非線形的なプランニングにおける一般化手法の提案

2P−10

伊藤智子　鈴木英明　伊藤正樹

NTT ソフトウェア研究所

1 はじめに

プランナーと呼ばれるプランの自動生成系では，状態を変化させるためのいくつかの規則が予め定められ，具体的な初期状態と目標状態の対を問題として与えることで，初期状態から目標状態へ到達するために要する規則の系列または規則を用いて変化する中間的な状態の系列（プラン）を出力する．しかし，プランナーでは具体的な問題のプランは得られるが，一般的な問題に対するプランは得られない．例えば，ハノイの塔の問題では，ディスクの個数を固定した場合の規則の系列は得られるが，汎用的なアルゴリズムは得られない．

汎用的なアルゴリズムを得るために，具体的な問題に対して得られたプランに対して，系列の構造に現れる繰り返しをまとめて一般解を得る手法が提案されている[1]．この手法では，具体的な問題の目標状態を構成する各要素（リテラル）に順序関係を与え，その問題のプランを順序関係に従ってリテラルを並べたグラフ構造で表現する．しかし，具体的なプランとグラフ構造が一致するのは，目標状態の各リテラルが規則によって一度生成されたあと変化することがない線形なプランに限られている．このため，非線形なプランを扱う問題の一般解を得られない．

本稿では，STRIPS や TWEAK などの既存のプランナーを用いて非線形的な問題の具体的プランを生成し，複数の具体的なプランにおいて，状態の一般化をプランの構造の一般化と共に考慮することで，汎用的なプランを生成する機構について提案する．

2 プランの一般化手法の概要

プランナーで作成された具体的な t 個のプラン P_k $(1 \leq k \leq t)$ は任意の状態 $state^k_j$ といくつかの状態の順序列 $P_k = \langle state^k_0, state^k_2, ..., state^k_m \rangle$ で表現される．プランを一般化するためには，(1) 状態を拡張することと (2) 複数の系列における構造の共通部分や繰り返しをまとめることが考えられる．本稿では，状態を拡張する一般化方法および逆の状態を縮退する具体化方法や文献[6]で述べられた具体的なプランにおける状態の同型概念はあらかじめ与えられるとして，(2) について考察する．

いま，与えられたプラン P_k に対する一般化写像を各々 $Ex_1, ..., Ex_t$ とし，具体化写像を $Ins_1, ..., Ins_t$ とする．例えば，ハノイの塔の問題に対して，1枚のディスクに対して一般化写像を $\{a_1 \rightarrow (x_1, x_2, ..., x_n), tb \rightarrow tb\}$ と与えることで，n 枚のディスクに拡張できる．また，具体化写像を $\{x_1 \rightarrow a_1, (x_2, ..., x_n, tb) \rightarrow tb\}$ と与えたとき，先の拡張結果を元に戻すことができる．

状態の一般化および具体化写像を用いて複数の具体的なプランから一般的なプランを求める手法は，既存の問題と新たな問題との対応関係を基にして既存の問題に対する解法を新たな問題に対する解法に変形する変形類推の枠組[2]と類似している．しかし，状態の一般化写像を具体的なプランに適用して得られるプランは，これを具体化したとき，プランナーで生成される具体的なプランに存在するいくつかの状態が存在しない抽象的なプランになるという問題がある．

この問題に対して，得られた抽象的なプランに対して状態の一般化写像および具体化写像を適用することで共通する構造の一般化を行ない，一般的なプランを導出する (図 1).

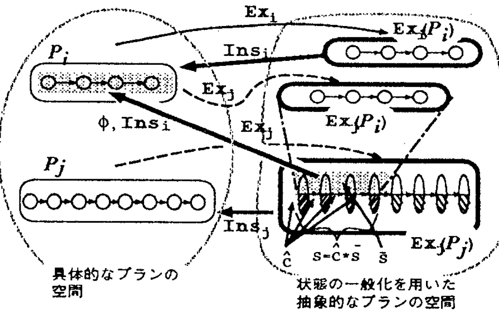

図 1: 解法の一般化の概要図

任意の拡張されたプラン $Ex_j(P_j)$ における $Ex_j(state^j_h)$ と $Ex_j(state^j_{h+1})$ におけるリテラルの異なりが2つ以上存在する場合，$Ex_j(state^j_h)$ から $Ex_j(state^j_{h+1})$ への状態遷移は1回のオペレータの適用で行なわれない．この状態遷移をを S として，S で変化しないリテラルの集合とそれ以外に分ける．これらを各々 $\hat{C}, \tilde{S}$ として，$S = \hat{C} * \tilde{S}$ で表す．このとき，プラン $Ex_j(P_j)$ とより抽象的なプラン $Ex_i(P_i)$ が存在し，$\tilde{S}$ に対して $Ins_i(\tilde{S})$ が P_i またはこれに同型写像 ϕ_l を適用した結果に等しいならば，$\tilde{S}$ は $Ex_j(P_i)$ と同じ構造と考えられる．拡張プラン $Ex_j(P_j)$ における部分的な系列 $\hat{C} * \tilde{S}$ を $\hat{C} * Ex_j(P_i)$ または $\hat{C} * \phi_l(Ex_j(P_i))$ で置き換える操作を繰り返し適用したプランを一般的なプランとして得る．

A Method for Generalizing the Structure of a Non-linear Plan.

Tomoko ITO, Hideaki SUZUKI and Masaki ITO.

NTT Software Laboratories

3 適用例題

非線形的なハノイの塔の問題を例として，本手法の有効性を示す．ディスクが1枚の時のプラン P_1 とディスクが2枚の時のプラン P_2 がプランナーにより次のように生成される。

$P_1 = \langle on(1,tb,a), on(1,tb,b)\rangle$

$P_2 = \langle on(1,2,a) \wedge on(2,tb,a), on(1,tb,c) \wedge on(2,tb,a), on(1,tb,c) \wedge on(2,tb,b), on(1,2,b) \wedge on(2,tb,b)\rangle$

ここで、a,b,c は個々のポールを表し、数字はディスクを表す．ここでは、ディスクの大きさが小さい順に数字を割り当てる．tb はテーブルを表している．on(1,2,a) はポール a でディスク 2 の上にディスク 1 が置かれていることを表し、on(1,tb,c) はポール c でテーブルの上にディスクが置かれている、つまり、ディスク 1 の下に他のディスクは置かれていないということを表している．

プラン P_1, P_2 に対して次のように一般化写像 Ex_1, Ex_2 および具体化写像 Ins_1, Ins_2 を与えて状態を一般化および具体化する．

$Ex_1 : \{1 \rightarrow (1,2,...,n), tb \rightarrow tb\}$

$Ex_2 : \{1 \rightarrow (1,2,...,n-1), 2 \rightarrow n, tb \rightarrow tb\}$

$Ins_1 : \{1 \rightarrow 1, (2,3,...,n,tb) \rightarrow tb\}$

$Ins_2 : \{1 \rightarrow 1, 2 \rightarrow 2, (3,4,...,n,tb) \rightarrow tb\}$

ここで、各一般化写像を使った結果 $Ex_1(P_1)$ と $Ex_2(P_2)$ は次のようになる．

$Ex_1(P_1) = \langle on(\{1,2,...,n\},tb,a), on(\{1,2,...,n\},tb,b)\rangle$

$Ex_2(P_2) = \langle on(\{1,2,...,n-1\},n,a) \wedge on(n,tb,a), on(\{1,2,...,n-1\},tb,c) \wedge on(n,tb,a), on(\{1,2,...,n-1\},tb,c) \wedge on(n,tb,b), on(\{1,2,...,n-1\},n,b) \wedge on(n,tb,b)\rangle$

$Ex_2(P_2)$ の $on(\{1,2,...,n-1\},n,a) \wedge on(n,tb,a)$ から $on(\{1,2,...,n-1\},tb,c) \wedge on(n,tb,a)$ への遷移は 1 回のオペレータの適用では行なえない．この状態遷移を S とすると、S は前の状態から変化しないリテラル $\hat{C} = on(n,tb,a)$ と変化するリテラル

$\tilde{S} = \langle on(\{1,2,...,n-1\},n,a), on(\{1,2,...,n-1\},tb,c)\rangle$

に分けらる．$\tilde{S}$ を Ins_1 を使って具体化すると、

$Ins_1(\tilde{S}) = \langle on(\{1,tb\},tb,a), on(\{1,tb\},tb,c)\rangle$ となる．ここで、規則 $on(\{\alpha,\beta\},\gamma,\square) = on(\alpha,\beta,\square) \wedge on(\beta,\gamma,\square)$ を使って展開すると、$Ins_1(\tilde{S}) = \langle on(1,tb,a) \wedge on(tb,tb,a), on(1,tb,c) \wedge on(tb,tb,c)\rangle$ となる．$on(tb,tb,\square)$ はテーブルの上にテーブルが乗っているとなり無意味なので削除すると、$Ins_1(\tilde{S}) = \langle on(1,tb,a), on(1,tb,c)\rangle$ となり、P_1 に同型写像 $\phi_{b\rightarrow c}$[6] を適用したプランと等しくなる．ここで、同型写像[6] とは、ポール a とポール b のように要素間の対称性、類似性が高く、要素の置換を行なってもプランの構造が変化しない場合に、要素を置換し新しくプランを作成する写像である。同型写像 $\phi_{b\rightarrow c}$ とは、ポール b をポール c に置換する写像で $\phi_{b\rightarrow c}(P_1) = \langle on(1,tb,a), on(1,tb,c)\rangle$ となる．$Ins_1(\tilde{S}) = \phi_{b\rightarrow c}(P_1)$ であるため、$\tilde{S}$ は $Ex_2(P_1)$ と同じ構造である．この操作を $Ex_2(P_2)$ のすべての状態遷移に対して適用すると、$Ex_2(P_2) = (on(n,tb,a) * \phi_{b\rightarrow c}(Ex_2(P_1))$ $,on(n,tb,b) * \phi_{a\rightarrow c}(Ex_2(P_1)))$ が得られる．ここで、$Ex_2(P_2)$ は P_n を表すが、$Ex_2(P_1)$ は P_1 に要素 2 が存在しないので、$Ex_2(P_1)$ には、n の要素が含まれない．そのため、$Ex_2(P_1)$ はプラン P_{n-1} となる．$Ex_2(P_2)$ を P_n,$Ex_2(P_1)$ を P_{n-1} で置き換えると、ハノイの塔の一般化プラン $P_n = (on(n,tb,a) * \phi_{b\rightarrow c}(P_{n-1}), on(n,tb,b)$ $\phi_{a\rightarrow c}(P_{n-1}))$ が得られる．

4 関連研究

文献 1) では，本稿と同様に STRIPS 形式でプランを記述する場合，証明における命題の繰り返しを判別しそれらをまとめるための数の一般化[3),4)] というより線形なプランニング問題の特殊解であるプラン系列をグラフ構造で表現し，その構造に対して数の一般化を用いる．

また，ABSTRIPS や ABTWEAK[5)] などでは，一般解を求めるのではなく，抽象的にオペレータを記述し直すことによりプランナーの計算量をできるだけ少なく，すなわち，探索空間を絞り込むための枝がりの方法を考慮している．

5 おわりに

STRIPS 形式でプランを記述し，プランの系列の構造のみを考慮した場合，プランの例題を一般化できるのは，線形なプランニング問題に限られていた．本手法では，系列の構造のみでなく予め状態の一般化を与えることで，非線形なプランニング問題にまで適用範囲を拡張する方法を提案した．部分的な繰り返しがネストした場合の一般化や、複数の例題からの状態の一般化写像の生成法などは今後の課題である．最後に、日頃御指導頂く後藤部長ならびにソ技部の皆様に深謝します．

参 考 文 献

1) I. Bostrom: "Generalizing the Order of Goals as an Approach to Generalizing Number", 7'th Conference of Machine Learning, 1991, pp. 260–267.

2) J. G. Carbonell: "Learning by Analogy:Formulating and Generalizing Plans from past Experience", Machine Learning An Artificial Intelligence Approach, Morgan Kaufmann, pp. 137–161.

3) J. W. Shavlik and G. F. DeJong: "An Explanation-Based Approach to Generalizing Number", IJCAI'87, 1987, pp. 236–238.

4) J. W. Shavlik: "Acquiring Recursive Concepts with Explanation-Based Learning", IJCAI'89, 1989, pp. 688–693.

5) Q. Yang and J. D. Tenenberg: "ABTWEAK: Abstracting a Nonlinear, Least Commitment Planner", Eight National Conference of Artificial Intelligence AAAI-'90, 1990, pp. 204–209.

6) 山口 智浩: "同型概念を用いた抽象化問題解決", 情報処理学会研究報告,93-AI-87, 1993, pp. 75–84.