記のように知識を用語によって書き換えると、一般化がはじめて可能になる。知識が増えてくると、類似した条件部や実行部を持つ知識が増えてくる。これらの類似点を一つの用語に置き換えることによって、複数の類似した知識を一つの知識にまとめることを行なうのが一般化である。

一般化には大きく二つの場合がある。一つは、If部が類似しThen部が同一である知識をまとめる場合。この場合は、If部をまとめる用語を獲得する。もう一つは、If部が同一でThen部が類似している知識をまとめる場合。この場合は、Then部をまとめる用語を獲得する。

今回は行動を表す用語を対象としているため、Then部が同一の場合を考える。獲得した用語によって上記のように知識を書き換えると、Then部が同一かどうかの

判定は容易である。しかし、用語を用いないパタン知識のままでは Then 部は単なる座標であり、その原点は知識によって任意に取れるため、同一の座標であっても意味をなさない。一方、Then 部を行動を表す用語で書き換えた場合に Then 部が同じ用語で表されている場合は同一の行動と見なすことができる。そこで同一の行動をとる知識をまとめるために、類似している If 部をまとめる用語を導入する。これが一般化による用語獲得である。

### 3.4 用語を含む知識の獲得

用語獲得の次の段階として、獲得した用語を含むハイブリッド知識の獲得がある。知識獲得の分裂時に新たに加える条件として、パタン知識獲得の場合は obj(石や盤端、空点)のみを考えた。実行に用いる用語の他に状態を記述する用語が獲得された段階では、盤面を用語で記述することが可能である。分裂の際にこれらの用語を加えることによって、用語を含む知識獲得が可能になる。この用語を含む知識獲得では一度に広い領域を条件部に加えるため、効率よい知識獲得が可能になると予想される。

### 3.5 まとめ

以上の結果をまとめると図 15 のようになる。まず 3.1 節では、石や盤端などの基本的な情報からパタン知識を獲得するアルゴリズムを提案および実装した。これが、図中で最初の用語獲得(左)から知識獲得(右)への矢印にあたる。次に 3.2 節では、獲得した知識から用語を獲得するアルゴリズムを提案および実装を行なった。これは、図中で最初の知識獲得(右上)から用語獲得(左下)への矢印である。次に 3.3 節では、用語による知識の書き換えの方法を提案および実装した。これは図中で 2 番目の用語獲得(左)から知識獲得(右)への矢印にあたる。そして同じく 3.3 節では、先ほどの用語によって書き換えた知識を用いて知識の一般化を行なう方法を提案した。これは図中で 2 番目の用語獲得(右上)から知識獲得(左下)への矢印にあたる。また 3.4 節では、用語を含む知識獲得の方法を提案した。これは用語獲得(左)から知識獲得(右)への矢印にあたる。以上のように我々が提案した囲碁知識と囲碁用語の相互獲得モデルの一部を実装し、また具体的なアルゴリズムを提案した。

## 4 考察と今後の課題

囲碁用語を獲得することによって以下のような利点が生じると予想される。まず、メモリの節約が期待される。圧縮の場合、多くの石を 1 つの用語で置き換えるの

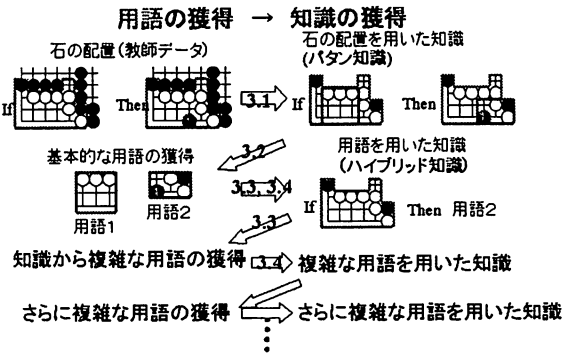


図 15: 本研究で扱った範囲

で 1 つの知識の記述量が減少する。また、一般化によって複数の知識を一つの知識にまとめることによっても全体の記述量は減少する。

次に、知識獲得の効率化が期待される。3.4 節で述べたように、知識獲得の分裂の際に用語を用いることで、探索空間を狭め、知識獲得を効率化できる。

更に、2 つの点でパフォーマンスの向上が期待できる。第 1 に、一般化によって適応範囲が拡大するため、用語を含む知識は一つの知識でより多くの場面に対応できる。また、一般化されているため、未知の場面であっても対応できる可能性がある。第 2 に、一般化によって正解率が高いが頻度が少ない知識をまとめた結果、正解率が高いまま頻度も高くすることが可能になると期待される。知識が分裂すると条件が増えるので、知識の使用頻度は少なくなり、正解率は高くなる。そのため、非常に正解率の高い知識であっても、頻度が少なくなり捨てられてしまっていた。ただし、一般化された用語を知識に導入することで、知識の頻度は高くなるが、正解率が落ちる場合がある。この場合は、用語を詳細化することによって、正解率を上げることができると考えられる。

以上の予想は実験によって確認されておらず、今後の課題である。特に、パフォーマンスの向上がどの程度なされるかが重要である。

## 5 結論

人間の熟達化の過程をモデル化するための重要なプロセスとして、用語獲得プロセスがある。我々は、囲碁用語の獲得と知識獲得が交互になされることによって、囲碁用語と囲碁の知識の両方が相互に獲得されるモデルを提案した。本モデルを検証するために、まず、盤面や石などの基本的な入力情報から知識を獲得する方式を提案し、実装した。次に、得られた知識から単純な用語を獲得する方式を提案し、実装した。更に、得られた

用語を用いて知識を記述し直すことで用語を用いた知識を獲得する方式を提案した。最後に、その用語を含む知識を一般化することで、複雑な用語を獲得する方式を提案した。以上のように、本モデルの一部を実装し、また具体的なアルゴリズムを提案した。

の検証。1997年度日本認知科学会冬のシンポジウム資料集「ゲームと認知科学」, pp. 25-31. 日本認知科学会, 1997.

## 参考文献

- [1] William G. Chase and Herbert A. Simon. Perception in chess. *Cognitive Psychology*, Vol. 4, pp. 55-81, 1973.
- [2] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. The MIT Press, 1996.
- [3] Takuya Kojima, Kazuhiro Ueda, and Saburo Nagano. An evolutionary algorithm extended by ecological analogy and its application to the game of Go. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 684-689. Morgan Kaufmann, 1997.
- [4] Takuya Kojima and Atsushi Yoshikawa. A two-step model of pattern acquisition: Application to Tsume-Go. In H. Jaap van den Herik and Hiroyuki Iida, editors, *Computers and Games*, No. 1558 in LNCS, pp. 146-166. Springer-Verlag, 1999.
- [5] Atsushi Yoshikawa, Takuya Kojima, and Yasuki Saito. Relations between skill and the use of terms - an analysis of protocols of the game of go -. In H. Jaap van den Herik and Hiroyuki Iida, editors, *Computers and Games*, No. 1558 in LNCS, pp. 282-299. Springer-Verlag, 1999.
- [6] 白柳潔. 碁における知識処理の基礎検討. Technical Report 12967, 日本電信電話株式会社 電気通信研究所, 1986.
- [7] 小島琢矢, 植田一博, 永野三郎, 吉川厚. 演繹的学習による囲碁知識の獲得とその修正. ゲームとモデル化シンポジウム論文集, 情報処理学会シンポジウムシリーズ, 第98-18巻, pp. 1-7, 1998.
- [8] 小島琢矢, 吉川厚. 囲碁における知識獲得と用語獲得の相互作用. 情報処理学会研究報告 ゲーム情報学 99-GI-1, 第99-53巻, pp. 71-78, 1999.
- [9] 小島琢矢, 吉川厚, 植田一博, 永野三郎. 囲碁名人のモデル構築に向けて: 認知科学と人工知能の融合