

## カードゲームをプレイするプロダクションシステムの学習方法

4 J-5

並川 青慈<sup>†</sup> 小堀 聡<sup>†</sup> 角所 収<sup>‡</sup><sup>†</sup>龍谷大学理工学部 <sup>‡</sup>兵庫大学経済情報学部

## 1 はじめに

学習については、さまざまな定義や分類が存在するとともに、それらを計算機で実現する方法も数多く提案されている。人間の学習過程については、「行動による学習」という視点でとらえようという考え方があり、(1) 初期問題空間の構成、(2) 良くない例の収集、(3) 良くない例を回避する手続きの獲得、良い例の収集と下位目標生成手続きの構成、下位目標構造内のパターンの発見、の段階があるとされている。

一方、プロダクションシステムは、人間の問題解決のモデルに用いられることがあるが、プロダクションシステムはルールに記述されていない状況に対して必ずしも正しい行動を取ることができないという欠点がある。そのような場合、人間であれば、その状況に対するルールを追加することで対処するが、プロダクションシステムにそのような学習を行わせる方法は、まだ一般的に確立されていない。これらの問題を解決するためにはシステムに学習の枠組を与え、事例から何らかの学習を行なう必要がある。

本研究においては、カードゲーム“Calculation”を解く人間のモデルをこの理論に基づいて提案し、計算機上のシステムへの応用を検討する。

## 2 Calculation の特徴

Calculation は一人で行なうカードゲームである。よく切った 52 枚の手札を 1 枚ずつめくり、4 列の「場」を作業空間として、4 列の「台」に予め定められた順番に置いていく。台札の並びは以下ようになる。

- 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
- 2, 4, 6, 8, 10, Q, 1, 3, 5, 7, 9, J, K
- 3, 6, 9, Q, 2, 5, 8, J, 1, 4, 7, 10, K
- 4, 8, Q, 3, 7, J, 2, 6, 10, 1, 5, 9, K

台へは列の先頭(上の列では左側)から順にしか置くことができない。場は先入れ後出しのスタックになっており、最後に置いたものから順にしか取り出すこと

A learning method for production system that plays card game.  
Seiji NAMIKAWA<sup>†</sup>, Satoshi KOBORI<sup>†</sup>

<sup>†</sup> Faculty of Science and Technology, Ryukoku University.  
Osamu KAKUSHO<sup>‡</sup>

<sup>‡</sup> Faculty of Economics and Informatics, Hyogo University.

ができない。全ての手札を台に並べられたら成功で、それ以外は失敗である。

一般にゲームが人工知能の題材とされるのは、ルールが明確であり、限定した世界の例として、部分的に人間の知能について分析し、実現するには、手ごろなものであるからである。それに加えて、Calculation の特徴として、

- 的確に場札を使用しなければ成功することができないという点で、高度な思考能力を要求される。
- 初心者の成功率は低いのにに対して、熟達者はかなり高い確率で成功することから、学習による効果が大きいといえる。

が挙げられる。

このことから、Calculation はゲームを解く人間の知的能力について考察し、それを具体的なシステムに応用することを検討する題材として適当であると考えられる。また、ゲームの成功率によってシステムを客観的に評価することができるという利点もある。

## 3 Calculation 解決のモデル

## 3.1 手続き的な解決のアルゴリズム

現在までに計算機での Calculation 解決アルゴリズムが考案されているが、その成功率は人間の熟達者よりも低く、戦略はほぼ共通している [1]。

これらのアルゴリズムでは場にカードを置く際にそれをどの台に出すかを決定してしまい、その後予定の変更は行なわない。場札の予定を決定すると台への出し方が制限されるのでそれが失敗の原因となる場合もあり、場札の予定の決定は非常に重要である。

## 3.2 人間の Calculation の解法

Calculation には明確な解法がないので、解法についての知識は経験的なものにならざるをえない。Calculation を解くには、台および場の状態を認識し、次に来る手札の予測を行ない、カードの置き方を推論する必要がある。ゲームの最終目標は台にすべての札を並べることだが、その過程において場を有効に利用しなければならない。場に置く際には、

- なるべく台札の逆順に並べる必要がある
- 場札が多くなるほど並びに矛盾が生じやすく、ゲームの進行に不利である
- 置く場所によっては他の場札の台へ出せる場所を制限する

といったことを考慮する必要がある。また、実際に人間がゲームをする場合は、一度場札の予定を決めても状況によって適宜に変更するのが普通で、予定は必ずしも決定的なものではない。以上のような点を考えると、有効な方略を獲得するには、プレイを繰り返しながら学習するしか方法がないといえる。

#### 4 学習機能を持ったモデル

プロダクションシステムにおける学習の方法には、

- ルールの中のパラメータを調整する
- ルールの条件部が結論部を修正する
- ルールを新しく生成する

などが考えられる。

上で述べたように、Calculationのように一定の解法で解くことができない問題に対しては、あらかじめ全ての手続きを記述するのではなく、学習により新たな手続きを獲得するようにしなければならない。つまり、新しいルールを生成するプロダクションシステムを構成する必要がある。Calculationの場合システムが学習しなければならないことは、手札を台と場のどちらに出すかの判断や場札の並べ方の戦略である。

システムの動作は、「行動による学習」の理論に基づいて設計した。まず、初期問題空間として与えられた知識を用いてゲームを行ない、その過程で良くない例の収集とそれを回避する手続きの獲得を行なう。これらによって得られた知識をそれぞれプロダクションルールとしてルールベースに蓄える。また、ゲーム中は台札を並べる、または場札を台札の逆順に並べるという目標から場札の並べ方に関する下位目標を生成し、それに近づくようにゲームを進める。以下、それぞれのより詳しい説明である。

##### 1. 初期問題空間の構成

最初に持っている知識は、手札をめくること、手札を場に置くこと、手札を台に置ける条件である。これらの知識を用いてまずゲームを行なう。

##### 2. 良くない例の収集

Calculationは場札の列の並びが台札の列の並びの逆順にならないときに失敗する。これは一つ

の場札の列で起こることもあれば、複数の場札の列に跨って起こることもある。その原因は場札の予定が間違っていた場合と絶対に成功しない手札の並びだった場合があると考えられる。場札を台に置けなくなる過程において、手札として次に来ると失敗に結び付くカードが増えてくる傾向がある。これらの失敗した場面と失敗する可能性のある手札が増える場面を一旦作業記憶に記憶する。

##### 3. 良くない例を回避する手続きの獲得

良くない状態に遭遇した時に、その前の状態から良くないと判断された状態以外への状態の変化を導く手の出し方があれば、良くないと判断された状態への変化が良くなかったと推論される。このとき、それでゲームが失敗したならば、すぐに後戻りしてその原因を特定し、そのときの状況を条件部とした良くないと判断された状態への変化を起こさないようなルールをルールベースに付け加える。そうでない場合はその状況を保留してゲームを続行する。途中でゲームが行き詰まったら、後戻りして保留した状況を失敗の原因の候補として検討する。

##### 4. 良い例の収集と下位目標生成手続きの構成

システムは台札が増える、または場札が台札の逆順に並ぶことを目標にする。この目標に近付くとシステムはそのときの状態を良い状態として、この状態を下位目標として設定する。

このようなシステムにおいて、新しいルールを生成するための具体的な方法としては、あらかじめ着目すべき条件や行為のパターンを用意しておき、探索により適切な組み合わせを見出すようにすることが考えられる。

#### 5 おわりに

ここでは、カードゲーム“Calculation”の特徴を挙げ、その問題の分析を行ない、行動による学習の理論を基礎にしたCalculation解決の推論と学習のモデルを提案した。

今後は今回提案したモデルを計算機上に実現し、モデルの正当性、有効性の検証を行ないたい。

#### 参考文献

- [1] 石畑 清：カルキュレーションのアルゴリズム，*Computer Today No.57*，サイエンス社，pp.40-44，(1993)。