論 文

Technical Papers

強化学習結果の再構築への概念学習の適用

Using Concept Learning for Restructuring Control Policy in Reinforcement Learning

松井 藤五郎 Tohgoroh Matsui
名古屋工業大学大学院工学研究科
Graduate School of Engineering, Nagoya Institute of Technology
tohgoroh@ics.nitech.ac.jp, http://www-sekilab.ics.nitech.ac.jp/~tohgoroh/

犬塚 信博 Nobuhiro Inuzuka
名古屋工業大学共同研究センター
Center for Cooperative Research, Nagoya Institute of Technology
inuzuka@ics.nitech.ac.jp, http://www-inzklab.ics.nitech.ac.jp/

世木 博久 Hirohisa Seki
名古屋工業大学知能情報システム学科
Department of Intelligence and Computer Science, Nagoya Institute of Technology
seki@ics.nitech.ac.jp, http://www-sekilab.ics.nitech.ac.jp/

伊藤 英則 Hidenori Itoh
（同　上）
itoh@ics.nitech.ac.jp, http://www-itolab.ics.nitech.ac.jp/

keywords: reinforcement learning, concept learning, inductive logic programming, autonomous agents

Summary

Conventional reinforcement learning has focused on learning in a stable environment. However, an agent may be given another environment which differs from the old environment. Thus, an autonomous agent needs a method to learn efficiently a new policy suited for the new environment.

In this paper, we propose a method to adapt to a new environment for an agent which has a task to reach goals. When an agent is provided with a new environment, our method learns a new partial policy using the precondition of agent's old policy. The precondition of a policy is a condition that says what must be satisfied in order to reach goals by using the policy. Similarly to learning the precondition of an action from the instances of action's success or failure by using concept learning, our method learns the precondition of a policy from the instances of policy's success or failure by using concept learning. We describe a method using inductive logic programming (ILP) as a concept learning method. Since ILP provides methods for learning relational knowledge that is not expressible in attribute-value learning, our method can use relational representation for the precondition. We applied our method to a blocks-world problem for evaluation. We have come to conclusion that our method is effective when the cost to carry out the task is high.

1. はじめに

従来の強化学習の研究は，エージェント周囲の環境が不変であるものと仮定して，最適な政策（policy）——状態観測から行動出力へのマッピング——を学習する．しかし，実際には，学習された政策が利用される環境は，その政策を学習した時の環境とは必ずしも同一ではない．マルチエージェントシステムの代表的な問題であるサッカーゲームの例を考えてみよう．エージェントは，ふつう，政策が利用される環境とできる限り同じ環境で学習を行う．ところが，実際に対戦するチームは学習時に対戦したチームとは異なるので，学習した政策が常に最適であるとは限らない．それどころか，その政策では目標に到達できないかもしれない．したがって，新しい環境を与えられたエージェントは，その環境に合った新しい政策を再び学習しなければならない．しかし，政策をはじめから学習しなおした場合には，それまでの政策を学習するのに匹敵する経済的・時間的コストが必要になる．そこで，はじめから学習しなおすのではなく，それまでの政策を新しい環境に合うように効率的に再構築する手法が必要になる．本論文の目的は，目標到達型のタスクを対象としたモデルなし・経験強化型の強化学習において，新しい環境に効率よく適応する——すなわち，少ないエピソード数で目標到達率を回復させ経済的・時間的コストを抑制する——ための基本的アルゴリズムを示し，その有効性を検討することである．

Kato らは，忘却の概念を導入した強化学習の手法・forgettable profit sharing を提案し，解析的な議論を行っている [Kato 00]．仮想キューを用いた forgettable profit

sharing では，ルール重みの更新時にすべての重みに忘却率をかけることによって無効ルールを抑制し，環境の変化に適応する．また，港らは，政策の学習後に環境が変化した場合にも，最適性を犠牲にしてタスク達成だけを考えれば，それまでの行動政策が新たな環境で部分的に適用できると考え，過去の環境で獲得した政策を部分的に修正する手法を提案している [港 00]．それまでの政策のうち新たな環境において不都合が生じる部分のみを学習しなおすことによって，新しい政策を学習する時間が短縮される——としている．この手法では，(a) 負の報酬を獲得した状態から $n_r = 1$ ステップ前までの状態および状態空間上でそれらと隣接する状態の集合を学習しなおす範囲として部分的な強化学習を行い，(b) 十分なタスク達成率が得られない場合には，n_r の値をひとつ増やして再び部分的な強化学習を行う．このようにして，十分なタスク達成率が得られるまで (b) を繰り返すことによって再学習の範囲を探索している．

本論文では，

(1) 再学習の範囲を判別するために政策事前条件を定義し，これを用いて政策を再構築する手法，

(2) 政策事前条件を概念学習によって獲得する手法

を提案する．概念学習アルゴリズムは，タスクの性質や事例の表現に合わせて選択する．本論文では，事例を一階述語論理で表現し，概念学習に帰納論理プログラミング（inductive logic programming, ILP）を用いた手法について述べる．ILP の観点から見ると，自律エージェントへの応用は ILP 研究におけるトピックのひとつである [Benson 96, Jacobs 98, Džeroski 98, Reid 00]．

以下，2 章では政策の事前条件について説明する．また，政策事前条件を用いて政策を再構築する方法について述べる．3 章では政策事前条件を概念学習によって獲得する方法を積み木の問題を用いて説明する．その後，実験による評価の結果を 4 章に示し，5 章で (1) 政策事前条件を用いた政策再構築，(2) 概念学習による政策事前条件獲得，それぞれの有効性について考察する．6 章では関連研究を紹介し，最後に，結論を 7 章で述べる．

2. 政策事前条件を用いた政策の再構築

2·1 問題設定と強化学習

次のようなタスクを持ったエージェントを考える．エージェントには，

- とりうる状態の集合 Q
- とりうる行為の集合 A
- 関数 $pre : Q \times A \to \{\mathbf{T}, \mathbf{F}\}$
- 目標 $goal : Q \to \{\mathbf{T}, \mathbf{F}\}$
- 初期状態 $q \in Q$

が与えられる（$\mathbf{T}$, $\mathbf{F}$ はそれぞれ真，偽を表す）．ここで，関数 pre は行為の事前条件（precondition）であり，$pre(q,a) = \mathbf{T}$ ならば，状態 q において行為 a が実行可能である——という．エージェントにとって未知の状態遷移関数 $\delta : Q \times A \to Q$ が定義されており，q において a を実行したとき，エージェントは $\delta(q,a)$ に置かれる．初期状態は，タスクごとに環境から与えられ，エージェントが選択することはできないものとする．エージェントの目的は，

(1) $goal(\delta(\cdots\delta(q,a_1)\cdots,a_n)) = \mathbf{T}$,

(2) $pre(\delta(\cdots\delta(q,a_1)\cdots,a_{i-1}),a_i) = \mathbf{T}$ $(i = 1,\ldots,n)$

を満たす一連の行為 $a_1,\ldots,a_n$ $(a_i \in A)$ を見つけることである．

強化学習は，時刻 t における報酬[*1] $r_t = r(q_t,a_t)$ を環境から受け取り，利得——最も単純な場合，報酬の総計——を最大にするような政策を学習する．（決定的な）**政策**（policy）とは，任意の状態 $q \in Q$ からある行為 $a \in A$ へのマッピング $\pi : Q \to A$ のことをいい，時刻 t の状態 q_t において次に行うべき行為 a_t は $a_t = \pi(q_t)$ として求められる．報酬 r_t は，本研究で扱う問題のように目標状態への経路を学習させる場合には，$goal(s_t) = \mathbf{F}$ かつ $goal(\delta(s_t,a_t)) = \mathbf{T}$ のとき $r_t(s_t,a_t) = 1$，そうでないとき $r_t(s_t,a_t) = -1$ とすることがよく行われる [Sutton 98]．状態遷移関数 δ と報酬関数 r の組は環境モデルと呼ばれ，モデルがエージェントにとって未知である場合には，モデルなしの強化学習アルゴリズムが用いられる．本研究では，モデルなし・経験強化型の強化学習法として無効ルール抑制定理を利用した profit sharing [宮崎 94] を用いている．

2·2 政策の失敗と政策事前条件

次に，政策 π を学習した後，学習時とは異なる環境が与えられた場合について考える．このような場合には，π に従って行動しても目標状態に到達できないことがある．しかし，最適性を犠牲にしてタスク達成だけを考えれば，π は新たな環境で部分的に適用できる [港 00]．本論文では，それまでの行動政策を新たな環境にも適用できる条件を政策の事前条件としてとらえる．

π を学習した時の環境を

$$e = \langle Q, A, \delta, pre, goal \rangle,$$

π を利用する時の環境を

$$e' = \langle Q, A, \delta', pre', goal' \rangle$$

とする．環境 e' において，状態 $q \in Q$ から政策 π に従って行動しても目標に到達できないとき，π は q で**失敗**する——という．すなわち，π を用いて

(1) $goal'(\delta'(\cdots\delta'(q,a_1)\cdots,a_n)) = \mathbf{T}$,

(2) $pre'(\delta'(\cdots\delta'(q,a_1)\cdots,a_{i-1}),a_i) = \mathbf{T}$ $(i = 1,\ldots,n)$

*1 ふつう，エージェントにとって報酬関数 r は未知である．

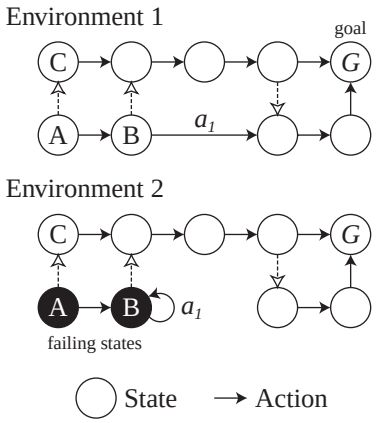

図 1 異なる環境の例．実線の矢印は，上の環境における最適政策 π を表している．環境が下の環境に変化すると，状態 B において以前最適だった行動 a_1 を行っても目標に到達できない．黒丸は，政策 π が失敗する状態を表している．

を満たす一連の行為 $a_1, \ldots, a_n$ $(a_i \in A)$ を見つけることができないとき，π は q で失敗する．

政策 π が失敗しないために状態 q が満たしていなければならない条件を π の**事前条件**とし，pre^{π}_{policy} と書く．すなわち，

$$pre^{\pi}_{policy}(q) = \mathbf{T}$$

ならば，そのときに限り，π に従って行動することによって q から目標に到達することができる．

このことを，状態遷移関数が異なるふたつの環境の例（図 1）を用いて具体的に説明する．初期状態はすべての状態の中からランダムに選択される．まず，エージェントが環境 1 で政策 π（実線の矢印）を学習したとする．このエージェントが環境 2 を与えられたとき π に従って行動すると，状態遷移が環境 1 と異なるため黒丸の状態からは目標に到達できない．つまり，π は環境 2 の黒丸の状態 q で失敗，すなわち，

$$pre^{\pi}_{policy}(q) = \mathbf{F}$$

である．また，それ以外の状態 q に対して

$$pre^{\pi}_{policy}(q) = \mathbf{T}$$

が成り立つ．

政策 π の事前条件 pre^{π}_{policy} と区別するため，以下では行為の事前条件を pre_{action} と書く．

2·3 政策の再構築

本節では，前節で述べた政策事前条件を用いて，環境変化後の環境 e' に合わせて政策を再構築する手法を提案する．e' の状態 $q \in Q$ において政策 π の事前条件を満たしているとき，π に従って行動すれば目標に到達できる．したがって，学習しなければならないのは，π の事前条件を満たさない状態でどの行為を選べばいいか——ということである．これを政策 π' とすると，π' の定義域は，π の事前条件を満たさない状態の集合

$$Q' = \left\{ q \in Q \;\middle|\; pre^{\pi}_{policy}(q) = \mathbf{F} \right\}$$

となる．

そこで，e' において π とその事前条件が与えられたとき，π の事前条件を満たす状態では π に従って行動し，そうでないときは新しい政策 π' を学習する．すなわち，

$$pre^{\pi}_{policy}(q) = \mathbf{T}$$

となる状態 q では行為 $a = \pi(q)$ を選択し，そうでないときは強化学習を用いて $\pi' : Q' \to A$ を学習する．このアルゴリズムを図 2 に示す．目標に到達できない状態だけで学習することによって学習の収束に要するエピソード数を抑制する．

RESTRUCTURE_POLICY$(\pi, pre^{\pi}_{policy})$
入力：政策 π，政策 π の事前条件 pre^{π}_{policy}
(1) 政策 π' を初期化する
(2) $path$ をスタックとする
(3) **do foever**
(4)　状態を初期化し，状態 q を観測する
(5)　$path \leftarrow \phi$，報酬 $reward \leftarrow 0$
(6)　**while** not $goal(q)$ **do**
(7)　　**if** $pre^{\pi}_{policy}(q)$
(8)　　**then** $a \leftarrow \pi(q)$
(9)　　**else**
(10)　　π' に従って行為 a を選択する
(11)　　$path$ に (q, a) をプッシュ
(12)　　$reward \leftarrow reward + r(q, a)$
(13)　　行為 a を実行する
(14)　　新しい状態 q を観測する
(15)　強化学習法に従い，$reward$ と $path$ に基づいて π' を更新する

図 2 政策事前条件を用いた政策再構築アルゴリズム．

3. 概念学習による政策事前条件の獲得

前章で述べた再構築手法には政策事前条件が必要である．では，政策 π の事前条件はどのようにして求められるのだろうか．Q が有限で，新しい環境 e' の任意の状態 $q \in Q$ を選んでタスクを始めることができるならば，π の事前条件は，すべての $q \in Q$ から政策 π を用いて行動し目標状態に到達できるかを調べることによって獲得できる．しかしそうでない場合[*2]には，実際に経験した状態の政策事前条件の値しか知ることはできない．

Shen は，行為モデルにおける行為の事前条件 pre_{action} が行為の成功例・失敗例から概念学習を用いて求められ

*2 対戦相手が存在するゲームなどのマルチエージェント問題においては，任意の状態を生成することは困難である．

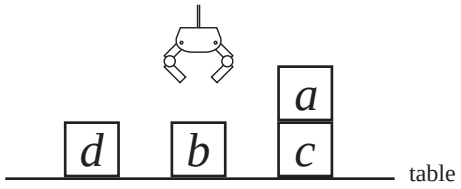

図 3 積み木の問題.

ることを示している [Shen 93]．そこで，本手法では，同様にして政策の事前条件 pre^{π}_{policy} を政策 π の成功例・失敗例から概念学習を用いて帰納的に獲得する．概念学習によって獲得した事前条件は，収集した事例だけでなく未知の事例も分類できることから，収集した事例を一般化したものとなる．

本手法は，(1) それまでの政策に従って行動し，その成功例・失敗例を収集する．(2) 概念学習を行うのに十分な事例が集まると，政策の失敗を表す概念を獲得する．獲得した概念に含まれる状態では，政策事前条件が満たされていないと考える．エージェントは十分な数の事例が集まると環境が変化したものと認識し，事例が集まらない場合には政策の再構築を行わない．

決定木学習や人工ニューラルネットワークなど様々な概念学習アルゴリズムが提案・開発されている．本手法で用いる概念学習アルゴリズムは，問題や事例の表現にあわせて適切に選択する．本章では，一階述語論理を用いて事例を表現し，概念学習に ILP を用いる手法について述べる．一階述語論理は，その知識表現および推論が人工知能の分野では古くから研究されており，知識ベースエージェントを表現する言語として用いられている．ILP では，与えられた正事例の集合 E^+，負事例の集合 E^-，背景知識 B から，

任意の $e \in E^+$ に対して $B \cup H \models e$,
任意の $e \in E^-$ に対して $B \cup H \not\models e$

を満たす仮説 H を見つける．B, H は一階述語論理に基づいたホーン節の集合である．このため，ILP は，属性-値学習など他の手法では学習できない関係表現を含んだ概念を学習でき，問題の構造的な面をとらえることが可能となる．また，背景知識を利用することによって複雑な概念の学習ができる．$B \cup H \models e$ となる事例の集合 $E_{B \cup H}$ は，$E_{B \cup H} \supseteq E^+$ となり，ILP によって E^+ が一般化される．

3·1 例題：積み木の問題

本節では，積み木の問題（図 3）を説明する．

いま，a, b, c, d の 4 つの積み木が机の上にある．ロボットは，$stack(x, y)$ という行為によって積み木 x を積み木 y の上に積むか，$putdown(x)$ という行為によって積み木 x を机の上に置くことができる．ロボットのタスクは，与えられた初期状態から作業を始めて目標状態を達成することであり，この問題の目的はタスクを達成するための政策をロボットに学習させることである．目標状態は，積み木を d, c, b, a の順に積んだ状態とする．

$$goal(s) = \begin{cases} \mathbf{T} & \left(\begin{array}{l} ontable(d) \wedge on(c,d) \\ \wedge on(b,c) \wedge on(a,b) \end{array} \right) \\ \mathbf{F} & (\text{それ以外のとき}) \end{cases}$$

ロボットが政策を学習した後，他の積み木の上に置かれている積み木は床の上にしか移動できない——というように環境が変化する．環境変化後，図 3 の状態で積み木 a を積み木 b または積み木 d の上に積むことはできない．

このような場合には，前の環境で学習した政策をそのまま用いることができないため，新しい環境に合わせて政策を再構築する必要がある．

LEARN_CONCEPT_OF_FAILURE($\pi, maxStep, B$)
入力： 政策 π，失敗基準 $maxStep$，背景知識 B.
出力： π の失敗を表す概念.
(1) $t \leftarrow 0$, $E^+ \leftarrow \phi$, $E^- \leftarrow \phi$
(2) // 事例の収集
(3) **do until** E^+, E^- が十分なサイズ
(4) 状態を初期化し，状態 q_t を観測する
(5) $i \leftarrow 0$, $E \leftarrow \{e_t\}$
(6) **while** not $goal(q_t)$ かつ $i < maxStep$ **do**
(7) $a_t \leftarrow \pi(q_t)$
(8) 行為 a_t を実行する
(9) 新しい状態 q_{t+1} を観測する
(10) $E \leftarrow E \cup \{e_{t+1}\}$
(11) $t \leftarrow t+1$, $i \leftarrow i+1$
(12) **if** $goal(q_t)$
(13) **then** $E^- \leftarrow E^- \cup E$
(14) **else** $E^+ \leftarrow E^+ \cup E$
(15) // 失敗を表す概念の学習
(16) $B' \leftarrow B \cup \dot{q}_0 \cup \dot{a}_0 \cup \ldots \cup \dot{q}_{t-1} \cup \dot{a}_{t-1} \cup \dot{q}_t$
(17) ILP を用いて以下を満たす仮説 H を求める
任意の $e_j \in E^+$ に対して $B' \cup H \models e_j$
任意の $e_j \in E^-$ に対して $B' \cup H \not\models e_j$
(18) **return** $B' \cup H$

図 4 ILP を用いて政策の失敗を表す概念を学習するアルゴリズム

3·2 アルゴリズム

本節では，一階述語論理を用いて事例を表現し，ILP を用いて政策の失敗を表す概念を学習するアルゴリズム（図 4）について述べる．このアルゴリズムは，与えられた政策 π に従って行動し概念学習のための事例を収集する部分（2–14 行目）と，集めた事例を ILP によって一般化し π の失敗を表す概念を獲得する部分（15–17 行目）から成る．

事例を収集する間，それまでの政策 π が失敗したことを表す概念の事例として

$e_t =$ `fail(s)`.

```
q̇t = { ontable(d,s),
       clear(d,s),
       ontable(b,s),
       clear(b,s),
       ontable(c,s),
       on(a,c,s),
       clear(a,s) }
ȧt = { stack(b,a,s) }
```

図 5 図 3 において $stack(b,a)$ を実行したときの記述．s は状況変数．

を生成する（10 行目）．ここで，s は時刻ごと生成されるユニークな状況変数（situation variable） [McCarthy 69] である．目標へ到達できなかった場合には e_t を失敗概念の正事例とし，到達した場合には負事例とする（12 行目–14 行目）．これを扱うために，それまでの平均ステップ数などを参考にして失敗基準 $maxStep$ を定め，$maxStep$ ステップ以内に目標に到達できないとき，これを失敗とする（6 行目）．また，時刻 t において観測した状態 q_t とそのときの行為 a_t を，同じ状況変数 s を用いてホーン節集合 $\dot{q}_t$ および $\dot{a}_t$ としてそれぞれ記述する．たとえば，q_t が図 3 の状態で，a_t が $stack(b,a)$ のときの記述 $\dot{q}_t$, $\dot{a}_t$ は図 5 のようになる．状況変数を用いることにより，これらがあるひとつの状況 s についての記述である——ということを表している．

事例収集が終了した時刻 t までの状態と行為についての情報

$$I_t = \dot{q}_0 \cup \dot{a}_0 \cup \ldots \cup \dot{q}_{t-1} \cup \dot{a}_{t-1} \cup \dot{q}_t$$

と与えられた背景知識 B をあわせて $B' = B \cup I_t$ とする．B' はロボットの経験を含んだ背景知識であり，B' と正負の事例集合 E^+, E^- から

（1） 任意の $e_j \in E^+$ に対して $B' \cup H \models e_j$

（2） 任意の $e_j \in E^-$ に対して $B' \cup H \not\models e_j$

を満たす仮説 H を ILP を用いて求める（17 行目）．ここで，背景知識 B は状態を分類するのに役立つような知識である．背景知識の具体例を付録 A に示す．

仮説 H を学習すると，任意の状態 $q \in Q$——未経験の状態を含む——における政策事前条件 pre^{π}_{policy} の値を予測できるようになる．時刻 t' の状態 $q_{t'}$ における政策事前条件 pre^{π}_{policy} の値は次のようにして予測する．まず，$a_{t'} = \pi(q_{t'})$ を求め，

$$I_{t'} = \dot{q}_0 \cup \dot{a}_0 \cup \ldots \cup \dot{q}_{t'} \cup \dot{a}_{t'}$$

を記述し，B とあわせて $B' = B \cup I_{t'}$ とする．次に，ユニークな状況変数 s を生成する．すると，このときの事例は

$$e_{t'} = \texttt{fail(s).}$$

として表される．これらと学習した仮説 H から，政策事前条件の予測値 $\widehat{pre}^{\pi}_{policy}$ を求める．

$$\widehat{pre}^{\pi}_{policy}(q_{t'}) = \begin{cases} \mathbf{F} & (B' \cup H \models e_{t'}) \\ \mathbf{T} & (\text{それ以外のとき}) \end{cases}$$

$\widehat{pre}^{\pi}_{policy}$ は事例収集時に経験した状態だけでなく未経験の状態も分類できるため，概念学習には経験を一般化する効果がある．ただし，この予測をするときには，事例発生時刻 t' 以後の情報——すなわち，将来の情報——を用いることはできない．したがって，与える背景知識 B は，事例発生時刻以後の情報を参照しないものにしておく必要がある．

3·3 ILP システム

前節で述べたアルゴリズムに従って ILP を行うために，ILP システム Progol [Muggleton 95] を改良した．Progol は逆伴意法（inverse entailment）とトップダウン探索に基づいたアルゴリズムである．Progol は，まず逆伴意法によって最も特殊な節——ボトム節と呼ぶ——を構成する．次に，ボトム節を包括する空間において近似 A*探索を行い，仮説を効率的にトップダウン探索する．

Progol や FOIL [Quinlan 90] など，ふつうの ILP システムでは，一般化されなかった事例はそのまま仮説論理プログラムに加えられる．たとえば，図 5 の記述が行われたときの事例

```
fail(s).
```

が正事例であり一般化されないとき，この事例はそのまま仮説に加えられる．しかし，状況変数はそれぞれの時刻ごとに異なる値をとるため，この節と別の時刻の事例がマッチすることはない．そこで，事例そのものではなく，事例 e_t の記述 $\dot{q}_t \cup \dot{a}_t$ を前件とした節から状況変数だけを一般化したものを仮説論理プログラムに加えることとする．この例の場合，

```
fail(X):-
  ontable(d,X),clear(d,X),ontable(b,X),
  clear(b,X),ontable(c,X),on(a,c,X),
  clear(a,X),stack(b,a,X).
```

が該当する．これにより，状況変数が異なるだけで同じ状態を表している事例とマッチするようになる．

4. 実　　験

本手法の効果を確認するため，3·1 節で述べた積み木の問題[*3]を用いて実験を行った．モデルなし・経験強化型の強化学習には profit sharing を用いた．各ルールの初期値は 0.1，行動選択は重みに比例した確率に基づくルーレット選択を用いた．報酬は，目標に到達するまで

*3 $|Q| = 73$.

は −1，目標に到達したときに 10 を与えることとし，目標に到達したときにエピソードに含まれるルールの強化を行った．初期状態はすべての状態から一様分布に従ってランダムに選択され，獲得報酬の期待値が 0 となる 10 ステップ目で目標に到達できないとき新しい初期状態へのリセットを行った．強化関数は，[宮崎 94] の無効ルール抑制定理を満たすよう公比 1/13 の等比減少関数

$$f_i = \frac{1}{13} f_{i-1} \quad (i = 1, \cdots, 10)$$

とした．政策事前条件を学習する際のパラメータ $maxStep$ は 10，収集事例数は正事例・負事例とも 100 とした．背景知識は，付録 A に示したものを与えた．

実験は，100,000 エピソード経過後に環境を変化させ，最大の重みをもつルールを選択する決定的行動を行ったときの目標到達率と平均獲得報酬[*4]の変化を計測した．これを 10 回ずつ行い，その平均を求めた．比較として，仮想キューを用いた忘却率 0.9999 の forgettable profit sharing [Kato 00]，政策を初期化して再学習した場合（re-learning），本手法において一般化を行わない場合（non-generalizing）の実験を行った．一般化を行わない実験では，事例収集時に政策事前条件を満たさない状態であると確認され，正事例として収集された状態だけで政策を学習しなおした．

結果を図 6・図 7 に示す．横軸は，環境が変化した時点を 0 としている．本手法（our method）と一般化を行わない手法では，初期の試行において事例の収集が行われている．本手法は，初めから学習しなおした場合とほぼ等しい目標到達率・獲得報酬を持つ政策を学習し，初めから学習しなおした場合よりも早く学習が収束した．また，本手法は，一般化を行わない場合よりも高い目標到達率・獲得報酬を持つ政策を学習した．

本システムが学習した仮説プログラム H を図 8 に示す．また，従来の Progol を用いて同じ事例，同じ背景知識から学習した結果を図 9 に示す．Progol では一般化されなかった事例が，本システムではその状況の記述を前件とする節から状況変数のみを一般化した節に一般化されたことが確認できる．

次に，本手法において収集する事例の数（図 4 の 3 行目）を小さくした場合について，同様の実験を行った．その結果を図 10・図 11 に示す．収集する事例数が少ないとき，再構築した政策の目標到達率が低くなった．これは，本手法で政策事前条件の獲得に用いている帰納的学習では事例数が少ないと正しい仮説を学習することができないためである．

これを確認するため，本手法における概念学習の精度を調べた．本手法ではそれまでの政策 π が失敗すると判断されたときは新しい政策を学習しなおすが，ここでは，失敗と判断されるときにも π に従って行動することによってその結果を確認し，精度を求めた．結果を表 1 に示す．表中の $|seqs|$ は事例の収集に要したエピソード数，$|Q_{E^+}|$ は収集した正事例に含まれる状態，すなわち，政策事前条件を満たしていないことを確認した状態の数，$|Q_H|$ は

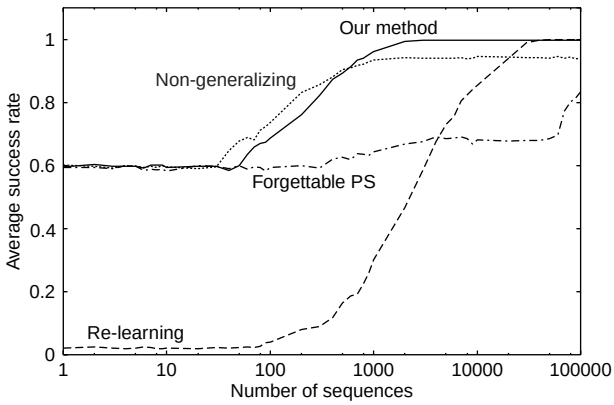

図 6 環境変化後の平均目標到達率．

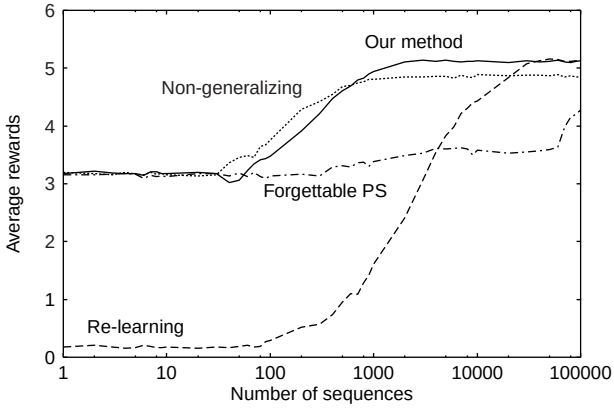

図 7 環境変化後の平均獲得報酬．

```
fail(A) :- on(B,b,A), on(c,C,A).
fail(A) :- on(B,a,A), ontable(a,A),
  ontable(d,A).
fail(A) :- on(b,B,A), ontable(B,A),
  ontable(c,A), clear(c,A).
fail(A) :- on(b,a,A), on(d,c,A),
  ontable(a,A), ontable(c,A),
  clear(b,A), clear(d,A), putdown(d,A).
```

図 8 学習した仮説プログラムの例（収集事例数が 100 のとき）．

```
fail(s80).
fail(A) :- on(B,b,A), on(c,C,A).
fail(A) :- on(B,a,A), ontable(a,A),
  ontable(d,A).
fail(A) :- on(b,B,A), ontable(B,A),
  ontable(c,A), clear(c,A).
```

図 9 図 8 の学習時と同じ入力から Progol で学習した結果．最初の節 `fail(s80).` は他のいかなる事例ともマッチしない．

*4 各時点での政策に従い，1,000 回のタスクを行って計測した．

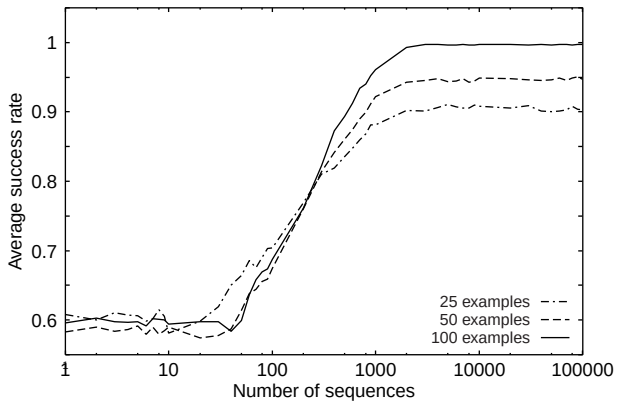

図 10 事例数を変えた場合の平均目標到達率.

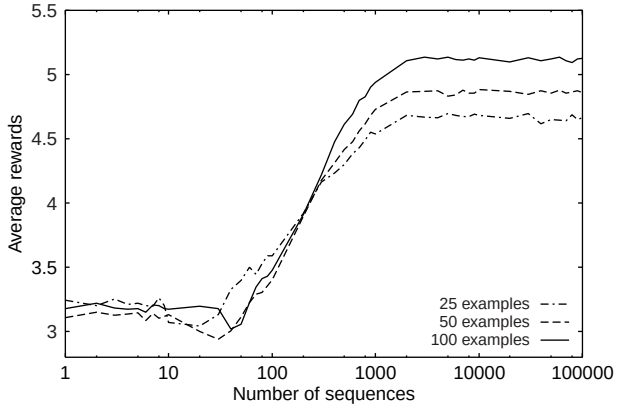

図 11 事例数を変えた場合の平均獲得報酬.

本システムが学習した仮説が政策事前条件を満たしていない状態に分類した状態の数を表す．事例を多く集めるにはより多くのエピソードを要したが，より多くの状態を経験しより精度の高い仮説を学習した．

5. 考　　察

本手法を (1) 政策事前条件を用いた政策再構築，(2) 概念学習による政策事前条件の獲得——という面から

(a) 学習した政策の性能（目標到達率・獲得報酬）
(b) 政策学習の収束に要するエピソード数
(c) 計算量

について考察する．

表 1 概念学習の平均精度．$|seqs|$ は事例の収集に要したエピソード数，$|Q_{E^+}|$ は収集した正事例に含まれる状態，すなわち，政策事前条件を満たしていないことを確認した状態の数，$|Q_H|$ は本システムが学習した仮説が政策事前条件を満たしていない状態に分類した状態の数を表す．

事例数	$\|seqs\|$	$\|Q_{E^+}\|$	$\|Q_H\|$	精度
25	9.4	3.0	27.7	0.707
50	19.4	3.8	24.6	0.796
100	39.6	5.9	29.2	0.858

5·1 政策事前条件を用いた政策再構築の有効性

まず，政策事前条件が与えられた場合について，2·3 節で述べた政策再構築の有効性を検証する．

(a) 部分的な強化学習によって，目標到達率は前の政策をそのまま使う場合よりも上昇する．ただし，政策事前条件は目標への到達だけに着目しているため獲得する報酬は最適でない可能性がある．
(b) 部分的な強化学習では対象となる状態の数が減るので，学習の収束が早くなると期待できる．実際に，実験では初めから強化学習をやり直した場合よりも早く収束した．また，港らは，移動ロボットの行動学習において実際に失敗を経験した状態だけで Q 学習を部分的に行うことによって学習が早く収束し，環境の変化に適応できたと報告している [港 00]．
(c) 行動選択時に政策事前条件が満たされているか調べるので，その分だけタスク実行時の計算量が増加する．実装では，調べた結果を記憶することによって調べる回数を各状態につき一度だけにした．政策更新の計算量は従来の強化学習に等しい．

以上から，政策事前条件を用いた政策再構築は，環境変化後に少ないエピソード数で目標到達率を回復させるのに有効であると考える．ただし，本手法では，[港 00] と同様に，強化学習中に環境が変化しないと仮定している．

5·2 概念学習による政策事前条件獲得の有効性

次に，政策事前条件の獲得に概念学習を用いることの有効性について考察する．

(a) 概念学習による一般化によってエージェントが経験していない状態も再学習の対象になる．これにより，再構築した政策の性能が良くなる [*5]．一方，分類誤りは再構築した政策の性能に影響し，これを次の 2 つに分けることができる．図 12 の例で示す．(図中の枠は分類誤りを生じる仮説を表し，枠内の状態を政策事前条件を満たしていない状態に分類する．)
- 学習した仮説が政策事前条件を満たしている状態を満たしていない状態に分類した場合（状態 C）．このとき，その状態では政策を学習しなおすことになるが，目標到達率への影響は小さい．
- 事前条件を満たしていない状態を満たしている状態に分類した場合（状態 B）．このとき，その状態ではそれまでの政策に従って行動してしまうので，その後目標に到達できなくなる可能性がある．特に，遷移した先の状態でも同じ誤りを続けると，目標に到達できない．

したがって，目標到達率を考えたとき後者の誤りは重大な誤りであり，再構築した政策の目標到達率の上限に影響を及ぼす．表 2 に重大誤り率と最終的な目標到達率の関係を示す．重大誤り率が小さいほど，

[*5] ただし，未知の状態がすべて政策事前条件を満たす状態に分類される場合の性能は，概念学習を行わない場合と変わらない．

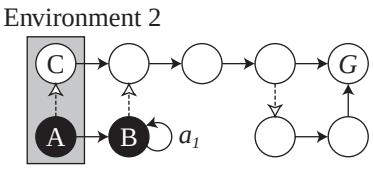

図 12 分類誤りを生じる仮説の例（図中の枠）．この仮説は，枠内の状態を政策事前条件を満たしていない状態に分類する．状態 B の分類は重大誤りとなる．

表 2 重大誤り率と最終目標到達率の関係．

事例数	重大誤り率	最終目標到達率
25	0.180	0.903
50	0.116	0.947
100	0.065	0.997

目標達成率は大きくなる．しかし，重大な誤りがあったとしても，政策事前条件を満たしている状態に遷移できれば，目標に到達することができる．

(b) 事例収集に要したエピソード数が強化学習を部分的に行うことによって減らされるエピソード数を超えてしまう場合には，学習収束までのエピソード数は学習しなおす場合よりも多くなってしまう．実験では，事例収集に要したエピソード数は強化学習の収束に要したエピソード数に比べて十分に小さいものであった．一般には，十分な精度を持つ仮説を学習するために必要な事例数やエピソード数を事前に求めることは難しい．

(c) 概念学習を一度実行する分だけ計算量が増加する．

以上から，概念学習を用いて政策事前条件を獲得することは，より性能の高い政策を再構築するのに有用であると考える．しかし，概念学習にかかる計算量が増加する，（多くの事例を収集する場合に）学習全体に要するエピソード数が初めから学習しなおすよりも大きくなる可能性がある——というデメリットがある．

6. 関 連 研 究

自律型ロボットの実用化のために，強化学習における環境の変化への適応が重要な問題となっている．港らは，[港 00] において，タスク達成率の低下によって環境の変化を認識し，政策を部分的に再学習する手法を提案している．この手法では実際に失敗した状態とその近傍の状態における行動しか改善されないが，本手法では，概念学習を用いることによって前の環境で学習した政策が失敗する状態を一般化しており，環境変化後に初めて経験する状態であっても失敗を未然に防げることがある．

また，Kato らは，[Kato 00] において，忘却の概念を導入した強化学習法・forgettable profit sharing を提案し，解析的な議論を行っている．仮想キューを用いた forgettable profit sharing では，ルール重みの更新時にすべての重みに忘却率をかけることによって無効ルールを抑制し，環境の変化に適応する．この手法は，強化学習中の環境の変化にも適応することができるが，ルールには環境変化後も前の環境と同じ報酬が分配されることを想定しており，本論文の実験のように，環境の変化によってエピソードが長くなり，分配される報酬が減少する場合には，適応するのに時間がかかる．

ILP を自律エージェントへ応用する研究という観点から見ると，Benson が TOP（Teleo OPerator） [Nilsson 94] で表現された行為モデルを ILP によって学習するエージェント TRAIL [Benson 96] を提案している．Ryan らは TOP を強化学習で学習する手法 RL-TOPs を提案し [Ryan 98]，Benson の TRAIL にこれを応用する研究 [Reid 00] を行っている．また，Džeroski らは [Džeroski 98] において ILP と Q 学習を組み合わせた手法を提案している．

7. ま と め

本論文では，モデルなし・経験強化型の強化学習において，エージェントが過去の環境で獲得した行動政策を新たな環境で適用するための条件——政策事前条件——を概念学習を用いて獲得し，政策を部分的に学習しなおすことによって環境の変化に適応する手法を提案した．また，一階述語論理を用いて政策事前条件の事例を表現するとともに概念学習に ILP を用いることによって一階述語論理に基づいた関係表現の利用を可能にし，積み木の問題を用いた実験によりその有効性を確認した．

政策事前条件を用いて政策再構築を行うと，新しい環境での学習収束に必要なエピソード数——タスクの実行回数——が学習しなおす場合と比較して少なくなると見込まれる．これに加え，概念学習自体はタスクを実行することなしに行えることから，本手法はタスクの経験に要する時間的・経済的コストが大きい問題に対して有用であると考える．しかしながら，本手法を用いる場合には

(1) 概念学習を行う分の計算コストが増加する，
(2) 十分な精度を持つ政策事前条件の学習に必要な事例数を事前に知ることができない

という点が問題になる．本手法では，事例が集まるまでは政策の再構築を行わない．したがって，事例を多く収集しようとした場合には，政策の再構築をはじめるまでに要するエピソード数が増加する．収集する事例が少なすぎると，概念学習により獲得した政策事前条件の精度が低くなり，十分な性能を持つ政策を再構築することができなくなる．

本論文では決定的環境における決定的政策を対象としたが，確率的環境における確率的政策を対象とする場合には，十分な数の事例を収集するとともに ILP に替えて統計的手法に基づいた概念学習アルゴリズムを用いて政策事前条件を獲得する方法を検討する必要がある．すな

わち，前の政策が失敗する確率の高い状態を再学習の対象とするよう政策事前条件を学習する．

今後は，確率的環境における本手法の有効性の検証，高い精度を持つ政策事前条件の学習に必要な事例数を見積もる方法および環境同定型の強化学習に対する本手法の応用可能性の検討などを行う必要がある．

謝　　辞

本論文に対して貴重なご意見を頂いた担当委員および査読者の方々に深謝いたします．

◇ 参 考 文 献 ◇

[Benson 96] Benson, S. S.: *Learning Action Models for Reactive Autonomous Agents*, PhD thesis, Stanford University (1996).

[Džeroski 98] Džeroski, S., De Raedt, L., and Blockeel, H.: Relational Reinforcement Learning, in *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 136–143 (1998).

[Jacobs 98] Jacobs, N., Driessens, K., and De Raedt, L.: Using ILP-Systems for Verification and Validation of Multi-agent Systems, in Page, D. ed., *Proceedings of the Eighth International Conference on Inductive Logic Programming*, pp. 145–154, Springer (1998).

[Kato 00] Kato, S. and Matsuo, H.: A Theory of Profit Sharing in Dynamic Environment, in Mizoguchi, R. and Slaney, J. eds., *Proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence*, pp. 136–145 (2000).

[McCarthy 69] McCarthy, J. and Hayes, P. J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence, in Meltzer, B. and Michie, D. eds., *Machine Intelligence 4*, pp. 463–502, Edinburgh University Press (1969).

[港 00] 港, 浅田：環境の変化に適応する移動ロボットの行動獲得, 日本ロボット学会誌, Vol. 18, No. 5, pp. 706–712 (2000).

[宮崎 94] 宮崎, 山村, 小林：強化学習における報酬割当ての理論的考察, 人工知能学会誌, Vol. 9, No. 4, pp. 580–587 (1994).

[Muggleton 95] Muggleton, S.: Inverse Entailment and Progol, *New Generation Computing*, Vol. 13, pp. 245–286 (1995).

[Nilsson 94] Nilsson, N. J.: Teleo-Reactive Programs for Agent Control, *Journal of Artificial Intelligence Research*, Vol. 1, pp. 139–158 (1994).

[Quinlan 90] Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, Vol. 5, pp. 239–266 (1990).

[Reid 00] Reid, M. and Ryan, M.: Using ILP to Improve Planning in Hierarchical Reinforcement Learning, in Cussens, J. and Frisch, A. eds., *Proceedings of the Tenth International Conference on Inductive Logic Programming*, pp. 174–190, Springer (2000).

[Ryan 98] Ryan, M. R. K. and Pendrith, M. D.: RL-TOPs: An Architecture for Modularity and Re-Use in Reinforcement Learning, in Shavlic, J. ed., *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 481–487, Morgan Kaufmann Publishers (1998).

[Shen 93] Shen, W. M.: Discovery as Autonomous Learning from the Environment, *Machine Learning*, Vol. 28, pp. 143–156 (1993).

[Sutton 98] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, The MIT Press (1998), 三上, 皆川 共訳: 強化学習, 森北出版 (2000).

〔担当委員：沼尾正行〕

2001 年 3 月 8 日　受理

◇ 付　　録 ◇

A. 背 景 知 識

3･1 節で述べた積み木の問題における背景知識 B の具体例を次に示す．

```
block(a).
block(b).
block(c).
block(d).
```

概念学習を行うときは，上の背景知識にそれまでの状態と行為に関する記述を加えたものをひとつの背景知識として扱う．さらに，ILP システムには，仮説に使用する述語の宣言および正事例・負事例が与えられる．したがって，実際に ILP システムに与えるファイルは次のようになる．ここで，述語 `modeh/2`, `modeb/2` は，それぞれ後件部，前件部に使用できる述語を宣言している．また，各引数の先頭についている`+`, `-`, `#`は，それぞれが入力，出力，定数であることを表している．末尾の `fail/1` は事例であり，先頭に`:-`がついている事例は負事例である．

```
:- modeh(1,fail(+state))?

:- modeb(4,ontable(-block,+state))?
:- modeb(3,on(-block,+block,+state))?
:- modeb(4,clear(+block,+state))?
:- modeb(1,stack(+block,+block,+state))?
:- modeb(1,putdown(+block,+state))?
:- modeb(1,(+block)=(#block))?
:- modeb(1,(+term)=(#term))?

block(a).
block(b).
block(c).
block(d).

ontable(c,s0).
on(d,c,s0).
on(a,d,s0).
on(b,a,s0).
clear(b,s0).
putdown(b,s0).
 (略)

ontable(a,s80).
on(c,a,s80).
clear(c,s80).
ontable(d,s80).
on(b,d,s80).
clear(b,s80).
putdown(b,s80).
 (略)

:-fail(s0).
:-fail(s1).
:-fail(s2).
:-fail(s3).
:-fail(s4).
 (略)

fail(s80).
fail(s81).
fail(s82).
fail(s83).
fail(s84).
 (略)
```

本論文の実験では使用していないが，状態の分類に役立ちそうな他の知識として次の above/3 などが考えられる．

```
above(X,Y,S):-on(X,Y,S).
above(X,Y,S):-on(Z,Y,S),above(X,Z,S).
```

著 者 紹 介

松井 藤五郎(学生会員)

1997 年名古屋工業大学工学部知能情報システム学科卒業．1999 年同大学院工学研究科博士前期課程電気情報工学専攻修了．現在，同博士後期課程に在学中．人工知能，特に機械学習，帰納論理プログラミングなどの研究に従事．情報処理学会会員．

犬塚 信博(正会員)

1987 年名古屋工業大学工学部情報工学科卒業．1992 年同大学院工学研究科博士課程電気情報工学専攻修了．工学博士．同年，同大学電気情報工学科助手．1999 年同大学知能情報システム学科講師．2000 年同大学共同研究センター助教授．現在に至る．1994 年より 1996 年まで英国インペリアルカレッジを訪問．人工知能，特に機械学習，知識表現に関する研究に従事．情報処理学会，電気情報通信学会各会員．

世木 博久(正会員)

1979 年東京大学工学部計数工学科卒業．1981 年同大学院工学系研究科修士課程修了．同年 4 月より三菱電機（株）中央研究所に勤務．1985 年〜1989 年（財）新世代コンピュータ技術開発機構に出向．1992 年より名古屋工業大学工学部知能情報システム学科助教授．1997 年 4 月同学科教授．工学博士．論理プログラミング，演繹データベース等に興味を持つ．電子情報通信学会，情報処理学会，ACM，IEEE Computer Society 各会員．

伊藤 英則(正会員)

1974 年名古屋大学大学院工学研究科博士課程電気電子専攻満了．工学博士．1974 年日本電信電話公社横須賀研究所勤務．1985 年（財）新世代コンピュータ技術開発機構出向．1989 年名古屋工業大学教授．現在知能情報システム学科所属．この間，数理言語理論，計算機ネットワーク通信，OS，知識ベースシステムなどの研究開発に従事．電子情報通信学会，電気学会，情報処理学会，形の科学会，日本ファジィ学会各会員．