

## 3L-2 RLSのメタ知識生成・実行

田島 守彦 (電子技術総合研究所)

### 1. はじめに

現在、手続き的知識一般の学習を指向する学習システムRLS (Recursive Learning System)をESPで開発中である<sup>1)</sup>。

モデルについてまとめておく。①知識はシステムが仮定するいくつかの型のいずれかに属する知識片から階層的に合成される。型には、基本型、逐次型、条件型、各種構造型 (OR, ORDER, TOTAL, et c.)がある。②メタ知識生成器(MKG)が知識片の属する型を頼りに、学習用のメタ知識の初期版を生成する。③メタ知識が対象知識を改善する。

このたび、RLSで走るゲームプログラム(RLS-0thello)のゲームプレイの中間結果から、失敗を自覚させ、必要なメタ知識を多数生成・実行させ、失敗の責任付与を実行をさかのぼって行わせる基本的な動作に成功したので報告する。

RLSの学習は対象知識を合成している知識片のうちのあるものに責任を付与しそれを修正することで行われる。各知識片には可変度と称する可変性の度合を示す値がつけられており<sup>2)</sup>、これが責任付与時などの制御に使用される。

0:責任を付与しない／できない。

1:責任を付与できる。ただしその片自体を変更はしない。更に下部の片を修正することが有り得る。

2:責任を付与できる。またその片自体を修正できる。

### 2. 各型に対応する基本的メタ知識

学習用のメタ知識片には固定的なもの(クラスオブジェクトとし、#で示す)と可変なもの

(インスタンスとし\$で示す。通し番号を付ける)。後者は必要時にその初期版が生成される。一般にはメタ知識もやはり知識片の合成からなり、それらの学習用メタ知識(即ちメタメタ知識)も同様に生成される。メタ知識の初期版として何が生成されるかは対象知識片の型によって決まる。本稿に關係するのは次の型の対象知識のみである。対応／生成されるメタ知識を右に記す。

- (1) 基本型: 対するメタ知識は生成されない。基本型の知識片が修正されることはないからである。
- (2) 逐次型: 失敗が判明した最終段から始めて順次前へ責任を付与して行くような知識\$blame\_ass\_s. #find\_startと\$go\_upの2片から成る。後者はOR型の知識片である。
- (3) 条件型: 実行済みの行為あるいは条件に責任を付与するOR型の知識\$blame\_ass\_c.
- (4) OR型: 次の代替案を選ぶ、それ自身は基本型のメタ知識#blame\_ass\_or.
- (5) ORDER型: 第一段階として、次の代替案を選ぶ、それ自身は基本型のメタ知識#blame\_ass\_order.

### 3. 基本的な学習の例

図1の局面からのプレイを考える。今計算機が白、相手が黒である。RLS-0thelloは「常に最大の数の石を取る」という、実は間違った戦略でプレイする。手の系列は次のようになった。

26: 白b8 27: 黒c8 28: 白a7 29: 黒e8  
30: 白a2 31: 黒a1 32: 白pass 33: 黒b3  
RLS-0thelloはプレイ時に多数の知識片を生成・実行している(\$np3, 12, 15, 22, 25, 32, 35, 42, \$oth\_ply4, 13, 16, 23, 26, 33, 36, 43, \$best\_moves6, 18, 28, 38など)。さて33手終了の時点で黒が過半数になり、

RLSは失敗を自覚し学習モードに入った。

図2はRLSの実行監視用のウィンドウに出力された実行過程である。各文について説明する。知識片の階層における位置を深い／浅いで示している。

①黒b3により失敗を自覚。\$np42の学習を始める。

②深い位置の対象知識片\$best\_moves38を修正しようとする。order型なので#fix\_orderによる。代替案なし。

③浅い所に戻り、\$oth\_ply36(C型)の修正を試みる。このメタ知識が\$blame\_ass\_c(OR型)。失敗。このメタ知識の学習を始める。が、④失敗。

⑤\$np35(\$np42より1段深い\$np)の学習。これに対するメタ知識\$blame\_ass\_s45を実行。失敗。メタ知識の学習を始める。

⑥\$blame\_ass\_s45の2番目の部品\$go\_up46(OR型)を#fix\_orで修正。\$go\_up46中の次の代替案が採用される。

⑦即ち#blame\_ass\_s45の学習は成功。対象知識のレベルでは視点が一手さかのぼり、責任が\$np32に移動する。

⑧②と同様。深い所の対象知識片\$best\_moves28を修正する。以下同様

.....

⑳26手目の白b8を打着した\$best\_move6を修正。代替案を見つける。成功。

㉑即ち最上位の片\$np42の学習が成功。

#### 4. RLSの管理部

以上述べた動作を行わせるためにRLSのメタ知識生成部および実行部の開発においては次のような点に留意した。

(1)メタ知識生成において、修正が不要な知識片を出来るだけ予測し、無駄なメタ知識の生成を防ぐ。

(2)メタ知識実行において、実行の失敗は対象知識への責任付与／修正の失敗である。メタ知識の修正を試みるが、簡単な場合にはPrologのバックトラックに任せる。

#### 5. 結び

RLS-0thelloのプレイから、失敗を自覚し、メタ知識およびメタメタ知識を生成・実行

し、手の系列をさかのぼって失敗の責任を知識片に付与し、修正するような動作をRLSに行わせる事が出来た。本例ではメタ知識21片(うちメタメタ知識片が3片)が生成された。RLSの動作には汎用性がある。現在、より高度な推論による学習を実装中である。 | 本研究はICOTとの共同研究です。ICOTの関係各位に感謝します。

#### 文献

- 1)田島、学習を指向する手続き的表現とシステムの構想、「人工知能システムの枠組み」シンポジウム(1987)。
- 2)田島、学習システムRLSの責任付与、39回情報処全大4C-3(1989)。

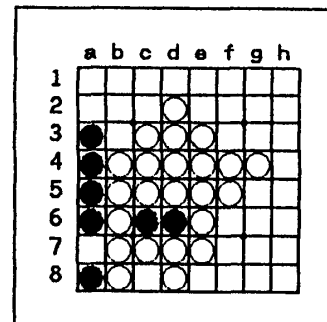


図1

```

output board
① L of $np42
② F of $best_moves38 by #fix_order
③ L of $blame_ass_c47
④ fail L of $blame_ass_c47
⑤ L of $blame_ass_s45
⑥ F of $go_up46 by #fix_or
⑦ success L of $blame_ass_s45
⑧ F of $best_moves28 by #fix_order
   L of $blame_ass_c53
   fail L of $blame_ass_c53
   L of $blame_ass_s51
   F of $go_up52 by #fix_or
   success L of $blame_ass_s51
   F of $best_moves18 by #fix_order
   L of $blame_ass_c59
   fail L of $blame_ass_c59
   L of $blame_ass_s57
   F of $go_up58 by #fix_or
   success L of $blame_ass_s57
⑳ F of $best_moves6 by #fix_order
㉑ success L of $np42
  
```

図2 実行過程