

エージェントの信念更新に関する 理由説明機能の作成について

奈良女子大学 理学部情報科学科 4 回生 荒添 理央

平成 21 年 2 月 16 日

概要

人は日常生活の中で、複数の選択肢のうち、どれか一つを選択しなければならないという場面に数多く遭遇しているだろう。このとき、その人がどの選択肢を選ぶのかというのは、目標達成に向けた、その人自身のさまざまな信念を考慮し、最終的な選択を下すことになる。自律的エージェントについても同様で、エージェントの行動選択は信念に基づいて行われる。そのため、信念を適切に維持することは重要な課題である。

エージェントの信念変更に特化した、効率の良いアルゴリズムとして、Alechia らが提案したものがある。そこで我々は、このアルゴリズムの実装を与え、その有効性を示すと共に、信念更新の理由を説明できるような機構の実現を行った。本論文ではそのうち信念についての導入と理由説明機構の実現について述べる。

1 はじめに

人は自分の信念を持ち、状況の変化に動的に対応し意志決定をすることで、目標達成に向けた一貫的な行動を取ることができる。同様の機構で目的を達成するエージェントの実現を考える場合、新しい情報が入ってきたときに、どのように無矛盾性を保って信念を維持するかという問題は信念変更において非常に重要であるが、一般に難しいものとされている。Alechina らは、エージェントのプランライブラリなどのように信念が有向グラフで表現されている場合に適用可能な、効率の良い信念変更アルゴリズムを提案した [2]。しかし、[2] では抽象レベルのアルゴリズムを与えるにとどまり、具体的な実装は示されていない。また、信念の変更方法が複数ある場合、そのうち結果として一つが選ばれたという理由を説明する機構が [2] にはない。そこで我々は、[2] で提案されている信念アルゴリズムの実装を行い、実例によってその有効性を示すと共に、信念更新の理由を説明できるような機構の実現を行った。なお、本研究は本学 4 回生の鈴木との共同研究であり、そのうち信念変更アルゴリズムの実装とその実例については [4] にて述べる。

本論文の構成を示す。2 節では研究の動機と背景について述べ、続いて 3 節で理由説明機構の実現について述べ、実例によってその有効性を示す。最後にまとめと今後の課題を述べる。

2 動機と背景

2.1 信念とは

信念とは人間が行動するときに誰もが持つ根本となる部分であり、従って、人間と同様の行為決定を行う合理的エージェントの実現にあたっても必要な概念である。合理的エージェントがある目標達成に向けて、矛盾しない一貫した行動をとるには信念を更新し、いかに無矛盾に維持できるか

ということは人工知能分野で重要な研究分野となっている。ここでいう‘信念’とは「信じている知識」であり、それは本人の勘違いも含めて認識していること、すなわち「不確かな知識」を指す。‘知識’とは異なるもので、個人の認識であるかで使い分ける。すなわち、‘信念’とは特定の人々の認識の世界のみで真と仮定されること、よって変更が可能なことである。

2.2 信念の変更

2.2.1 信念変更について

信念変更とはデータベースとしての信念の動的な変化を観察することに目的を置いており、今の信念状態に新しい情報を付け加えるとき、新しい信念状態へどのように書き換えが起きたのかに着目する。また、信念を変更するやり方が一意に定まらない場合もある。例えば、エージェントが $p \supset q$ および p 、 q という信念を持っていて、 q を信念から除かねばならない場合、 q を除くだけでは $p \supset q$ と p から q が出てしまうので、 p と $p \supset q$ のいずれかも除かねばならない。しかし、どちらを取り除くべきかはこれだけでは一意に決まらない。このような場合に信念変更が満たすべき、いくつかの合理的と思われる指針が統合性制約として定義されており、それに従うものとしてある。その制約の中で、“信念の変更に伴う変更は最小にとどめるべきである”、つまり、信念変更は情報は無料ではないという暗黙の前提の上で行うものとされており、いま持っている情報をできる限り保持したいという考え (情報の経済性) を持って行われるものである。

2.2.2 信念変更の種類

信念の変更方法は一般に複数あるとされ、それは以下の3つに大別される。信念状態を K とするとき、

拡張 ある命題 A が新しい情報として得られたとき、それを単に受け入れる。 $K + A$ と表記され、矛盾する命題を含む可能性がある。

リヴィジョン 新しい情報 A を受け入れる代わりに K 中の命題の一部を削除し、信念全体に矛盾がないようにする方法で、 $K * A$ と表記される。

コントラクション 新しい情報を付加することなしに、 K 中の一部の命題を削除し、一貫性を保つ方法で、 $K \div A$ と表記される。

これらの操作は、信念集合に対してではなく、信念ベースに対して行うものである。

信念ベースとは、すべての論理的帰結を含んだ文の集合である信念集合の一部であり、原理・原則的な命題の集まりのことを意味する。信念集合による信念状態の表現では、一般に公理となる原理・原則的な命題から演繹のプロセスを経て得られる論理的帰結として導かれる文までを含んでしまい、この区別がなされていない。信念ベースは、そういった原理・原則的な命題と演繹によってもたらされた新しい命題とを区別する考え方を持つものである。

2.2.3 AGM 原理

上記でも述べたように、信念変更の際には情報の経済性なる制約がある。この情報の経済性を定量的に実現するのは一般に困難であるが、この問題意識に立って定めた、信念状態の表現に信念集

合を用いて合理的な基本原理がある。これが AGM 基本原理である。信念集合におけるコントラクション (AGM 基本原理) として

$$(K \dot{-} 1) K \dot{-} A = C_n(K \dot{-} A)$$

$$(K \dot{-} 2) K \dot{-} A \subseteq K$$

$$(K \dot{-} 3) \text{ If } A \notin K, \text{ then } K \dot{-} A = K$$

$$(K \dot{-} 4) \text{ If not } \vdash A, \text{ then } A \notin K \dot{-} A$$

$$(K \dot{-} 5) \text{ If } A \in K, \text{ then } K \subseteq (K \dot{-} A) + A$$

$$(K \dot{-} 6) \text{ If } C_n(A) = C_n(B), \text{ then } K \dot{-} A = K \dot{-} B$$

が定義されている。(K \dot{-} 1) と (K \dot{-} 2) については $K \dot{-} A$ は信念集合であり、それは K と包含関係にあるということを示している。(K \dot{-} 3) では情報の経済性を定義しており、(K \dot{-} 4) からは A が恒偽であるとき、 A をコントラクションした結果は A を含まないということが言える。(K \dot{-} 5) においては情報の経済性を保証する意味で定義されているが、制約の厳しさからこれを基本原理として認めない立場もある。(K \dot{-} 6) は中身が命題として同値なものであれば、コントラクションの結果も同じという等価性が示されている。

これらに対応して、信念集合におけるリヴィジョンについても同様の原理が存在する。このリヴィジョンとコントラクションは独立しておらず、互いに導出可能な関係にある。 $K * A$ とは最初に K から $\neg A$ をコントラクションしておいて、後から A を含むように拡張したものであると ' $\dot{-}$ ' を使って以下のように定義することができる。

$$K * A \equiv (K \dot{-} \neg A) + A$$

このようにして定義されたリヴィジョンは Levi 同一化 ' $R(\dot{-})$ ' と呼ばれ、右辺の ' $\dot{-}$ ' がコントラクションの AGM 基本原理 $K \dot{-} 1$ から $K \dot{-} 4$ および $K \dot{-} 6$ を満たすとき、リヴィジョンのそれも満たす。また、逆にリヴィジョンを満たす基本原理からコントラクションを作り出すこともでき、これを Harper 同一化 ' $C(\dot{-})$ ' と呼ぶ。

2.2.4 信念変更アルゴリズムの実装と実例

本来、信念変更においては、前述で取り上げた AGM 基本原理をそのままプログラム化することが望ましい。しかし、これを満たす信念変更の一般的な実現は、NP 完全な問題となることが知られており、膨大な計算量と時間が必要となってしまうため、信念変更は難しいものとされている。[2] では、信念を有向グラフで表現できる場合に有効な信念変更アルゴリズムで、(K \dot{-} 1) から (K \dot{-} 4)、および (K \dot{-} 6) を満たし、かつ効率の良いアルゴリズムが提案されている。エージェントの信念やプランライブラリはほとんどの場合、有向グラフで表現できるため、このアルゴリズムはエージェントの信念更新には多くの場合有効である。従って、信念を元に行為決定を行うエージェントの実現を考えると、このアルゴリズムを実装することは有効であると考えられる。

信念変更アルゴリズムのコントラクションの例を簡単に説明する。このグラフ内のノードは、信念と正当化を表す。

上図のような有向グラフが与えられ、 B をコントラクションしたい場合、まず、 B から出ているエッジ、ノードを削除する。そして、次に B を導出する全てのノード、そこから B に向かって出

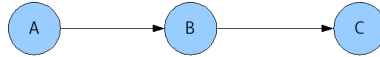


図 1: 信念変更コントラクションアルゴリズムについて

ているエッジを削除する。最後に B 自身を削除することで、B をコントラクションできる。これは $(K-1)$ から $(K-4)$ 、および $(K-6)$ を満たすアルゴリズムになっており、実行することでコントラクションの機能を持つ。また、実行時間として、 k を支持リスト内にある信念の個数の最大数、 r をプランの個数、 n を信念ベース内のリテラルの個数とすると、このアルゴリズムの計算量は、 $O(kr+n)$ で表される。これは、一般的な信念変更の問題が NP 完全であるのに比べ、はるかに効率の良いものである。

[2] において、信念変更アルゴリズムは提案されているが、実際に実装は与えられていなかった。そこで我々はこの信念変更アルゴリズムを実装し [4]、実例を作成することでその有効性を示した。

2.3 理由説明機能の必要性

しかし、従来は [2] で提案された信念変更アルゴリズムには、最終的に更新された結果を出力する機能しかなく、信念の変更方法が複数ある場合、そのうち一つが選ばれた理由を説明する機構が [2] にはなかった。

しかし、実際には信念更新の結果だけが欲しい場合ばかりでなく、その更新方法が選ばれた理由を把握できることが望ましい場合がある。特に、エージェントと人間の相互作用においては、エージェントがある行動を取った理由が人間に理解できることは望ましい。

例えば、エージェントが行う信念変更の結果、 p という信念を残す可能性と q という信念を残す可能性があり、前者の場合 a という行動、後者の場合 b という行動を取るとする。いま、エージェントが a という行動を取った時、その理由が人間には分からない。エージェントが p という信念を残した理由が分かれば、その行動の理由が明確になる。

そこで我々は、この信念変更アルゴリズムに、理由説明機能を追加した。例のように複数の選択肢がある場合に、どうしてその結論に至ったのか、結果だけでなく、その行動に至った経緯を信念更新という観点から理由・説明を提示できる行動説明につながるものを作成することにした。

3 理由説明機能の作成

3.1 AgentSpeak と Jason

AgentSpeak は論理プログラミングに基づいたエージェント記述言語であり、自律エージェントのモデルとして近年広く用いられている BDI エージェントをプログラム化するための抽象的なフレームワークを提供するものである。Jason[3] とは AgentSpeak の拡張言語のインタプリタであり、マルチエージェントシステムである。我々は、上に述べた信念変更アルゴリズムを Jason 上で実装したため、この理由説明機能も Jason 上で実装した。

3.2 実装方法

3.2.1 データの管理

理由説明表示に必要とされる根拠となるデータ、一つの信念に関する情報は下図のように信念ベースにてまとめて管理する。

信念名 (信念 A, [信念 B, ...])

第一引数 (信念 A) の信念を派生させる材料となるものをリスト化し、このように記述することで、‘リストから派生される信念として、第一引数のものがある’ということが定義できる。これらの信念ベースに与えておく情報として、有向グラフを形成するノード・エッジ情報、つまり、あるノードが別のノードを導出できるエッジを持つとき、そのノードやエッジ一つ一つに番号を与え、ノード間の関係などをデータ化したもの等が挙げられる。

下の例で説明すると、ノード r1 から出ているエッジ ‘2’ よりノード r2 が導出され、そしてそのエッジ ‘2’ の信憑度として 0.8 という値が与えられているということになる。

node_edge(r2, [r1, 2, 0.8])

このように信念ベースにおいて一つの信念に関する情報をまとめて管理、記述しておくことで、エージェント間でのやりとりの際に行動説明につながるデータの受渡しがしやすくなる。表示したい際には、この信念のデータから必要なものだけを取り出し、出力すれば良いので、理由説明表示しやすいという利点がある。

3.2.2 print 関数での出力

Jason 上では、システムが動いているコンソールにメッセージを出力する内部関数として print 関数が定義されている。本機能では、そこで、実装した信念変更アルゴリズムにおいて、いくつかの候補のうち一つが特定される過程で、この関数を用いてその選択の説明となるデータを出力することで、変更が行われた理由を明示することができる。また、実装する際、必要な過程でこの print 関数を入れるが、プログラム内のどこに挿入するかということが、この機能作成において重要となっている。

例えば、

```
-+tem_route2("r1-r2", D1+L1);
?tem_route2("r1-r2", D2);
.print("tem_distance2 = ", D2);
```

このように記述することで、信念の追加・削除が行われ、信念更新された後、理由説明となるデータを print 関数にて表示させている。これにより、信念変更アルゴリズムの実装において、いくつかの候補となる選択肢から一つが特定されるとき、その根拠となるデータが提示できることから、理由説明表示が可能となる。

3.3 例題

3.3.1 高速道路の経路探索

信念更新の結果によってとりうる行動が左右されるような応用の典型として、経路のいずれかが不通になる場合の代替経路選択の問題が考えられる。そこで我々は、実現した機能の有効性を示す例として、高速道路経路探索 (大阪～名古屋) の例を取り上げる。

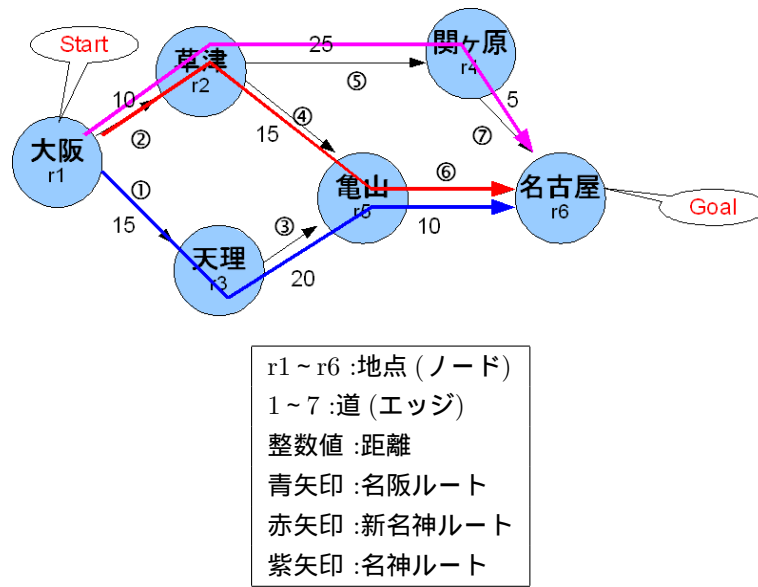


図 2: 高速道路経路探索の例

図 2 は、大阪～名古屋の経路を有向グラフで表したものである。このグラフを形成する信念は

```
node_edge(r2, [r1, 2, 0.8])
node_edge(r3, [r1, 1, 0.8])
node_edge(r4, [r2, 5, 0.8])
node_edge(r5, [r2, 4, 0.8])
node_edge(r5, [r3, 3, 0.8])
node_edge(r6, [r4, 7, 0.8])
node_edge(r6, [r5, 6, 0.8])
```

と表される。

これらを信念ベースに与えることで、図のようなグラフを生成する。地点間の距離はあくまでも仮のものとし、それぞれの道 (エッジ) には簡単のため、番号を与えており、操作としては、スタート地点の大阪からゴールの名古屋まで今のエージェントの信念状態で最も最適である最短経路を探索することにする。いま、大阪 - 名古屋間ルートとして名阪ルート、名神ルート、新名神ルートの 3 つのルートを設定しており、デフォルトでは最短経路を探索することになるが、ここではもしその探索した経路が通行不可能であるという信念を受け取った際、エージェントがどのように信念更

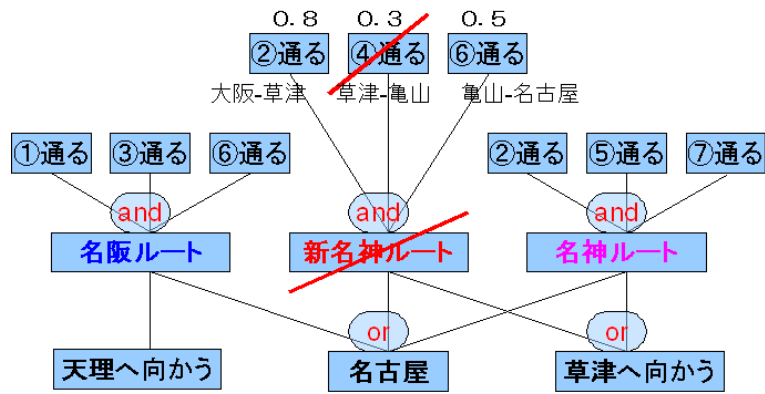


図 3: 例題における信念変更の考え方

新を行うかという点に着目する。通行不可能なルートの情報から、通れない道を特定し、通行可能なルートから最短で行けるものとして1つのルートを選択するまでの考え方を図3に示す。

今回の例では、この図のようにノードとエッジを and と or で関係づけて考えることができ、信念削除の方法として前述のコントラクション版信念変更アルゴリズムを用いることができる。

例として、最短ルートである新名神ルートが通れないという情報を受け取ったとする。そこで、図3の and の新名神ルートが通れるという信念を削除(コントラクション)する場合、2、4、6の道が通れるという情報を全て残しておく、これらから新名神ルートが通れるという信念が再び出てしまうため、'2の道が通れる'という信念、'4の道が通れる'という信念、'6の道が通れる'という信念のいずれかを削除せねばならない。ここで、どの信念を削除するかは、信念ベースに予めデータとして与えてある、信憑度を用いる。

[2]では個々のリテラルに評価値を決めておき、その比較で選択しているが、今回我々は各信念に信憑度を与えることでそれを実装している。従ってこの場合、信憑度が最も低い0.3である、'4の道が通れる'というデータが入った信念がベースから削除される。その後、この新名神ルート自身の信念も4の道が通れるという信念が削除されたことから、andの関係が成り立たなくなり、削除(コントラクション)される。そして、スタートから草津方面へ向かう、もう一方のルートである名神ルートを見つけることになる。しかし、名阪ルートもゴールへ行ける道として残っているので、その場合には2つの全体距離を比較し、最短でゴールまで行けるルートが最適な信念であると選択され、更新されることになる。

また、2、4、6の3つの道の信憑度判定において、2の道の信憑度が最も低いと判定され、'2の道が通れる'という信念が削除された場合を考える。この場合は and の関係より、新名神ルートだけでなく、名神ルートも通行不可能であると判断され、これらのルート情報が入った信念は削除される。すると、'草津へ向かう'という信念との or の関係も成立しなくなってしまうため、この信念も削除され、最終的には信念としてまだ残っている名阪ルートがこのときの最適な信念として、更新されることになる。

3.3.2 実験結果

以下、最短ルートが通れなくなった場合に、通れない道を特定し、新たにルートを探索している部分を抜粋したものである。

```

[ag1] -----route r1→r6 search-----
[ag1] r6_route1("r1-r2-r5-r6",35)
[ag1] r6_route2("r1-r2-r4-r6",40)
[ag1] r6_route3("r1-r3-r5-r6",45)
[ag1] (Goal)=(1/3/6)\{(2/4/6)\(2/5/7)\}
[ag1] no_route : r1-r2-r5-r6
[ag1] route2-4-6 →route_value[0.8,0.3,0.5]
[ag1] min 0.3
[ag1] remove : n_e(r5,[r2,4,0,3])
[ag1] best_route("r1-r2-r4-r6",40)
[ag1] contraction belief! r6_route2("r1-r2-r5-r6",35)
[ag1] ***passable route***
[ag1] r1-r3-r5-r6 : 45
[ag1] r1-r2-r4-r6 : 40
[ag1] distance compare...
[ag1] min([45,40],40)
[ag1] best_route("r1-r2-r4-r6",40)
[ag2] best route : r1-r2-r4-r6 ,dis40
[ag2] route search end!! bye.

```

図 4: 例題による実験結果

表示として、まずスタート(大阪)からゴール(名古屋)までの考えられるルートの表示をし、そのルートと道の関係を示している。

そして、最短ルートである‘新名神ルートが通れる’との信念を削除するのに伴い、‘4の道が通れる’という信念が削除された。その結果、名神ルートと名阪ルートが通れるという信念は残ったので、名神ルートが選択された。

これに比べ、もし‘2の道が通れる’という信念が削除された場合、‘名阪ルートが通れる’という信念だけが残るので、名阪ルートが選択される。このとき、名阪ルートを選択したのはなぜかという問いに対して我々の実現した機能を使えば、それは‘2の道が通れない’と判断したからだという説明ができる。

このように、エージェントがなぜ複数の経路のうち、その経路を選んだのかという理由・説明をこの理由説明機能で表示することが可能となった。これにより、エージェントの行動の根拠が明確になり、人間にとっての理解がしやすくなった。

4 まとめ

本研究では、信念変更の重要性、信念維持の難しさから提案されている信念変更アルゴリズムを実装し、そこで発生した問題点から理由説明機能の作成を行った。またそれぞれの実例も作成し、有効性を検証、そしてエージェントの行動の理由説明表示を可能としたことで、第三者から見た場合にも、どのような過程で信念更新が行われているのかが分かるようになった。また、人間との相互作用において、エージェントの取った行動を人間が理解できるようになることにつながるといえる。

今後の課題としては、大量の情報量でも対応できるように、そしてこの機能を実際に例として作られたプログラムに組み込んで、使えるように改良することが挙げられる。また、将来的にはこの機能をエージェント同士の対話機能に活用するなどの応用が考えられる。

謝辞

本研究を遂行するにあたり、いつも親身に御指導して下さった新出尚之准教授、鴨浩靖助教に、深く感謝し、厚く御礼申し上げます。また、丁寧な助言を与えて下さった藤田恵先輩、野口真理子先輩に感謝の意を表します。

参考文献

- [1] 東条敏, 「知の科学『言語・知識・信念の論理』」, オーム社, 2006年.
- [2] Natasha Alechina, Rafael H. Bordini, Jomi F. Hubner, Mark Jago, and Brian Logan, 「Automating Belief Revision for AgentSpeak」, 2006.
- [3] Rafael H. Bordini, Jomi Fred Hudner, Michael Woldridge, 「Programming Multi-Agent Systems in AgentSpeak using Jason」, WILEY, 2007年.
- [4] 鈴木恵里, 「エージェントの信念変更アルゴリズムの実装とその実例」, 奈良女子大学理学部情報科学科 2008年度卒業論文, 2009.