

スタックを用いた議論プロトコルの実現法と議論評価法に関する研究

Control of Arguments with Stacks and Evaluation Arguments

登坂 大悟*¹ 澤村 一*²
Daigo Tosaka Hajime Sawamura

*¹新潟大学大学院自然科学研究科
Graduate School of Science and Technology, Niigata University

*²新潟大学工学部情報工学科
Dept. of Information Engineering, Faculty of Engineering, Niigata University

Argumentation is paid attentions as a way of interaction, communication, conflict resolution and attaining a consensus for agents. So many argument-based agent systems have been proposed.

In this paper, we propose an argument-based agent system with two mechanisms to approve our old system (such as [Maeda 01]): The first, for comparison conflicting arguments, we introduce new argument evaluation rule which computes the reliability of an argument with respect to conflicting conclusion with taking account of agents' view. The second, we introduce the mechanism with a stack to control the flow of complex argumentation.

1. はじめに

近年、エージェントによる人間の意志決定支援の手段としてコンピュータを用いた議論が着目されており、議論によって様々な問題の解決を図ることが期待されている。

現在では問題解決のための様々な議論フレームワークが提案されている。その一つとして [Maeda 01] で挙げられる契約ネットプロトコルを用いた議論エージェントシステムがある。このシステムの問題点として、提案者の数的不利 (1 対多) と、議論評価式の欠陥が挙げられる。本稿ではその解決法として、スタックを用いた議論の流れの制御と新しい議論評価法を述べ、新たなシステムを提案する。

なお、本稿において“論証”とはエージェントが知識ベースから推論エンジンを用いて構成した推論の列のことを示し、エージェントが論証同士の比較や論証に対し反論をぶつけるといった過程を“議論する”と呼ぶ。

2. 議論するエージェントシステム

この節では、[Maeda 01] の議論するエージェントシステムの概要を述べ、問題点を指摘する。

2.1 議論するエージェントシステムの概要

このシステムは、議題の提案と賛成論証の提出を行う“提案者”1人、議論への参加と反対論証の提出を行う“参加者”多数、論証の優劣判定と議論の仲介を行う“議長”1人、の3種類のエージェントから構成される。各エージェントは知識ベースと論証構築のための推論エンジンを持つ。

以下にシステムの主な流れを示す。

- (1) 提案者は議題に関して作成した論証を議長に送る。
- (2) 議長は送られてきた論証を参加者に送り、反論を募る。
- (3) 参加者は送られてきた論証に対して作成した反論を議長に送る。
- (4) 議長は送られてきた反論同士を比較して、最も強い反論を選択する。そして、提案者の論証と最も強い反論を比較し、判定を行う。提案者の議論が反論に勝った場合、提

案者の勝ちでシステムは終了する。反論が提案者の論証に勝った場合、議長は提案者の論証に対して勝った反論を提案者に提示し、さらなる反論を募る。提案者は提示された論証に対する反論を作成し、議長に送る。

- (5) 議長は参加者の反論と送られてきた提案者の論証を比較し判定を行う。提案者の論証が勝った場合、議長はその論証の反論を募るため参加者に対して提案者の論証を送り、ステップ (3) へ。提案者の論証が負けた場合や、提案者が論証を作成できなかった場合、議長は参加者に対し提案者の論証を補強するサポート論証を募る。
- (6) 参加者は議長から求められたサポート論証を作成し、議長に送信する。
- (7) 議長は、送られてきたサポート論証と参加者の反論を比較する。サポート論証が負けた場合は、提案者の負けでシステムは終了する。サポート論証が勝った場合、提案者はサポートされて勝ったものとなりシステムが終了する。

システムの詳細は、[Maeda 01] を参照されたい。

2.2 議論するエージェントシステムの問題点と解決法

上で述べたシステムは、提案者 (1人) に対し参加者 (多数) なので、提案者は数的不利に立たされていた。この解決策として、提案者が自分の論証のサポートとなる論証を参加者から募ることができた。我々が新しく提案するシステムでは、議題に関して賛成する複数の参加者 (提案者も含む) を賛成派エージェントとし、議題に関して反対する複数の参加者を反対派エージェントとする。賛成派、反対派の双方を多数とすることで議論の公平性を図る。この拡張によって複雑になる議論の制御のために、スタックを用いた議論の制御機構を導入する。また、サポート議論を募る必要がなくなったため、議論評価式を新しく定義し、反論の有効性を総合的に判断する方法を導入する。

3. 議論に関する諸定義

本論文で使用する定義を述べる。

3.1 規則

エージェントが持つ知識は規則の集合とする。

連絡先: 登坂 大悟, 新潟大学大学院自然科学研究科情報理工
棟 502, 新潟市五十嵐 2 の町 8050, Tel 025-262-7490, E-
mail:tosaka@cs.ie.niigata-u.ac.jp

表 1: もっともらしさ付き含意オペレータ

もっともらしさ付き含意オペレータ	Plausibility
$:-$	0
$<-$	1
$<<-$	2
$<<<-$	3
$<<<<-$	4
...	...

定義 1 (規則) 規則を次の形式で定義する.

$$L_0 \Leftarrow L_1 \wedge \dots \wedge L_n .$$

ただし, $L_i (0 \leq i \leq n)$ はリテラルであり, 各リテラルは原子命題 (A), または否定された原子命題 ($\neg A$) である. また, \wedge は連言 (*and*) を表す. \Leftarrow は $\{:-, <-, <<-, \dots\}$ の内の任意の 1 つを表し, これをもっともらしさ付き含意オペレータと呼ぶ. \square

もっともらしさ付き含意オペレータにはもっともらしさ (*Plausibility*) を表す値が設定されており, 値が小さい方がもっともらしい規則としている. それを表 1 に示す.

規則の左辺のリテラルを結論と呼び, 連言肢のリテラルを前提と呼ぶ, ただし, 前提は空でも良いとし, 前提が空の規則を事実節と呼ぶ.

3.2 論証

定義 2 (論証) 次の条件を満たす規則の列 $[r_1, \dots, r_n]$ を論証とする.

- (1) 規則 r_i の前提にある各リテラル L は, 規則 $r_k (1 \leq k < i \leq n)$ の結論に現れる.
- (2) 規則 r_i の結論 L は, 規則 $r_j (1 \leq i < j \leq n)$ の前提に現れる.
- (3) 1 つの論証に同じ結論を持った 2 つ以上の規則は含まれていない. \square

定義 3 (論証の結論) 論証 $A = [r_1, \dots, r_n]$ に関して, 各規則 r_i の結論を $concl_i(A)$ と表す. 論証 A の結論は A に含まれる各規則の結論の集合 $\{concl_1(A), \dots, concl_n(A)\}$ である. また特に, 論証 A の規則 r_n の結論 $concl_n(A)$ を論証 A の最終結論と呼ぶ. \square

定義 4 (部分論証) 全ての論証 A に対して, A' が A の部分列であり, A' が定義 2 を満たすとき, そしてそのときに限り, A' を A の部分論証とする. \square

定義 5 (論証木) 論証 $A = [r_1, \dots, r_n]$ の論証木を, 以下の条件を満たすリテラルの木として定義する.

- 規則 r_n の結論 $concl_n(A)$ を論証木の根とする.
- 各ノード $concl_i(A)$ の子は, r_i の前提に含まれるリテラルである.

また論証木の高さとは, 論証木の根から各葉までの距離の最大値である. \square

3.3 反論

定義 6 (反論) 論証 $A = [r_1, \dots, r_n]$ に対し, $\exists i, (1 \leq i \leq n)$ $concl_m(B) = \neg concl_i(A)$ を満たす論証 $B = [s_1, \dots, s_m]$ を反論と呼ぶ. またこの時の $(concl_i(A), concl_m(B))$ を論証 A と反論 B の対立する結論と呼ぶ. \square

4. 議論評価法

このシステムでは, 論証における結論の信憑性の値を *RP* (*Reliability Point*) と呼び, この値が大きいほど結論の信憑性が高いものとしている. 以下の 3 つの定義は, *RP* を計算するための評価基準の定義である.

4.1 信憑性の計算における 3 要素

定義 7 (証拠の多さ) 論証 A の結論 L に関する証拠の多さを次のような関数 $Evi(A, L)$ で表す. L を最終結論とする論証 A の部分論証 $A' = [r'_1, \dots, r'_m]$ の論証木の葉の数を $Evi(A, L)$ の値とする. \square

$Evi(A, L)$ の値が大きいほど証拠が多く, 結論の信憑性が高い.

定義 8 (簡潔さ) 論証 A の結論 L における簡潔さを次のような関数 $Dir(A, L)$ で表す. L を最終結論とする論証 A の部分論証 $A' = [r'_1, \dots, r'_m]$ の論証木の高さを $Dir(A, L)$ の値とする. \square

$Dir(A, L)$ の値が小さいほど簡潔な論証であり, 結論の信憑性が高い.

定義 9 (もっともらしさ) 論証 A の結論 L に関するもっともらしさを $Pla(A, L)$ で表す. $L = concl_l(A)$ なる規則 r_l を $L_0^i \Leftarrow L_1^i, \dots, L_k^i, \dots, L_l^i. (1 \leq k \leq l) (l$ は前提のリテラル数になる) と表し, もっともらしさ付き含意オペレータの値を $Pla_{op}(i)$ と表す. この時, 次の手順で $Pla(A, L)$ を求める.

- if $l = 0$ then $Pla(A, L_0^i) = Pla_{op}(i)$
- if $l = 1$ then $Pla(A, L_0^i) = \max(Pla_{op}(i), Pla(A, L_1^i))$
- if $l \geq 2$ then $Pla(A, L_0^i) = \frac{Pla_{op}(i) + \sum_{k=1}^l Pla(A, L_k^i)}{l+1}$ \square

$Pla(A, L)$ は値が小さいほどもっともらしい結論なので, 結論の信憑性が高い.

4.2 重みによる信憑性の計算

以上の 3 つの要素から, 信憑性の値 *RP* を計算する. 各エージェントは 3 つの評価基準に対する重みを持っており, エージェント α における, $(Evi(A, L), Dir(A, L), Pla(A, L))$ の各重みを $(w_E(\alpha), w_D(\alpha), w_P(\alpha))$ で表す. この重みは, エージェントの性格や価値観を特徴づける値を表す.

定義 10 (結論の信憑性) 3 つの評価基準とそれぞれに対する重みによって, エージェント α による, 論証 A の結論 L の信憑性 (*RP*) を次の式で定義する.

$$RP(A, L, \alpha) = \{Evi(A, L) \times w_E(\alpha) - Dir(A, L) \times w_D(\alpha) - Vag(A, L) \times w_V(\alpha)\} \times 100 \quad \square$$

4.3 信憑性の値を用いた議論比較

定義 11 (論証の勝敗) エージェント α が, 論証 $A = [r_1, \dots, r_n]$ と A に対する反論 $B = [s_1, \dots, s_m]$ を比較し, 対立する結論 $(L, \neg L)$ に関して, $RP(B, \neg L, \alpha)$ の値が $RP(A, L, \alpha)$ より大きければ, 反論 B の勝ちとし, 小さいかまたは同じなら, 論証 A の勝ちとする.

定義 12 (論破) 論証 B に対して反論 A があり, A と B の論証比較において, A が B に勝ったとき, またそのときに限り, A が B を論破していると呼ぶ. \square

定義 13 (正当化) 論証 A に関して、次の条件の内どちらかが満たされた場合、論証 A が正当化されたと呼ぶ。

- 論証比較において、 A を論破した反論がない場合。
- 論証 A を論破している全ての論証が、正当化された論証から論破された場合。 □

定義 14 (却下) 論証 A を論破している全ての論証の内、1つ以上の論証が正当化された場合、論証 A は却下される。 □

5. 反論選択法

我々のシステムでは、1人のエージェントは1つの論証に対して反論を複数構築でき、その複数の反論の中から最も効率的な反論を選ぶ。この節では複数の反論の中からどのようにして効率的な反論を選ぶのかについて述べる。

5.1 反論選択の基準の3要素

複数の反論の中から1つを選ぶために、反論について以下の3つの評価基準を定義する。ただし、反論の作成者はエージェント α とする。

定義 15 (RPの差) 論証 $B = [s_1, \dots, s_m]$ に対して、エージェント α が作成した反論を $A = [r_1, \dots, r_n]$ とする。論証 A の最終結論を L とする。この時、RPの差 (RP_{DIF}) を次の計算式で定義する。

$$RP_{DIF}(A, B, \alpha) = RP(A, L, \alpha) - RP(B, \neg L, \alpha) \quad \square$$

また、 $RP_{DIF} > 0$ を満たさない場合、反論 A は反論の選択候補の中から外れる。なぜなら、この条件を満たさない場合、エージェント α は反論 A が論証 B を論破するのに十分でないと考えからである。 RP_{DIF} の値が大きい反論ほど、論証 B に対してより強く反論していることになる。

定義 16 (RPの平均値) エージェント α が作成した反論 $A = [r_1, \dots, r_n]$ におけるRPの平均値 (RP_{AVE}) を次の計算式で定義する。

$$RP_{AVE}(A, \alpha) = \frac{\sum_{i=1}^n RP(A, concl_i(A), \alpha)}{n} \quad \square$$

$RP_{DIF} > 0$ という条件があるため、エージェント α が選択する反論の全結論のRPが高いほど、相手はエージェント α の論証に対する反論を作成することが難しくなる。そのため、エージェント α は、1つの反論に対して全結論のRPの平均をとり、この値が高い反論を選択する。

定義 17 (RPの最低値) エージェント α が作成した反論 $A = [r_1, \dots, r_n]$ におけるRPの最低値 (RP_{MIN}) を次の計算式で定義する。

$$RP_{MIN}(A, \alpha) = \min x_i$$

ただし、 $x_i = RP(A, concl_i(A), \alpha)$, ($1 \leq i \leq n$) である。 □

エージェント α の選択する論証にRPの低い結論が1つでもあると、相手はその結論に対する反論を作り易くなる。そのため、エージェント α は RP_{MIN} の値が大きい論証を選択する。

5.2 反論評価による選択

3つの反論選択基準から、反論の効率を表す値 EP (EffectPoint) を求める。各エージェントは、 RP_{DIF} , RP_{AVE} , RP_{MIN} の各評価値に対して独自の重みを持っており、エージェント α が持つ各重みを、それぞれ $w_{DIF}(\alpha)$, $w_{AVE}(\alpha)$, $w_{MIN}(\alpha)$ と表す。

定義 18 (反論効率値) エージェント α にとって、論証 B に対して作った反論 A の効率 EP を次のように定義する。

$$EP(A, B, \alpha) = RP_{DIF}(A, B, \alpha) \times w_{DIF}(\alpha) \\ + RP_{AVE}(A, \alpha) \times w_{AVE}(\alpha) \\ + RP_{MIN}(A, \alpha) \times w_{MIN}(\alpha) \quad \square$$

エージェント α が複数構築した反論から選ぶのは EP が最も大きい反論である。

6. スタックを用いたシステムの構築

この節では、我々が提案するシステムの大まかな流れとスタックの動作を説明する。

6.1 スタックによる議論の制御

システムでは、議論の流れの制御のために以下のようにスタックを用いる。

- (1) スタックには直前の論証を論破した論証を順次積み上げていく。
- (2) スタックに積み上げた論証は、定義 13,14 に従って、正当化または却下されることが確定した時点で、スタックから削除される。また、正当化・却下を調べる必要がなくなった論証も削除される。
- (3) スタックに積み上げられた論証が上の手続きによって全て処理され、最初にスタックに積まれた(エージェントから提出された)論証の正当化・却下が確定した時点で、システムは終了する。最初の論証が正当化されたなら提案者の勝ち、却下されたなら提案者の負けとなる。

6.2 システムの流れ

このシステムの動作は議論準備段階と論争段階の2つに分けることができる。議論準備段階では、議長は議論に参加するエージェントを募り、参加者に対し、議題に対するエージェントの立場(賛成派、反対派)を表明してもらう。以降、提案者と賛成派の立場をとったエージェントを総称して賛成派エージェント、反対派の立場をとったエージェントを総称して反対派エージェントと呼ぶ。

議論段階は2つのフェイズに分かれている。賛成派が反対派の論証に対し論破を試みるフェイズをメインフェイズと呼び、逆に、反対派が賛成派の論証に対し論破を試みるフェイズをカウンターフェイズと呼ぶ。システムはこの2つのフェイズを終了条件を満たすまで繰り返す。以下で2つのフェイズの流れを示す。

6.3 議論段階(メインフェイズ)

議論準備段階が終わり最初にメインフェイズに移行してきたとき、議長は提案者に議題に賛成する最初の論証を提出してもらう。議長は提出された論証を最初の論証としてスタックに積み、カウンターフェイズに移行する。

メインフェイズへの移行が2回目以降であるとき、スタックの一番上の論証は反対派の論証になっている。その反対派の論証を反論対象とよぶ。議長は反論対象に対する反論を賛成派エージェントに求める。賛成派エージェントは反論対象に対す

る反論を作り，5節で述べた方法で各自が最も効率的な反論を選択し，その反論を議長に提出する。

議長は反論対象と提出された全ての反論に対して論証比較を行う。反論が勝った場合，その反論は反論対象を論破しているものとして，勝った反論全てをスタックに積み，それらに対する反論を募るためカウンターフェイズに移行する。

賛成派から提出された全ての反論に対して反論対象が勝った場合，または反論が1つもなかった場合は，反論対象が正当化されたものとして，6.1節のスタックの動作に従ってスタックの削除が行われる。

6.4 議論段階 (カウンターフェイズ)

カウンターフェイズではスタックの一番上の論証は賛成派の論証になっており，議長はその賛成派の論証を反論対象とし，反論対象に対する反論を反対派エージェントに求める。反対派エージェントは反論対象に対する反論を作り，5節で述べた方法で各自が最も効率的な反論を選択し，その反論を議長に提出する。

議長は反論対象と提出された全ての反論に対して論証比較を行う。反論が勝った場合，その反論は反論対象を論破しているものとして，勝った反論全てをスタックに積み，それらに対する反論を募るためメインフェイズに移行する。

反対派から提出された全ての反論に対して反論対象が勝った場合，または反論が1つもなかった場合は，反論対象が正当化されたものとして，6.1節のスタックの動作に従ってスタックの削除が行われる。

6.5 議論の例とスタックの動作

この節では，我々が提案するシステムの動作例を示し，議論に応じてスタックがどのようにして変化して行くのかを示す。以下の流れと図1は括弧付き番号が対応しており，流れの中の鍵括弧はスタックの操作を示す。図1は，四角が論証，矢印が論証の論破の方向を示している。

- (1) 提案者から議長へ論証 P_1 が送られてくる。[Push P_1]
- (2) 論証 P_1 に対し，反対派エージェントは反論 O_2 を提出。 O_2 が P_1 を論破する。[Push O_2]
- (3) 論証 O_2 に対し，賛成派エージェントは反論 P_{3a}, P_{3b}, P_{3c} を提出。 P_{3a}, P_{3b}, P_{3c} が O_2 を論破する。[Push P_{3a}, P_{3b}, P_{3c}]
- (4) 論証 P_{3a} に対し，反対派エージェントが反論 O_{4a} で攻撃。 O_{4a} が P_{3a} を論破する。[Push O_{4a}]
- (5) 論証 O_{4a} に対し，賛成派エージェントが反論で攻撃することができなかったため， O_{4a} が正当化され，正当化された O_{4a} は P_{3a} を論破しているため， P_{3a} は却下される。[Pop from O_{4a} to P_{3a}]
- (6) 論証 P_{3b} に対し，反対派エージェントが反論 O_{4b}, O_{4c} で攻撃。 O_{4b} が P_{3b} を論破する。[Push O_{4b}]
- (7) 論証 O_{4b} に対し，賛成派エージェントが反論 P_5 で攻撃。 P_5 が O_{4b} を論破する。[Push P_5]
- (8) 論証 P_5 に対し，反対派エージェントが反論で攻撃， P_5 が反論全てに勝ったため， P_5 が正当化され， P_5 に論破されている O_{4b} は却下される。[Pop from P_5 to O_{4b}]
ここで定義13より，スタックの一番上の論証 P_{3b} を論破する論証がなくなるので，論証 P_{3b} は正当化され，正当化された P_{3b} は O_2 を論破しているため， O_2 は却下される。 O_2 を論破している論証 P_{3c} は， O_2 の却下により論破の対象が無くなるため，今後の議論において考慮する必要がなくなる。そのため P_{3c} は， P_{3b}, O_2 と共にスタックから削除される。[Pop from P_{3b} to O_2]

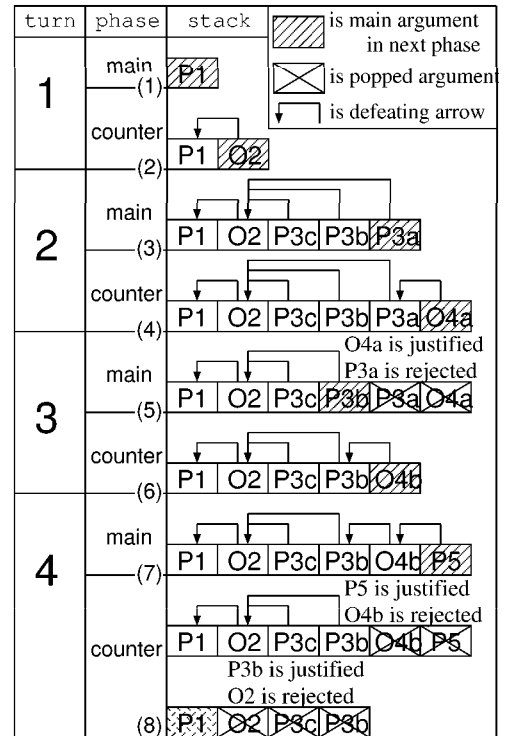


図 1: スタックの動作

論証 P_1 を論破する論証がなくなったので，論証 P_1 が正当化され，議論の終了条件を満たすので，システムが終了する。

7. まとめと今後の課題

新たな議論評価値 RP と反論効率値 EP を導入したことにより，重みによるエージェントの個性の表現に成功した。また，スタックを用いた制御法で，多対多のエージェントシステムを制御できた。今後の課題としては，議論終了時における知識の獲得や，エージェントの役割の自由化，等が挙げられる。

参考文献

- [Maeda 01] Hajime Sawamura, Shinya Maeda: An Argumentation-Based Model of Multi-Agent Systems, *Information Modeling and Knowledge Base XII*, pp.137-150 (2001).
- [Prakken 97] Henry Prakken: Dialectical proof theory for defeasible argumentation with defeasible priorities (preliminary report), *ModelAge Workshop*, pp.202-215, (1997).
- [Prakken & Sartor 98] Henry Prakken, Giovanni Sartor: Modelling Reasoning with Precedents in a Formal Dialogue Game, *Artificial Intelligence and Law 6:*, pp.231-287, (1998).