

所与の仕様との逐次的な照合に基づくパターンの獲得

芦田 昌也[†] 竹内 昭浩[†] 柴田 史久^{††}
角 所 考^{†††} 北 橋 忠 宏^{††}

本稿は、問題解決のための作業の進行状況が一種のパターンとして観測されることを前提に、指示に基づく問題解決の手法を、与えられた指示と整合性の高いパターンを発見する処理として定式化し、その具体的な手法を提案するものである。問題解決のために与えられる指示は記号列に変換され、発見すべきパターンを特定する仕様と見なす。パターンは部分的に観測され、仕様と照合するために認識処理が施される。仕様と最も整合するパターンの発見は、部分的に観測されたパターンの認識結果を節点とする木構造における節点の探索プロセスとなる。節点を評価するためある条件を満足する評価関数を用いると、仕様と最も整合するパターンの認識結果に対応する節点を必ず発見することが示される。また、提案手法を未知環境における移動ロボットのナビゲーションに適用し、評価関数の異なる場合の動作例をシミュレーションにより示す。

Pattern-obtaining Method Based on Sequential Matching with the Given Specification

MASAYA ASHIDA,[†] AKIHIRO TAKEUCHI,[†] FUMIHISA SHIBATA,^{††}
KOH KAKUSHO^{†††} and TADAHIRO KITAHASHI^{††}

This article proposes a method for problem-solving based on instructions for a task achievement. Assuming that progress of the task is observable by sensors and the result of observation is thought as pattern information, the method is described as finding a pattern satisfying specification that is a symbolic description of given instructions. The method is based on informed tree-search and is successful in obtaining the pattern corresponding to the specification using appropriate evaluation functions.

1. はじめに

人間は様々な方法を利用して、問題の解決を行うことができる。事前の知識が不十分であり、過去に解決した経験のない、初めて取り扱う問題の場合には、解決に至るまでの指示を受けてから問題解決に着手する方法が考えられる。この方法は実際に作業をしながらその状況を観測し、指示と作業の進行状況との整合性を確認しつつ問題解決を図るものとしてモデル化できよう。本稿はこのようにモデル化される問題の探索について論じるものである。

このモデル化において状態を規定するものは、内・

外界センサを用いて観測される作業の進行状況であり、それらは一種のパターンとして獲得されるものとする。見知らぬ土地で現在地から目的地までの道順の指示を頼りに移動することは、この種の問題の典型例であり、経路上の道しるべや地形の特徴などの観測結果が獲得されるパターンとなる。

指示が言葉¹⁾や実演^{2),3)}などで表現される場合には、指示と作業の進行状況との整合性を確認するために獲得したパターンに認識処理を施し(以後、作業認識)、指示と照合しうる記述を得る必要がある。作業途中では、その時点までの進行状況を表現する限定されたパターンが観測されるにとどまり、その部分的なパターンには認識対象全体が含まれていない可能性がある。また、認識対象の対象モデルと評価関数が与えられても、一般にはある程度の認識誤りは不可避である。そのため、作業認識の結果と与えられた指示との整合性の評価は不正確なものにならざるをえない。これに起因して、与えられた指示に沿って作業が進行していると判断しても、現実には指示どおり進行していないこ

[†] 和歌山大学経済学部

Faculty of Economics, Wakayama University

^{††} 大阪大学産業科学研究所

The Institute of Scientific and Industrial Research,
Osaka University

^{†††} 京都大学総合情報メディアセンター

Center for Information and Multimedia Studies, Kyoto
University

などが生じ、問題解決を完了できない恐れがある。

たとえば、「2つ目の交差点を右折せよ」という指示が与えられている場合に、図1の中央付近を隣接する2つの交差点と見なすか、1つの交差点と見なすかで走行経路は異なり（同図、実線および破線の経路）、指示された経路に沿って目的地に到達できないことがある。問題解決を完了するためには、指示に沿った作業の実行と進行状況の詳細な観測に加え、過去の作業認識の結果をつねに再検討し、状況に応じて過去に遡って作業をやり直す必要がある。

このような指示に基づく問題解決の手法は、部分的なパターンの観測と認識、および指示との照合を逐次的に実行し、指示と整合性の高い全体的なパターンを発見する処理として基本的には定式化できる。具体的には、部分的なパターンの認識結果を節点とし、作業の進行にともなって成長する木構造において、指示と最もよく一致する認識結果に対応した節点を探索するプロセスとして実現される。本稿ではこの探索手続きを提案し、その手続きが適切な条件下で、指示と最もよく一致する認識結果に対応した節点を必ず発見することを示す。以下2章、3章において、指示に整合するパターンを発見する処理、およびその実現手法を論じ、その特徴に言及する。4章では、提案手法を未知環境におけるナビゲーションに適用した動作例を示し、若干の検討を加える。5章では、本研究の意義と残された課題について述べる。

2. 指示に整合するパターンの発見

本稿で提案する手法が取り扱う問題の前提条件と、提案手法の概略について述べる。

2.1 問題設定

対象となる問題においては、作業の進行状況がセンサを用いて観測可能であり、その観測結果は認識処理を施すことにより、与えられた指示と照合可能な記述が得られることを前提とする。また作業の内容は、適切な手順を経てやり直しが可能であるものとする。

問題解決に至るまでの指示は、あらかじめ定められた記号を用いて記号列に変換する。この記号列は発見すべきパターンの仕様となる。パターンを発見する主体をエージェントと呼ぶ。これは問題解決のための作業を実行する主体に対応する。作業途中ではパターンが部分的に観測される。作業の進行にともなってパターンの全体が次第に明らかになる。問題解決のための作業手順が異なると、作業の進行状況も一般には異なるため、エージェントが観測可能なパターンは複数存在する。十分に時間をかければ、個々のパターンは

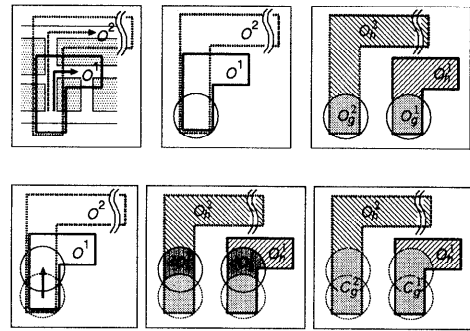


図1 パターンの模式図

Fig.1 Illustration of patterns.

その部分的な観測を経て全体が観測可能であり、存在するすべてのパターンも同様にして観測可能であると仮定する。この仮定が成立する作業環境を有限であると表現する。

記号列で表現された仕様と照合し整合性を評価するために、パターンには認識処理を施す。認識対象のモデルは賦与され、認識結果は記号列で記述する。この記述をパターンの解釈と呼ぶ。パターンの解釈と仕様とを照合し、両者の一致の程度（以後、一致度）を評価する。一致度の評価関数も与えられるものとする。

以上の前提の下で、本稿では有限の作業環境に混在する様々なパターンの中から、それらの部分的な観測と認識、および仕様との照合を繰り返し、与えられた評価関数を用いて仕様に最も整合するパターンを発見する探索手法について論じる。

2.2 手法の概略

図1は、指示された経路に沿って環境中を移動する場合に観測されるパターンを、太い実線の囲み O^1 と太い破線の囲み O^2 によって表現した模式図である。円で示された範囲をエージェントの観測範囲とする。各パターンにおいて観測された部分を部分パターン O_g^i 、未観測の部分を残余パターン $O_h^i (= O^i - O_g^i)$ と呼ぶことにする。エージェントの行動にともなってパターンの新たな部分が観測される。この部分を追加パターン dO_h^i とする。追加パターンと観測済の部分パターンを併合し、部分パターンは更新される ($O_g^i \leftarrow O_g^i \cup dO_h^i$)。これにともない残余パターンも更新される ($O_h^i \leftarrow O_h^i - O_g^i$)。

パターンの部分的な観測を経て、仕様と整合するパターンの全体を発見するために、本稿では次の手法を提案する。部分パターンに対する認識処理において、その評価のいかんにかかわらず可能な解釈をすべて生成する。この時点で残余パターンは未観測であるが、実際に観測された場合に得られるであろう解釈を推測

する。部分パターンの解釈と残余パターンの推測された解釈の接続をそのパターン全体の解釈と見なす。このようにして得られたすべての解釈と仕様との一致度を部分パターンが更新されるたびに再評価する。最も一致度の高い解釈を生成している部分パターンに対する残余パターンを観測できるような行動を実行し、追加パターンを観測する。以下これを繰り返し、仕様にも最も一致する解釈が得られるパターンを解と見なす。

部分パターンの更新にともなって、ある時点まで仕様と部分的に整合していた部分パターンが整合しなくなることや、あまり整合していない部分パターンが次第に整合しはじめるような変化が生じる。残余パターンから得られる解釈を予測することと、その時点までに生成された解釈と仕様との一致度をつねに再評価することによりこの整合性の変化に対応し、対象とするパターンを切り替えながら処理を継続することで、仕様と最も整合するパターンの発見を目指す。この一致度の再評価によるパターンの切替えは、過去の認識結果の再検討と、作業の修正・再実行に対応する。

3. 木探索による手法の実現

前章で述べたように、本稿で提案する手法では、状態が過去に遡って再評価される。そこで求解の過程を以下のような木探索として実現する。

3.1 解釈木

パターンは複数存在することや、1つの部分パターンから複数の解釈が得られることから、パターンの解釈も一般には複数存在すると考えられる。部分パターンに認識処理を施して得られる長さ x の解釈のうち、 y 個目のものを記号列 $P_x^y = \langle p_1, p_2, \dots, p_x \rangle$ と記述する。 p_1, \dots, p_x のそれぞれを解釈の要素と呼ぶ。 P_x^y と解釈された部分パターンが更新されても、 P_x^y に変化は生じない場合と、部分パターンの更新により、 P_x^y に新たな解釈の要素 p_{x+1}^i が追加される場合の2通りがある。 i は追加される要素が幾通りか存在する可能性を示す添字である。新たな要素の追加は、新たな解釈 $P_{x+1}^i = \langle p_1, \dots, p_x, p_{x+1}^i \rangle$ の生成を意味する。この過程は探索木で表現される。解釈を節点(状態)とし、部分パターンの更新前の解釈 P_x^y と更新後の解釈 P_{x+1}^i とを枝で接続する。枝が解釈の派生関係を示し、子節点が親節点の解釈を継承する木構造となる(図2)。この木構造を解釈木と呼ぶことにする。なお、解釈と節点は1対1に対応するため、今後は解釈とそれに対応する節点の両方を記号 P_x^y で表現する。

部分パターンの更新による新たな解釈の生成にともない解釈木は成長する。この成長する木構造上で仕様

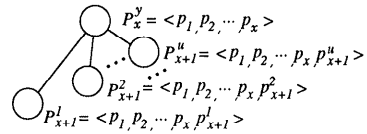


図2 木構造による探索空間の表現
Fig. 2 Search space by tree structure.

に最も整合する解釈に対応する節点を探索する。探索の際には、仕様と解釈との一致度を評価する評価関数を定義し、その評価値に基づく探索手法を利用する。

3.2 評価関数

仕様と解釈とを照合し、両者の一致度を評価する評価関数を定義する。仕様は記号列 $\langle c_1, c_2, \dots, c_n \rangle$ で記述する。 n は仕様の長さである。仕様を記述する要素 c_i とパターンの解釈 $\langle p_1, p_2, \dots, p_n \rangle$ の要素 p_i との一致度を評価する評価関数の存在を仮定する。これを照合関数と呼ぶ。

仮定1 照合関数の存在: 仕様を記述する記号 c_i と解釈の要素 p_i を照合し、両者の一致度を $[0, 1]$ の実数値で表現する照合関数 $k_{c_i}(p_i)$ が存在する。一致度が高くなるほど関数値は大きくなるものとする。□

仕様と解釈との照合は、先頭の要素から順に実行され、その一致度は次のように定義される。

定義1 部分パターンの解釈と仕様との一致度: 仕様 $\langle c_1, c_2, \dots, c_n \rangle$ と部分パターンの長さ x の解釈 $P_x^y = \langle p_1, \dots, p_x \rangle$ との一致度 $g(P_x^y)$ は照合関数 $k_{c_i}(p_i)$ を用いて次のように表される。

$$g(P_x^y) = \sum_{i=1}^x k_{c_i}(p_i) \tag{1}$$

□

仕様の記号列と比較して部分パターンから得られた解釈の記号列が短い場合、仕様の残りの部分と残余パターンから得られるであろう解釈との一致度を推測する評価関数の存在を仮定する。これを推定値関数と呼ぶことにする。

仮定2 推定値関数の存在: 長さ n の仕様 $\langle c_1, \dots, c_x, c_{x+1}, \dots, c_n \rangle$ が与えられ、 $P_x^y = \langle p_1, \dots, p_x \rangle$ と解釈される部分パターンが観測されている場合に、その残余パターンから得られるであろう解釈が、仕様の $\langle c_{x+1}, \dots, c_n \rangle$ の部分に対して、どの程度一致するかを推定する推定値関数 $h(P_x^y)$ が存在する。□

照合関数と推定値関数の存在を前提として、部分パターンと残余パターンからなるパターン全体の解釈と仕様との一致度は次のように定義される。

定義2 パターン全体の解釈と仕様との一致度: 部分

パターンの解釈 P_x^y と仕様との一致度 $g(P_x^y)$ と、推定値関数 $h(P_x^y)$ を用いて、パターン全体の解釈と仕様との一致度 $f(P_x^y)$ は次のように表される。

$$f(P_x^y) = g(P_x^y) + h(P_x^y) \quad (2)$$

□

パターン全体の解釈と仕様との一致度の評価関数により算出される評価値に基づいて、仕様と最も整合する解釈に対応する節点を探索する。

3.3 解釈木の探索手法

部分パターンの更新にともない、その認識結果である解釈と仕様との一致度は変化する。1つの部分パターンから複数の解釈が得られている場合には、それぞれの解釈と仕様との一致度がすべて変化することや、それぞれの解釈から新たな解釈が派生する可能性がある。ただし、ある解釈から可能なすべての解釈が同時に派生するような部分パターンの更新が行われるとは限らないために、その時点では派生しない場合も起こりうる。これらの現象は解釈木上で次の性質となって現れる。

- (a) 複数の節点の評価が変化することがある。
- (b) 複数の節点から子節点が生成されることがある。
- (c) 子節点の生成の際に、可能なすべての子節点が同時に生成されるとは限らない。

解釈木上の各節点は式(2)で評価し、探索手続きはその値が最大となる節点を探索する。式(2)の形式の評価関数を用いる探索手続きとして、A*アルゴリズムが知られている^{5),6)}。本稿では解釈木の性質を考慮し、従来のA*アルゴリズムに修正を加えた探索手続きを提案する。

A*アルゴリズムを用いた木構造の探索では、各節点の評価は初期節点からその節点を經由し、目標節点に至る経路のコストが用いられる。各時点で最も評価の良い節点のみ展開の対象となる。その節点から展開可能なすべての子節点が現れることを前提とし、いったん展開された節点の評価値は、次にいずれかの節点が展開されるまで変化することはない。この状況の下で最終的にコスト最小の経路を発見しようとする⁷⁾。一方、本稿で探索の対象となる木構造では、節点の評価に一致度を用いており、探索は一致度の最も高い節点を発見するように進められる。節点の評価はいずれかの子節点の生成時だけでなく、部分パターンの更新ともなっており変動する。また節点の評価とは無関係に、いずれかの節点で子節点が生成される可能性がある。さらに子節点の生成時には、一部の子節点が未生成のまま残されることもある。このような不確定な要素を考慮し、最終的に一致度最大となる節点を発見するた

めには、部分パターンを更新する際に必ず節点を再評価するという修正を加える必要がある。

図3にこの手続きを示す。表1、表2はそれぞれ図3の手続きで使用される副手続きと変数の説明である。plan(n)はその時点で最も評価の高い節点(以後、注目節点)の評価がさらに高くなる追加パターンの観測を目的とした行動を計画する(step 3)。そのような計画ができない場合は、その節点の子節点がすべて生成されているものと見なし、その節点をclosedリストに移す(step 4, 5, 6)。action(p)は計画された行動を実行し、追加パターンを観測した後、部分パターンを更新する(step 8)。その結果、注目節点であるかどうかにかかわらず生成された節点はopenリストの後ろに追加する(step 9)。openリストのすべての節

step 0	init(open)
step 1	n := first(open)
step 2	if goal(n) then return
step 3	while (plan(n) == NULL) {
step 4	remove(n, open)
step 5	add(n, closed)
step 6	n := first(open) }
step 8	action(plan(n))
step 9	step 8 で生成された子節点を open の後ろに追加する
step 10	open のすべての節点を再評価し、評価値の降順にソートする
step 11	step 1 に戻る

図3 探索手続き

Fig. 3 Search procedure.

表1 探索手続きで用いられる副手続き

Table 1 Sub procedures.

副手続き	仕様
init(l)	リスト l を初期化する
first(l)	リスト l の先頭要素を取り出す
remove(n, l)	リスト l から節点 n を削除する
add(n, l)	リスト l に節点 n を加える
goal(n)	節点 n が解であるかどうか判定する
plan(n)	節点 n の解釈に基づいて観測行動を計画する
action(p)	計画 p の実行

表2 探索手続きで用いられる変数

Table 2 Variables used by the search procedure.

変数名	説明
n	注目節点
open	次に注目節点となる節点の候補のリスト
closed	すべての子節点が生成された節点のリスト

点を再評価した後、*open*リスト中で i 番目、 $i+1$ 番目の順に連続して並んでいる節点 m_i, m_{i+1} の評価値 $f(m_i), f(m_{i+1})$ の関係が $f(m_i) < f(m_{i+1})$ である場合には、 m_i と m_{i+1} の順序を交換する。この操作を順序の交換が実行されなくなるまで繰り返し（すなわち、バブルソートである）、評価値の降順に節点を並べ替える（step 10）。なお $plan(n)$ と $action(p)$ は、実際の問題における部分パターンへの表現に依存するため、これらの具体的な形式は問題ごとに個別に定義されるものとなる。

3.4 探索手続きの特徴

提案する探索手続きは、一致度が最大となる節点を必ず発見することが示される。仕様の長さを n 、長さ n の解釈で仕様と最も一致する解釈（以後、最適解）に相当する節点を P_n^T 、長さ n の解釈で P_n^T 以外の解釈（以後、準最適解）に相当する任意の節点を P_n^F 、最適解、準最適解以外の任意の節点を P_x^y とする。節点 P_n^T の評価値を f^* 、節点 P_x^y に対する推定値関数 $h(P_x^y)$ の真の値を $h^*(P_x^y)$ とする。 $h(P_x^y)$ は残余パターンの解釈 $\langle p_{x+1}, \dots, p_n \rangle$ が存在しない時点での仕様との一致度の推定値である。一方、 $h^*(P_x^y)$ の値は、残余パターンが観測され、その解釈 $\langle p_{x+1}, \dots, p_n \rangle$ が実際に得られた場合の一致度に対応する値である。したがって、 f^* と $h^*(P_x^y)$ の値は、探索の途中の時点では知ることは現実には不可能である。

まず、図3の探索手続きにおいて次の補助定理が成立する。

補助定理1 次の条件が満足される時、必ず P_n^T を発見できる。

解釈木のすべての節点について、

$$h(P_x^y) \geq h^*(P_x^y) \quad (3)$$

$$h(P_n^T) = 0 \quad (4)$$

$$h(P_n^F) = 0 \quad (5)$$

が成立し、かつ、解釈木の根から最適解に至る経路上の節点 P_x^T が現れる。

証明： P_x^T が存在するにもかかわらず、 P_n^T を発見せずして終了する可能性がないことを証明する。 P_n^F の評価値は式(2)と式(5)から $f(P_n^F) = g(P_n^F) + h(P_n^F) = g(P_n^F)$ である。 P_n^F は準最適解であるから、次式が成立する。

$$f^* > f(P_n^F) = g(P_n^F) \quad (6)$$

P_x^T の評価値 $f(P_x^T) = g(P_x^T) + h(P_x^T)$ に式(3)を適用すると次式が導かれる。

$$f(P_x^T) \geq g(P_x^T) + h^*(P_x^T) = f^* \quad (7)$$

さて、 P_x^T が存在するにもかかわらず、 P_n^T を発見して終了すると仮定すれば、 P_x^T と P_n^F の評価値に関し

て、次式が成立しなければならない。

$$f(P_n^F) = g(P_n^F) > f(P_x^T) \quad (8)$$

ここで、式(7)と式(8)から

$$g(P_n^F) > f(P_x^T) \geq f^* \quad (9)$$

が導かれる。これは式(6)と矛盾する。ゆえに P_x^T が存在するときに、 P_n^F を発見して終了することはない。

□

補助定理1は、式(3)、式(4)、式(5)の成立と、最適解に至る経路上の節点の存在が前提になっているため、これらの成立可能性に言及しておく。

補助定理2 節点 P_n^T と P_n^F に対する推定値関数の値は、 $h(P_x^y) \geq h^*(P_x^y)$ に反することなく $h(P_n^T) = 0$ 、 $h(P_n^F) = 0$ とすることができる。

証明：節点 P_n^T と P_n^F には、仕様 $\langle c_1, \dots, c_n \rangle$ と照合できる解釈の要素がすべて現れているから、 $h^*(P_n^T) = 0$ 、 $h^*(P_n^F) = 0$ である。したがって、式(3)に反することなく $h(P_n^T) = 0$ 、 $h(P_n^F) = 0$ とすることができる。

□

補助定理3 $h(P_x^y) \geq h^*(P_x^y)$ を満足し、かつ最適解に至る経路上の節点が探索の過程で必ず現れる推定値関数 $h(P_x^y)$ が少なくとも1つ存在する。

証明： $h^*(P_x^y)$ の値は実際に残余パターンが観測され、 $\langle p_{x+1}, \dots, p_n \rangle$ と解釈された場合の $\langle c_{x+1}, \dots, c_n \rangle$ との一致度である。仮定1により、 $0 \leq h^*(P_x^y) \leq n-x$ である。ゆえに、次式の推定値関数が存在する。

$$h(P_x^y) = n-x \quad (10)$$

さて、最適解に至る経路上の節点を P_x^T 、その子節点を P_{x+1}^y とする。ここで $f(P_x^T)$ と $f(P_{x+1}^y)$ の差 d は次式ようになる。

$$\begin{aligned} d &= f(P_x^T) - f(P_{x+1}^y) \\ &= h(P_x^T) - h(P_{x+1}^y) \{g(P_{x+1}^y) - g(P_x^T)\} \end{aligned} \quad (11)$$

式(10)の推定値関数を用いると、式(11)第1項と第2項の部分は

$$\begin{aligned} h(P_x^T) - h(P_{x+1}^y) &= (n-x) - \{n - (x+1)\} \\ &= 1 \end{aligned} \quad (12)$$

である。式(11)第3項と第4項の部分は、式(1)と子節点が親節点を継承することから次式ようになる。

$$\begin{aligned} g(P_{x+1}^y) - g(P_x^T) &= \sum_{i=1}^{x+1} k_{c_i}(p_i) - \sum_{j=1}^x k_{c_j}(p_j) \\ &= k_{c_{x+1}}(p_{x+1}) \end{aligned} \quad (13)$$

仮定1により $0 \leq k_{c_{x+1}}(p_{x+1}) \leq 1$ であるから $d \geq 0$ 、すなわち

$$f(P_x^T) \geq f(P_{x+1}^y) \quad (14)$$

が成立する。このことは、評価値の変動に応じて注目節点に移る場合において、 P_x^T の子節点が注目節点として選択されるようになるのは、 P_x^T の子節点がすべて生成された後であることを示している。 P_x^T のすべての子節点が生成されると最適解に至る経路上の節点 P_{x+1}^T が必ず含まれる。また、解釈木の根も最適解に至る経路上の節点の1つである。したがって、必ず最適解に至る節点が見れる。ゆえに式 (10) は、探索の過程で最適解に至る経路上の節点が見れる推定値関数の1つである。 □

以上から次の定理が成立する。

定理 1 提案された探索手続きは、仮定 1 の条件を満足する照合関数と式 (3), 式 (4), 式 (5), および式 (14) を満足する推定値関数を用いることで必ず解を発見する。

証明: 補助定理 1, 補助定理 2, 補助定理 3 より明らかである。 □

提案した探索手法が必ず解を発見することは、仕様と最も整合する解釈を与えるパターンの発見を保証することに対応する。そのようなパターンを発見する手法として、指示に基づく問題解決の手法が定式化されていることを考慮すると、定理 1 は本稿で対象とする問題に対する問題解決の完了を保証するものとらえることができる。

4. 未知環境におけるナビゲーションへの応用

本章では、移動経路の指示に基づいて街路状の未知環境を移動するロボットのナビゲーションに提案手法を適用し、その具体的な動作例を示す。

4.1 問題設定

図 4 に示す街路状の環境において、 S から G への経路を「最初の十字路を右折し、すぐ次の十字路で停止せよ」という指示に従ってロボットを移動させる。移動経路の指示が発見すべきパターンの仕様となり、ロボットの走行経路の観測結果が部分パターンに対応する。仕様と最も整合するパターンを発見することによって、指示された経路に沿った目的地までの移動が

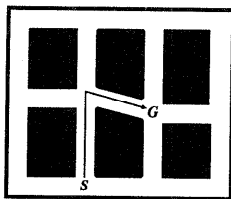


図 4 街路状の環境
Fig. 4 Town-like environment.

達成されることをシミュレーションにより示す。

4.2 仕様の記述

移動経路の指示を発見すべきパターンの仕様として次の形式で記述する。

$$\langle (T_1, D_1), (T_2, D_2), \dots, (T_n, D_n) \rangle \quad (15)$$

T_i は経由する交差点の種類、 D_i はその交差点での進行方向を示す記号である。種類や進行方向が移動経路の指示に明示されていない箇所はいずれも記号 * で表現する。仕様の第 1 項は初期位置とする。なお、この例題での指示「最初の十字路を右折し、すぐ次の十字路で停止せよ」は次のように記述される。

$$\langle (*, *), (4, 右), (4, *) \rangle \quad (16)$$

4.3 部分パターンと解釈

ロボットは自身を中心とする円内の全方位を観測し (図 5(a)), その円内に存在する障害物までの距離を計測する。得られた距離情報に基づき、障害物領域を 1, 自由領域を 0 とした占有グリッド表現に変換する⁸⁾。図 5(b) は、1 回の観測で獲得される部分パターンの模式図である。画像処理的手法を用いて部分パターンの自由領域の骨格線とその分岐点を抽出する⁹⁾ (図 5(c))。抽出された骨格線と分岐点に環境中の通路や交差点を対応させ、部分パターンを部分的な経路として解釈する。

分岐点とそれに接続する骨格線を交差点として認識するとき、複数の認識結果が生じる。たとえば図 6 の模式図のように 2 つの分岐点が抽出された場合は、図 6(a) に示すように分岐点 1 つを 1 つの交差点とする認識結果と、図 6(b) に示すように 2 つの分岐点を 1 つの交差点とする認識結果などがある (3 つ以上の分岐点が抽出された場合も同様である)。分岐点が 1 つしか抽出されていない時点でも、それを複数の分岐点からなる交差点の一部として認識することも可能で

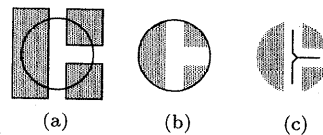


図 5 部分パターンと骨格線・分岐点の例
Fig. 5 Partial pattern, skelton and junction.

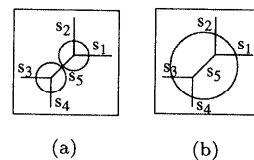


図 6 外部骨格線と内部骨格線の例
Fig. 6 Type of skelton.

ある. このように分岐点の集合を個々の交差点として認識した結果が解釈の要素であり, それらの列からなる経路全体としての認識結果がパターンの解釈である.

4.4 評価関数

1つの分岐点を1つの交差点として認識する場合, その分岐点から延びる骨格線を外部骨格線(図6(a)の s_1 から s_5)と呼ぶ. 複数の分岐点の集合を1つの交差点として認識する場合, その集合の要素で隣り合う分岐点間を接続する骨格線を内部骨格線(図6(b) s_5)と呼び, それ以外の骨格線を外部骨格線(図6(b) s_5 以外)と呼ぶ(今後, 両者の区別が不必要である場合は単に骨格線と表記する). 内部骨格線は通路が交差した部分に作られる交差点内部の領域に対応し, 外部骨格線は隣接する交差点に至る通路に対応する.

まず, 骨格線を通路として認識した場合の評価関数を定義する. 骨格線 s_i の長さを l_{s_i} , L を正定数とする.

$$S(s_i) = 1 - \frac{L}{l_{s_i} + L} \tag{17}$$

長い骨格線ほど評価が高くなり, 部分パターンの更新にともなって骨格線の長さが変化した場合には, それに応じて評価値が変化する.

次に, 1つ以上の分岐点の集合 V を1つの交差点 p_{i_V} として認識した場合の評価関数を定義する. V における内部骨格線の集合を I_V , 外部骨格線の集合を E_V , 外部骨格線の総数を N_V とする.

$$C(p_{i_V}) = \left\{ \sum_{s_u \in E_V} S(s_u) - \sum_{s_v \in I_V} S(s_v) \right\} / N_V \tag{18}$$

ただし分子が負になる場合, 評価値は0とする. 通路に対応する外部骨格線が長く, 交差点内部の領域に対応する内部骨格線が短いほど評価は高くなる. 交差点を構成している骨格線の長さの変化により, 交差点の評価は変化する.

仕様の要素 $c_i = (T_i, D_i)$ との照合対象である交差点 p_{i_V} において, 進行方向 D_i に通路が延びている場合に1, それ以外の場合は0をとる変数を a_i とし, 次式のように照合関数を定める.

$$k_{c_i}(p_{i_V}) = \begin{cases} a_i C(p_{i_V}) & T_i = N_V \\ 0 & T_i \neq N_V \end{cases} \tag{19}$$

たとえば, 図5(c)の分岐点の集合 V を3差路 p_{i_V} であると認識し, $c_i = (3, \text{右})$ と照合する場合, $k_{c_i}(p_{i_V}) = C(p_{i_V})$ であるが, $c_i = (3, \text{左})$ と照合する場合は $k_{c_i}(p_{i_V}) = 0$ となる. なお抽出された骨格線

の長さは $l_{s_i} > 0$ であるから, 式(17)の値域は $(0, 1)$ の実数値である. そのため式(18)の分子は N を超えることはなく, 式(19)の照合関数の値域は $[0, 1)$ となり仮定1の条件を満足する.

4.5 動作例

推定値関数が異なる2つの動作例を示す. 仕様と最も一致する解釈を発見するエージェントの探索の様子を, 解釈木の成長過程の模式図と節点の評価の変動を記したグラフに示す. また, このエージェントが追加パターンを観測するための行動の様子を骨格線で描かれる地図(以下, 走行経路図)の変化として示す.

図7, 図10は, エージェントが注目節点を切り替えた時点と探索を終了した時点の走行経路図である. 各図中, 実線の矩形の部分は1つの交差点として認識した部分であり, 破線の矩形は1つの交差点の一部として認識している部分である. これらを太線で接続した部分が, その時点で指示に沿っていると判断した走行経路に対応する.

図8, 図11はエージェントが注目節点を切り替えた時点と探索を終了した時点の解釈木の模式図である. 各時点の注目節点は灰色で示すが, その他の節点の多くは模式図への記入を省略した. 図中A~Eの記号が付けられた節点の解釈は, 図7(a)~(e), および

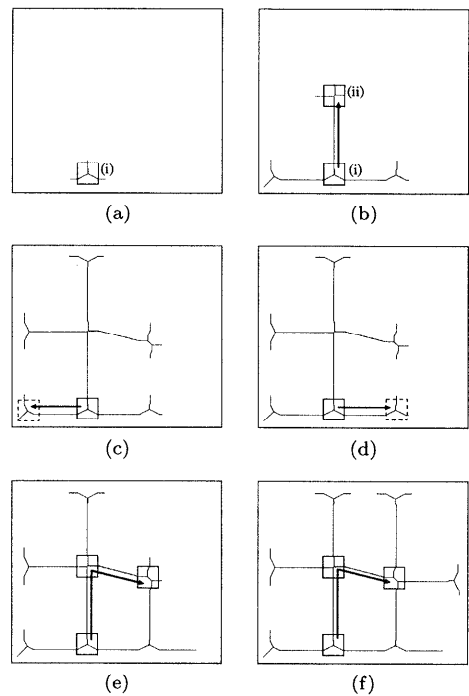


図7 動作例1 (走行経路図)
Fig. 7 Example 1 (route maps).

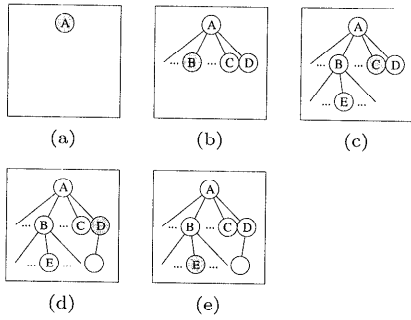


図 8 動作例 1 (解釈木の模式図)
Fig. 8 Example 1 (illustrations of tree).

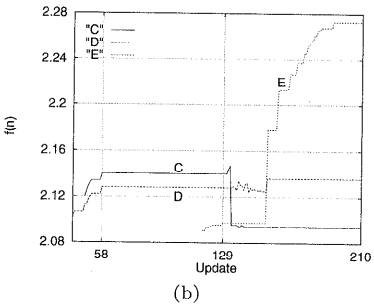
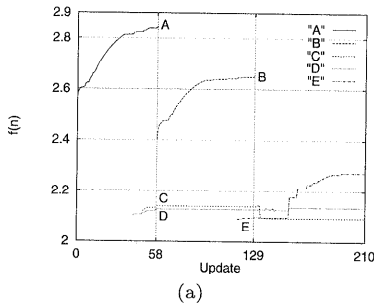


図 9 動作例 1 (節点の評価)
Fig. 9 Example 1 (evaluation of nodes).

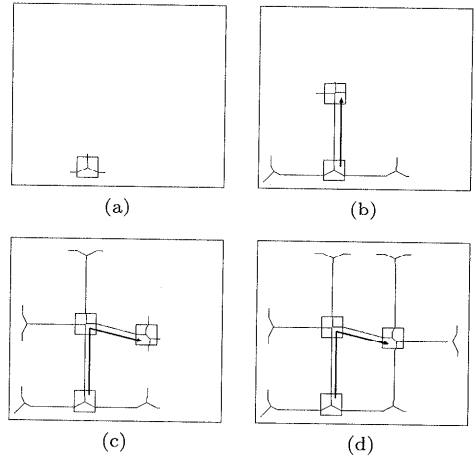


図 10 動作例 2 (走行経路図)
Fig. 10 Example 2 (route maps).

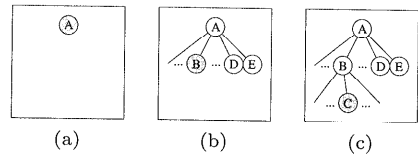


図 11 動作例 2 (解釈木の模式図)
Fig. 11 Example 2 (illustrations of tree).

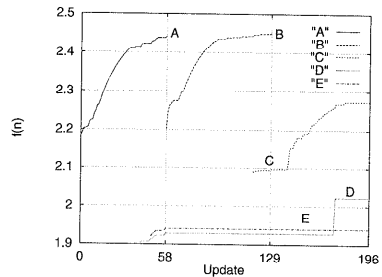


図 12 動作例 2 (節点の評価)
Fig. 12 Example 2 (evaluation of nodes).

図 10 (a)~(c) の走行経路に対応する。

図 9 (a), (b), 図 12 は横軸に部分パターンの更新回数, 縦軸に節点の評価値をとり, 代表的な節点の評価の変動を示したグラフである。図中 A~E のラベルが付けられたグラフはそれぞれ図 8, 図 11 の節点に対応する。図 9 (b) は, 同図 (a) における節点 C, D, E のみ表示したグラフである。グラフの開始点は子節点が生成された時点 (解釈が生成された時点) であり, 終了点は節点が closed リストに移された時点に対応する。

4.5.1 動作例 1

仮定 1 を満たす式 (19) の照合関数と, 式 (3), 式 (4), 式 (5), および式 (14) を満たす式 (10) の推定値関数を利用し, 定理 1 を満足する評価関数

$$f(P_x^y) = \sum_{i=1}^x k_{c_i}(p_i) + (n - x) \quad (20)$$

により節点の評価を算出した場合の動作例を示す。

初期位置の交差点を認識した後 (図 7 (a)), まず, 同図 (i) から右に延びる骨格線上に分岐点が抽出され, 新たな解釈が派生する。次に左に延びる骨格線上に分岐点が抽出され, 新たに解釈が派生する。ここまで注目節点は変化しない。最後に上に延びる骨格線上に分岐点が抽出され, さらに新たな解釈が派生し, この時点で注目節点が B 節点に移る (図 7 (b)), ならびに図 8 (b) の B 節点。図 9 (a) の 58 回目の更新時。探索は継続し図 7 (b) の (ii) から左, 右, 上に延びる骨格

線上にこの順序で分岐点が抽出され、それにもなつて解釈が順次派生する。その後、注目節点が C 節点に移り図 7(c) の経路が指示に沿った経路の一部であると判断する。これは図 9(a) のグラフで 129 回目の更新時に対応する。図 7(c) の破線の矩形部分付近の追加パターンを観測した後、さらに注目節点が D 節点に移り図 7(d) の経路が指示に沿った経路の一部であると判断する。図 7(d) の破線の矩形部分付近の追加パターンを観測した後、注目節点は E 節点に移り図 7(e) の経路を指示された経路と見なし、探索を終了する。

4.5.2 動作例 2

次に、推定値関数の違いによる動作の変化を調べるために、経験的に推定値関数を定めた動作例を示す。長さ n の仕様に対して、部分パターンから長さ x の解釈が得られている場合、図 4 の環境においては、残余パターンの解釈と仕様の残りの部分との一致度は $0.8 \times (n - x)$ 未満となることが実験を通して明らかになった。動作例 2 ではこの事実を反映した推定値関数 $0.8 \times (n - x)$ を用いて、次式の評価関数を構成した。

$$f(P_x^y) = \sum_{i=1}^x k_i(p_i) + 0.8 \times (n - x) \quad (21)$$

図 10(b) に至るまでの動作は、動作例 1 の場合と同様である。これ以降、動作例 1 の場合は図 7(c), (d) のように指示された経路以外の経路に対応する節点に注目節点に移ったことに対し、動作例 2 では図 10(c) 以降、指示された経路に対応する節点を注目し続けて探索を終了した。

4.6 考 察

いずれの動作例においても、最適解を発見して探索を終了した。部分パターンの更新にもない節点の評価が変動すること、またその変動に応じて注目節点が切り替わることが確認された。

2つの動作例からは、推定値関数により予測される一致度の重要性が示唆される。動作例 1 の推定値関数は、仕様と完全に一致する解釈が残余パターンから得られることを仮定していることに相当する。そのため、実際に残余パターンの一部分が観測され、仕様とあまり一致しないことが明らかになるまで、どの節点にも高い評価が与えられる。この結果、最適解に至る経路上の節点でなくとも注目節点となる(図 8(c), (d))。しかし、動作例 2 の推定値関数のように、残余パターンの解釈と仕様と一致度について、より現実的な評価を仮定すると、最適解に至る節点を主たる注目節点とした探索が可能であると考えられる(図 11(b), (c))。

これは従来の A* アルゴリズムにおいて、評価の推定値が真の評価値に近いほど、途中で展開される節点の数が減少すること¹⁰⁾に類似する特徴であると推察される。

5. ま と め

指示に基づく問題解決の手法が、所与の仕様と整合するパターンを発見する処理として定式化された。具体的な処理は、パターンの部分的な観測と認識、仕様との照合をともなう木探索として実現された。提案手法は、

- パターンの実際に観測された部分の解釈と仕様との一致度を評価する照合関数
- 実際の一致度を下回ることなく、パターンの未観測の部分の解釈と仕様との一致度を予測する推定値関数

が存在するという条件の下で、仕様と最も整合するパターンの発見を保証している。このことは作業認識に誤りが生じて、過去に遡って動作を修正し、最終的には問題解決を完了することに対応する。

しかし動作の修正は、必ずしも実際の作業認識の誤りに起因するわけではなく、誤りの可能性を探るために行われることもある。このような動作の修正を減少させるには、作業環境に固有の特徴を反映させるなどの調節を推定値関数に加え、作業の進行状況に対する予測能力を向上させる必要があろう。

本稿で議論した木探索の過程は、情報が不十分な状態での一種の思考プロセスのモデル化としてとらえることができる。この場合、照合関数は思考過程における中間的な結論の正しさを表現し、推定値関数は経験則の適用の適切さを表現する。これらに基づく節点の評価の変動は、思考過程で生じる一種の迷いを表現していると考えられる。本稿で示した定理が最終的に最適解の発見を保証していることは、思考の収束の保証に対応するものと見なせるであろう。

なお提案手法では、仕様に整合するパターンが複数存在する場合に、仕様の作成者の意図とは異なるパターンを発見して終了する可能性がある。したがって、類似するパターンの存在が既知である場合には、拘束力の強い仕様を用いることや、評価関数を修正する必要があると考えている。本稿では、未知環境でのナビゲーションへの応用を試みた。対象とする問題にやや特殊性が見られるものの、この問題の範疇においては、提案手法は比較的一般性の高いものであると考えられる。今後は、ナビゲーション以外の応用可能性を探っていきたい。

参 考 文 献

- 1) 岡田豊史, 開 一夫, 安西祐一郎: ロボットコマンド学習システム Acorn-II とその評価, 人工知能学会誌, Vol.9, No.6, pp.882-889 (1994).
- 2) Kuniyoshi, K., Inaba, M. and Inoue, H.: Learning by showing, *Robotics and Automation*, Vol.10, No.6, pp.799-822 (1994).
- 3) 池内克史, カン, S.B.: 視覚によるハンドの教示, 日本ロボット学会誌, Vol.13, No.5, pp.599-602 (1995).
- 4) 白井良明: 記述と照合, 電子情報通信学会誌, Vol.71, No.11, pp.1142-1148 (1988).
- 5) Nilsson, N.J.: *Problem-Solving Method in Artificial Intelligence*, McGraw-Hill (1971).
- 6) Russell, S.J. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, Prentice Hall (1995).
- 7) 白井良明: 人工知能の理論, コロナ社 (1992).
- 8) Latombe, J.C.: *Robot Motion Planning*, Kluwer Academic Publishers (1991).
- 9) 長谷川純一, 奥水大和, 中山 晶, 横井茂樹: 画像処理の基本技法〈技法入門編〉, 技術評論社 (1986).
- 10) 上野晴樹, 石塚 満: 知識の表現と利用, オーム社 (1987).

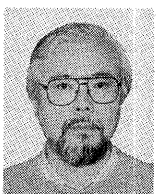
(平成 10 年 8 月 12 日受付)

(平成 11 年 5 月 7 日採録)



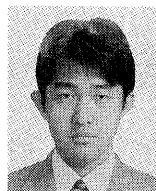
芦田 昌也 (正会員)

1966 年生. 1990 年大阪大学工学部通信工学科卒業. 1996 年同大学院通信工学専攻博士課程退学. 現在和歌山大学経済学部助教授. 空間理解に関する研究に従事. 電気学会, 電子情報通信学会, 日本ロボット学会各会員.



竹内 昭浩 (正会員)

1947 年生. 1969 年大阪大学基礎工学部電気工学科卒業. 1974 年同大学院博士課程修了. 同年和歌山大学経済学部講師. 同助教授を経て 1985 年より教授. 工学博士. 学習理論, 意思決定論に関する研究に従事. 電子情報通信学会, 人工知能学会, 社会経済システム学会各会員.



柴田 史久 (正会員)

1972 年生. 1996 年大阪大学大学院基礎工学研究科情報工学専攻博士前期課程修了. 1999 年同研究科博士後期課程修了. 博士 (工学). 同年同大学産業科学研究所助手. 移動ロボットのナビゲーションおよび学習に関する研究に従事. 日本ロボット学会, 計測自動制御学会各会員.



角所 考 (正会員)

1964 年生. 1988 年名古屋大学工学部電気工学科卒業. 1993 年大阪大学大学院博士課程修了. 博士 (工学). 1992~1994 年日本学術振興会特別研究員. 1993~1994 年スタンフォード大学ロボティクス研究所客員研究員. 大阪大学産業科学研究所助手を経て, 1997 年より京都大学総合情報メディアセンター助教授. 視覚メディア理解に関する研究に従事. 第 41 回情報処理学会奨励賞受賞. IEEE, ACM, 電子情報通信学会各会員.



北橋 忠宏 (正会員)

1939 年生. 1962 年大阪大学工学部通信工学科卒業. 1968 年同大学院博士課程修了. 同年大阪大学基礎工学部助手. 同助教授, 豊橋技術科学大学助教授, 教授を経て, 1986 年大阪大学産業科学研究所教授. 工学博士. 3次元物体認識のための視覚システム, 自然言語処理, メディア処理, 学習・推論機構に関する研究に従事. IEEE, 電子情報通信学会, 人工知能学会, 日本認知科学会各会員.