

## メッセージ通信を用いた分散型結論発見

岡本 孝之<sup>†,††</sup> 井上 克巳<sup>††</sup>

<sup>†</sup> 神戸大学大学院自然科学研究科 〒658-8501 神戸市灘区六甲台町 1-1

<sup>††</sup> 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: <sup>†</sup>034t212n@y03.kobe-u.ac.jp, <sup>††</sup>ki@nii.ac.jp

**あらまし** 知識が地理的に分散している状態において、個々が持つ情報を他人に知られたくないという理由から問題を解く際に知識を一つに集めることができない場合がある。本研究の目的はそのような状況においても情報を一つに集めず、個々が協力し合って問題を解決するというものである。そこで本研究では結論発見に対して完全な手法を2つ提案する。一つ目は Amir と McIlraith が行った定理証明の手法を拡張するものであり、2つ目はより協調的な問題解決手法である。実験により、2つの提案手法の比較・考察を行う。

**キーワード** メッセージ送信, コミュニケーション言語, 結論発見

## Distributed consequence finding with message communication

Takayuki OKAMOTO<sup>†,††</sup> and Katsumi INOUE<sup>††</sup>

<sup>†</sup> Graduate School of Science Technology, Kobe University

1-1 Rokkoudai-cho, Nada-ku, Kobe, 658-8501 Japan

<sup>††</sup> National Institute of Informatics Department 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan

E-mail: <sup>†</sup>034t212n@y03.kobe-u.ac.jp, <sup>††</sup>ki@nii.ac.jp

**Abstract** When knowledge is geographically distributed, individual information might not be able to be collected to one agent because it does not want to be known to others. We aim that individual agents cooperate each other to solve the problem without collecting information to one agent in such a situation. This paper proposes two complete algorithms to the consequence finding. The first one extends the technique of theorem proving from distributed knowledge bases by Amir and McIlraith. The second one is a more cooperative method than the first one. We compare these two methods with experiments.

**Key words** message passing, communication language, consequence finding

### 1. はじめに

定理証明や推論の研究において、さまざまな知識を知識ベースで構築することは大変興味深い問題である。この知識ベースは何百、何千もの公理から構成される巨大なものである。このような問題を扱う状況において2つの場合が考えられる。一つ目は公理集合が一箇所に集中している場合、そして2つ目は複数の公理集合がもともと地理的に分散している場合である。Amir と McIlraith は、前者のような問題を扱うアプローチとして一つの巨大な公理集合をより小さな複数の公理集合に分割する方法を挙げている [2]。巨大な問題をそのまま解いたのでは計算量が非常に大きくなってしまいう理由から Amir らは公理集合を分割し、効率化を図っているのである。

これに対し、本研究では後者の問題を扱う。例えば問題を解く際に自分の持つ情報すべてを相手に知られたくない状況があ

り、知識を一つに集めることができない場合がある。本研究ではそのような状況においても情報を一つに集めず、個々が協力し合って問題を解決するというものである。つまり、公理集合が分散された状態において分散されたまま問題を解くことを目的としている。

本研究ではこのような状況における結論発見 [4], [6] に対して完全な解法を2つ提案する。ここで完全とは、問題が分割されている場合にも分割されていない場合と同じ結論が得られることをいう。まず提案手法1として、Amir らが定理証明について行った手法 [2] を結論発見に拡張する。結論発見とはある与えられた公理集合に隠れている結論を発見することであり、自動演繹にとって興味深い問題である。また結論発見は定理証明のように与えられた式を導けるかを調べるだけでなく、ある特徴を持つ定理を発見するという点で定理証明とは異なる。提案手法1では公理集合間をコミュニケーション言語と呼ばれる言

語でつなぐ必要があり、公理集合とコミュニケーション言語のつながりをグラフと捉える。問題を解く際にはこのグラフを木構造に変換する。まず根の公理集合において結論発見を行い、得られた結論がコミュニケーション言語から構成されるような節であれば、子の公理集合へその結論を送信し、加える。この操作を葉の公理集合にたどり着くまで繰り返す。結論発見に対して完全であるためにはコミュニケーション言語の決定方法が重要であり、その決定方法を提案する。

次に提案手法 2 として Amir らの手法を用いることなく、より協調的に結論発見を行う完全なアルゴリズムを提案する。この手法はまずある節に注目し、その節と融合可能な節が他の公理集合に存在すれば、その公理集合に節を送信し、加える。そして節を加え終わった後に結論発見を行い、得られた結論と融合可能な節が他の公理集合に存在すれば結論を送信し、加えるという操作を繰り返すものである。この手法では、融合可能な節が他の公理集合に存在しなくなるか、送信しようとする節が他の公理集合の節に包摂されるという状況になるまでメッセージを送信し続ける。なお実験により、これら 2 つの提案手法の比較・考察を行う。

この論文は次のように構成されている。まず 2 章で基礎的な語句や Amir が行った定理証明について説明する。次に 3 章で各公理集合における推論手続きである SOL 導出について説明する。4 章では結論発見に対して完全な手法を 2 つ提案しする。5 章で実験を行い、6 章でまとめと今後の課題について述べる。

## 2. パーティションに基づいた定理証明

この章では基礎的な語句の説明やパーティションを用いて問題を解くためのメッセージ送信アルゴリズムについて説明する。

### 2.1 準備

公理集合を  $\mathcal{A}$  とし、 $\mathcal{A} = \bigcup_i \mathcal{A}_i$  とするとき、各公理集合  $\mathcal{A}_i$  をパーティションと呼ぶ。 $S(\mathcal{A}_i)$  は  $\mathcal{A}_i$  に出現する記号の集合 (非論理記号の集合) とし、 $\mathcal{L}(\mathcal{A}_i)$  はその式の集合 ( $S(\mathcal{A}_i)$  から構築される論理式の集合) とする。またパーティションからグラフを定義し、グラフ  $G = (V, E, l)$  とする。 $V$  はノードの集合であり、各ノード  $i$  は個々のパーティション  $\mathcal{A}_i$  と等しい。 $E$  は辺の集合である。 $\mathcal{A}_i$  と  $\mathcal{A}_j$  が共通の言語を持つとき、すなわち  $E = \{(i, j) \mid S(\mathcal{A}_i) \cap S(\mathcal{A}_j) \neq \emptyset\}$  となるような  $E$  が存在するとき、二つのノード  $i, j$  は辺  $(i, j)$  でつながっている。また  $l$  はラベルの集合である。辺  $(i, j)$  のラベルを  $L(i, j)$  と表し、 $\mathcal{A}_i$  と  $\mathcal{A}_j$  の間コミュニケーション言語と呼ぶ。 $L(i, j)$  の初期値は  $S(\mathcal{A}_i) \cap S(\mathcal{A}_j)$  である。初期値以外は以降で定義する。

### 2.2 メッセージ送信アルゴリズム

この章ではメッセージ送信アルゴリズムについて説明する。各  $\mathcal{A}_i$  をパーティション、 $G = (V, E, l)$  をパーティション間のつながりを表したグラフ、クエリー  $Q$  は  $\mathcal{A}_k$  において証明したい式であるとする。

[定義 2.1] (関係  $\prec$ ) [2]

各  $\mathcal{A}_i$ 、 $G$ 、 $Q$  が与えられたとき、 $dist(i, k) (i, k \in V)$  を  $G$  におけるノード  $i, k$  間の最短路の長さとする。 $dist(i, k) < dist(j, k)$  を満たすとき、 $i \prec j$  とする。

[アルゴリズム 2.1] (メッセージ送信アルゴリズム) [2]

- (1) 定義に従って  $\prec$  を決定する。
  - (a) 各パーティション  $\mathcal{A}_i$  に対して並列に結論発見を行う。
  - (b)  $i \prec j$  となるようなすべての  $(i, j) \in E$  に対して、 $\mathcal{A}_j$  の結論  $\varphi$  が見つかれば、 $\varphi \in \mathcal{L}(L(i, j))$  を満たしていたら、 $\varphi$  を  $\mathcal{A}_j$  の公理集合に加える。
  - (c)  $\mathcal{A}_k$  において  $Q$  が証明されたら、YES を返す。

[定理 2.1] (健全性と完全性) [2]

公理集合を  $\mathcal{A}$  とし、 $\mathcal{A} = \bigcup_{i \leq n} \mathcal{A}_i$  としたときの  $\mathcal{A}_i$  をパーティション、 $G$  をパーティションから構成されるグラフとする。ここでグラフ  $G$  は木構造であるとする (閉路を持たない)。もし各パーティションで行われる推論手続きが結論発見に関して健全かつ完全であるなら、メッセージ送信アルゴリズムは YES を出力する。

### 2.3 パーティションが閉路を形成するグラフ

定理 2.1 では与えられるグラフが木構造である必要がある。もしグラフが閉路を形成している場合には、メッセージ送信アルゴリズムをそのまま用いるだけでは定理証明を完全に行うことができない場合がある。またどの順序でメッセージを送信すればよいか分からない。パーティション間においてコミュニケーション言語を決定することをラベル付けというが、グラフが閉路を形成する場合、適切なラベル付けが必要となる。これら 2 つの問題を解決するアルゴリズムを以下に示す。

[アルゴリズム 2.2] (閉路除去アルゴリズム) [2]

- (1)  $G$  において最小の長さを持つ閉路  $v_1, \dots, v_c (v_i \in V, v_1 = v_c)$  を見つける。もし閉路がなければ  $G$  を返す。
- (2)  $a < c$  であり、 $\Sigma_{a+j < c} |L(v_j, v_{j+1}) \cup L(v_a, v_{a+1})|$  が最小となるような  $a$  を決める。
- (3) すべての  $j < c, j \neq a$  について、 $L(v_j, v_{j+1})$  を  $L(v_j, v_{j+1}) \cup L(v_a, v_{a+1})$  とする。
- (4)  $E$  を  $E \setminus \{(v_a, v_{a+1})\}$ 、 $L(v_a, v_{a+1})$  を  $\emptyset$  とし、(1) へ行く。

閉路を形成するグラフが木構造へ変換される例を図 1 に示す。左側の図は右側の図に変換することができる。まず、最小の閉路 (1,3),(3,4),(4,1) に注目し、辺 (4,1) を削除する。その辺のコミュニケーション言語を (1,3),(3,4) に加える。そして再び最小の閉路 (2,3),(3,4),(4,2) に注目し、辺 (2,3) を削除する。そしてその辺のコミュニケーション言語を (3,4),(4,2) に加える。(s はすでに (3,4),(4,2) に存在している) 最後に最小の閉路 (1,3),(3,4),(4,2),(2,1) に注目する。辺 (4,2) を削除し、s を残りの辺に加える。このようにすると  $\mathcal{A}_2$  から  $\mathcal{A}_1$  に  $\neg p \vee s$  が、 $\mathcal{A}_1$  から  $\mathcal{A}_3$  に  $q \vee r \vee s$  が、 $\mathcal{A}_3$  から  $\mathcal{A}_4$  に  $r \vee s$  が送信される。そして  $\mathcal{A}_4$  で s という結論が得られる。

最小の長さの閉路が複数ある場合、閉路の共通の辺を除くことに注意する。共通の辺がない場合には、コミュニケーション言語の総和が最小になるように辺を削除する。削除する辺の順序指定があるのには理由がある。どの辺から削除しても木構造に変換できるし、問題を解くことができる。しかし閉路除去アルゴリズムは各パーティション間のコミュニケーション言語の

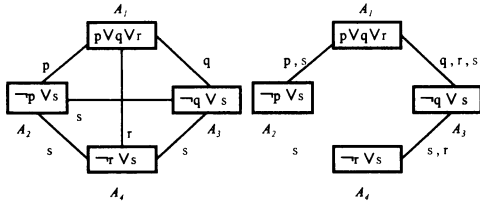


図1 閉路を形成するグラフと木構造への変換

総和をできるだけ少なくすることを目的にしている。コミュニケーション言語が少なくなれば、送信できるメッセージの数を少なくすることができる。つまり、メッセージの送信先で行われる融合の数を減らすことができる。このため削除する辺には順序が決まっている。

### 3. SOL 導出

本章では各パーティションで結論発見を行うための推論手続きである SOL 導出の定義 [4] とこれに用いる語句を説明する。また SOL 導出の健全性と完全性 [4] について述べる。

#### 3.1 SOL 導出

ある与えられた公理集合に隠れている結論を発見することは、自動演繹にとって興味深い問題であり、これを結論発見と呼んでいる。結論発見は定理証明のように与えられた式を導けるかを調べるだけでなく、ある特徴を持つ定理を発見するという点で定理証明とは異なる。そこで結論発見を自動で行うために、Inoue [4] は、**特徴節**と呼ばれるある特徴をもった”面白い”節を導き出すことに着目し、結論発見問題を再構成した。この特徴節は、与えられた節集合  $\Sigma$  から導出され、包摂に関して極小であり、**生成領域**と呼ばれる表現言語の語彙で構築される。また特徴節の概念はアブダクション、非単調推論、主項の生成、知識コンパイル、および帰納的論理プログラミングを含む人工知能にとって興味ある様々な推論問題に有用である。上記において包摂とは、2つの節  $C$  と  $D$  があるとしたとき、 $C\theta \subseteq D$  が成り立ち、 $C$  が  $D$  より多くのリテラルを持たないような代入  $\theta$  が存在することであり、このことを  $C$  は  $D$  を**包摂する**という。また  $C$  は  $D$  を包摂するが、 $D$  は  $C$  を包摂しない場合、 $C$  は  $D$  を**真に包摂する**という。節集合  $\Sigma$  に対して  $\mu\Sigma$  は  $\Sigma$  のどんな節にも真に包摂されない  $\Sigma$  の節の集合を表す。

以下に特徴節を表現する生成領域 (*production field*) の定義を示す。

[定義 3.1] **生成領域** [4]

- (1) リテラルの集合を  $\mathcal{L}$  と表す。変数を持つリテラルが  $\mathbf{L}$  に存在するときはいつもそのすべての例も  $\mathbf{L}$  に存在するならば、リテラルの集合  $\mathbf{L} \subseteq \mathcal{L}$  は例化のもとで閉じているという。
- (2) 生成領域  $\mathcal{P}$  は  $(\mathbf{L}, Cond)$  で与えられる。ここで、 $\mathbf{L}$  を  $\mathcal{P}$  の**特徴リテラル**といい、 $\mathcal{L}$  の部分集合であり、例化のもとで閉じているものとする。 $Cond$  は満たされるべき条件で、 $Cond$  が指定されない場合、生成領域  $\mathcal{P}$  は  $(\mathbf{L})$  と記述される。
- (3) ある節  $C$  のリテラルがすべて  $\mathbf{L}$  に属しかつ  $C$  が

$Cond$  を満たすならば、節  $C$  は生成領域  $\mathcal{P} = (\mathbf{L}, Cond)$  に属するという。 $\Sigma$  を節集合とすると、 $Th_{\mathcal{P}}(\Sigma)$  は  $\Sigma$  の定理であり  $\mathcal{P}$  に属するすべての節の集合を表す。

(4) 節  $D$  が  $\mathcal{P}$  に属しており、 $C$  が  $D$  を包摂するときはいつも  $C$  も  $\mathcal{P}$  に属するならば、生成領域  $\mathcal{P}$  は安定しているという。

[定義 3.2] **特徴節** [4]

$\Sigma$  を節集合、 $\mathcal{P}$  を安定した生成領域とすると、 $\mathcal{P}$  に関する  $\Sigma$  の特徴節の集合を次で定義する。

$$Carc(\Sigma, \mathcal{P}) = \mu Th_{\mathcal{P}}(\Sigma)$$

空節  $\square$  はすべての安定した生成領域  $\mathcal{P}$  に属し、もし  $\Sigma$  が充足不可能であるならば、またそのときのみ、 $Carc(\Sigma, \mathcal{P}) = \{\square\}$  である。これは定理証明が結論発見の特別な場合であることを意味している。

新しい節  $C$  が節集合  $\Sigma$  に追加されるとき、新たに導かれた新しい「面白い」節は**新特徴節**と呼ばれ、次で定義される。

[定義 3.3] **新特徴節** [4]

$\Sigma$  を節集合、 $\mathcal{P}$  を安定した生成領域、 $C$  を節とすると、 $\Sigma$  と  $\mathcal{P}$  に関する  $C$  の新特徴節の集合を次で定義する。

$$Newcarc(\Sigma, C, \mathcal{P}) = \mu [Th_{\mathcal{P}}(\Sigma \cup \{C\}) - Th_{\mathcal{P}}(\Sigma)]$$

また上式の定義は、以下の式と等価である [4]。

$$Newcarc(\Sigma, C, \mathcal{P}) = Carc(\Sigma \cup \{C\}, \mathcal{P}) - Carc(\Sigma, \mathcal{P})$$

Inoue [4] は OL 導出に Skip 操作を追加することによって SOL 導出を実現した。SOL 導出が特徴節または新特徴節を計算するのに有効な理由は、生成領域  $\mathcal{P}$  に属す定理のみを容易に生成できる点である。また新特徴節  $Newcarc(\Sigma, C, \mathcal{P})$  を求めるのに、**先頭節**と呼ばれる新しく追加された節  $C$  から推論が行われるため、手続きが新たに追加された情報に関係した定理を直接導くことができるという点で、結論発見にとって望ましいものである。

#### 3.2 SOL 導出の健全性と完全性

SOL 導出の健全性と完全性 [4] は以下の通りである。

[定理 3.1] (健全性と完全性)

- (健全性) 節  $S$  が  $\Sigma + C$  と  $\mathcal{P}$  からの SOL 演繹で得られるならば、 $S$  は  $Th_{\mathcal{P}}(\Sigma \cup \{C\})$  に属する。
- (完全性) 節  $F$  が  $Th_{\mathcal{P}}(\Sigma)$  に属さず、 $Th_{\mathcal{P}}(\Sigma) \cup \{C\}$  に属するならば、 $F$  を包摂する節  $S$  を得る  $\Sigma + C$  と  $\mathcal{P}$  からの SOL 演繹が存在する。

### 4. 提案手法

2章では Amir らの手法を用いた定理証明について説明を行った。しかし、この手法では結論発見に対して完全ではない。この章ではまず、結論発見に対して完全ではない例を挙げる。そして結論発見に対して完全であるために提案手法 1 として Amir らの手法を拡張する。さらに提案手法 2 としてより協調的なアルゴリズムを提案する。

#### 4.1 例

まずはじめに、Amirらの手法では導くことができない結論が存在する例を示す。この問題はお金を下ろすにはカードか通帳が必要であることを導く問題であり、 $card \vee bankbook$  という結論を導く問題である。この問題を解く上で  $need\_money$ ,  $get\_money$  という節が外部からの応答として得られるものとする。

• $\neg holiday \vee close$	
• $\neg normal\_day \vee open$	
• $normal\_day \vee holiday$	
• $\neg need\_money \vee \neg open \vee ATM \vee counter$	
• $\neg need\_money \vee \neg close \vee ATM$	
• $\neg ATM \vee card \vee \neg get\_money$	
• $\neg counter \vee bankbook \vee \neg get\_money$	
• $need\_money$	(外部からの応答)
• $get\_money$	(外部からの応答)

図2 例題 get money 問題

この問題が示す内容は以下の通りである。

- 休日は銀行は閉まっている。
- 平日は銀行は開いている。
- 平日または休日である。
- お金が必要で銀行が開いているなら ATM か窓口に行く。
- お金が必要で銀行が閉まっているなら ATM に行く。
- ATM でカードがなければお金を下ろせない。
- 窓口で通帳がなければお金を下ろせない。
- お金が必要である。
- お金を下ろした。

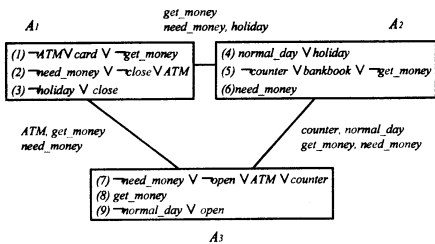


図3 パーティションの構成

ここで図6のようにパーティションが構成されていたと仮定する。 $A_1$ と $A_3$ 間の辺を削除し $ATM$ を残りの辺に加える。しかし $card \vee bankbook$ という結論を導くことができない。この結論を導くためには(1)から(7)までのすべての節が融合される必要がある。閉路除去アルゴリズムを用いたコミュニケーション言語の決定方法では、 $L(1,2)$ に $card$ が含まれていないため(1)と(5)を融合することができない。 $card \vee bankbook$ という結論を得るためには(1)が $A_2$ に送信できるようなコミュニケーション言語の決定方法が必要になる。そこでこのような問題を解決するコミュニケーション言語の決定方法を提案する。

#### 4.2 提案手法1

ここでは結論発見に対して完全であるためにコミュニケーション言語の決定方法を新たに提案する。

公理集合を $A$ とし、 $A = \bigcup_{i \leq n} A_i$ としたとき、各パーティションを $A_i$ とする。 $A_i$ と $A_j$ 間の共通言語を $C(i,j)$  ( $j \leq n, i \neq j$ ),  $A_i$ と $A_j$ 間のコミュニケーション言語を $L(i,j)$ とする。閉路の場合に辺 $(s,r)$ を削除するものと仮定し、 $A_s$ と $A_r$ 間のコミュニケーション言語を $L(s,r)$ とする。また生成領域を $\langle L \rangle$ とする。ここでコミュニケーション言語に生成領域のリテラルを加えるという操作を行う。すると木構造の場合のコミュニケーション言語は

$$L(i,j) = C(i,j) \cup L$$

となる。また閉路の場合のコミュニケーション言語は

$$L(i,j) = C(i,j) \cup L(s,r) \cup L$$

となる。

[アルゴリズム 4.1] (結論発見のアルゴリズム)

根のパーティションを $A_i$ とし、 $A_i$ の子のパーティションを $A_j$ とする。

(1) パーティション間に閉路が存在する場合には、閉路除去アルゴリズムを用いて木構造にする。

(2) 各パーティション間のコミュニケーション言語を定める。

(3)  $A_i$ において結論発見を行う。

(4) 公理集合 $A_i$ と $A_i$ で得られた結論うち、包摂に関して極小の節がコミュニケーション言語から構成されているか判定する。そのような節が存在すればメッセージとして $A_j$ に送信し、 $A_j$ に加える。

(5) メッセージが送信可能な限り(メッセージがコミュニケーション言語から構成されるものが存在する限り) $A_j$ において(3)と(4)の操作を繰り返す。各パーティションで得られた結論のうち包摂に関して極小の節が結論である。

ここで提案手法1の完全性について述べる。

[定義 4.1] (完全性)

• 分割されていなければ導出できるようないかなる節 $C$ も、ある分割と通信の方式によってある節 $D$ が導出でき、 $D$ が $C$ を包摂するならば、その分割・通信の方式は完全である。

• また、どのような分割をしても、ある通信の方式を用いることで、その分割・通信の方式が完全であるならば、その通信の方式は完全である。

公理集合を $A$ とし、 $A = \bigcup_{i \leq n} A_i$ としたとき、各パーティションを $A_i$ とする。またメッセージを受信し、以降のパーティションにメッセージを送信しないようなパーティションを $A_k$ とする。

[定理 4.1] (提案手法1の完全性) [7]

提案手法1は $A_k$ において結論発見に対して完全である。

[定理 4.2] (提案手法1の健全性) [7]

提案手法1は $A_k$ において結論発見に対して健全である。

提案したコミュニケーション言語の決定方法を用いると図9のようになる。このようにすると(1)は $L(1,2)$ から構成され(5)と融合することができる。また結論発見のアルゴリズムを用いることにより $card \vee bankbook$ という結論を含めたすべての結論を得ることができる。

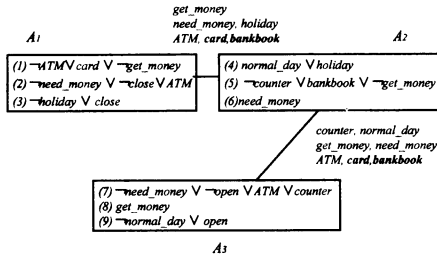


図4 新しいコミュニケーション言語の決定

### 4.3 提案手法2

提案手法1では、辺を削除する際に残りの辺にコミュニケーション言語を加えるため、パーティション間のコミュニケーション言語が大幅に増えてしまう可能性がある。その結果、知識の構成の仕方によってはパーティションが持っている知識の大半を他のパーティションに送信しなくてはならないということが起こり、一つのパーティションの負荷が大きくなってしまふ。このことを解決するために各パーティションが協力し合って問題を解いていくようなアルゴリズムを提案する。

[アルゴリズム 4.2] (結論発見のアルゴリズム)

(1) 単位節が存在する場合にはその補リテラルを含む節が存在するパーティションに単位節を送信する。

(2) 先頭節の存在するパーティションにおいて SOL 導出を用いて結論発見を行う。新しく結論が得られれば、公理集合に加える。公理集合中で包摂される節を消去する。先頭節とそこから得られる結論のうち包摂に関して極小の節の集合を  $\Sigma$  とする。

(3)  $\Sigma$  中の節  $C$  の各リテラルに注目する。そのリテラルの補リテラルを持つ節が他のパーティションに存在するか探し、存在すれば  $C$  を補リテラルを含むすべてのパーティションに送信し、公理集合に加える。 $C$  が送信したパーティションの節に包摂される場合は公理集合から削除する。(補リテラルが他のパーティションに見つからない場合、 $C$  を送信しない)

(4) そのパーティションにおいて  $C$  をそれぞれ先頭節として、SOL 導出を用いて結論発見を行う。新しく結論が得られれば、公理集合に加える。先頭節とそこから得られる結論のうち包摂に関して極小の節の集合を  $\Sigma$  とする。

(5) 新しい結論が導出されなくなるまで3から5の操作を繰り返す。

ここで提案手法2の完全性について述べる。

[定理 4.3] (提案手法2の完全性) [7]

提案手法2は結論発見に対して完全である。

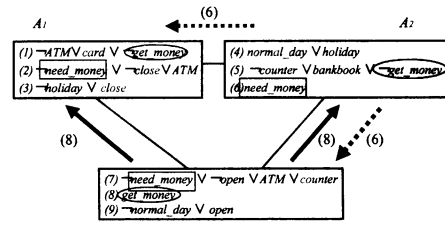


図5 提案手法2

提案手法2ではももとの節をメッセージとして送信することを行わない。あるパーティションで得られた結論を他のパーティションに送信し、そのパーティションで得られた結論をまた他のパーティションに送信するという操作を繰り返す。つまり、問題を解くのに各パーティションが協力し合い、少しずつ問題を解いていき、最終的な結論を得ようになっている。よって特定のパーティションのみ演繹数が大きくなる可能性が低くなり、各パーティションが均等に処理を行うことができる。しかしデメリットも存在する。パーティションで得られた結論と融合可能な節が他のパーティションに存在するのならばすべてのパーティションにすべての結論を送信しなければならない。よって提案手法1と比較すると送信しなければならないメッセージの数は大きくなってしまふ。

## 5. 実験

この章では問題を分割しない場合、提案手法1を用いた場合、提案手法2を用いた場合について実験を行い、融合回数とメッセージの送信数の比較を行う。問題は4.1の例題を用いる。各パーティションで推論を行うシステムとして SOL 導出を Java 言語で実装した SOLAR [3] を用いた。提案手法1におけるメッセージの送信は  $A_1 \rightarrow A_2 \rightarrow A_3$  の順に行われるものとした。

### 5.1 融合回数の比較

それぞれの手法に対して融合回数の比較を行う。以下に実験結果を示す。

表1 各手法における融合数

	$A_1$ [回]	$A_2$ [回]	$A_3$ [回]	合計 [回]
問題を分割しない場合	-	-	-	659
提案手法1を用いた場合	19	51	461	531
提案手法2を用いた場合	27	62	63	152

まず、問題を分割しない場合と提案手法1の比較を行う。提案手法1では子のパーティションに送信する節をコミュニケーション言語で構成されるものに限定するため、各パーティションで行われる融合回数の総和は分割しない場合よりも減少している。問題の規模が大きくなればなるほど分割しない場合と提案手法1の融合数の差が大きくなると考えられる。より効率的に問題を解くためには、各パーティションの中のみで行われるようなローカルな推論の数が多くなるような分割が必要である。

次に問題を分割しない場合と提案手法2の比較を行う。表を見ると提案手法2の方が問題を分割しない場合よりも融合数

が減少しているのが分かる。提案手法2ではあるパーティションで得られた結論を融合可能な節が存在するすべてのパーティションの公理集合に加えるという操作を繰り返す。送信先のパーティションで得られる結論は送信前の節と比較すると推論の進んだ状態だと言える。つまり、推論を行う際にある程度推論の進んだ状態の節を使うことができ、冗長な計算を減らすことが可能となるため融合数が減少されると考えられる。

最後に提案手法1と提案手法2の比較を行う。提案手法1を用いた場合では根のパーティションでは公理の数が少ないため、融合の回数は小さい。メッセージを子のパーティションに送信して公理集合を増やしていくことによって融合できる節が集まってくるため、子孫のパーティションの方が融合の回数が大きくなっているのが分かる。つまり、公理の大半が子孫のパーティションに集まりやすくなる。その結果、処理の大半が子孫のパーティションに委ねられてしまう。実験の結果を見ると処理の大半が $A_3$ に集中していることが分かる。これに対し、提案手法2を用いた場合には得られた結論を融合可能なパーティションにすべてに送信する。このことにより、各パーティションが問題を協力し合って解くことができ、一つのパーティションの負荷が大きくなる可能性を防ぐことができる。実験の結果を見ると一つのパーティションに処理の大半を任せるのではなく、各パーティションが協力し合って処理を行っていることが分かる。

## 5.2 メッセージ送信数の比較

次に提案手法1を用いた場合と提案手法2を用いた場合のメッセージ送信数の比較を行う。以下に実験結果を示す。

表2 各手法におけるメッセージ送信数

	$A_1$ [回]	$A_2$ [回]	$A_3$ [回]	合計 [回]
提案手法1を用いた場合	3	5	0	8
提案手法2を用いた場合	5	7	8	20

提案手法1を用いた場合、閉路が存在する際には一辺を削除して残りの辺にコミュニケーション言語を加えるという操作が行われる。この操作により閉路が木構造に変換されるため、根のパーティションから葉のパーティションへと一方向にメッセージの送信が行われる。つまり、各パーティションで導き出された結論は子のパーティションに送信するだけでよい。それに対し提案手法2では各パーティションで得られた結論がその他のパーティションの節と融合できるのなら、すべてのパーティションにすべての結論を送信するようにしている。得られた結論がその他のパーティションに融合できる節が存在しなくなるか、その他のパーティションの節に包摂されて送信できなくなるまで、メッセージの送信を行わなければならないためメッセージの送信数は大きくなる。実験の結果を見ても分かるように、提案手法2の方が提案手法1よりもメッセージ送信数が大きくなっている。問題の規模が大きくなればなるほど提案手法1と2のメッセージ送信数の差は大きくなると考えられる。

## 6. まとめと今後の課題

公理集合が最初から分散された状態において分散されたまま

結論発見を行う手法を2つ提案した。一つ目はAmirらの手法を結論発見に拡張する方法、2つ目はAmirらの手法によらずより協調的に解く方法である。そして問題を分割しない場合、提案手法1、提案手法2にそれぞれに対して実験を行った。まず融合回数の比較を行った。提案手法1についても2についても問題を分割しない場合と比較したところ、融合数の減少が見られた。提案手法1を用いた場合、より効率的に問題を解くためにはパーティションの構成が重要であることが分かった。またメッセージの送信数という面から提案手法1、提案手法2に対して比較を行った。結果、提案手法1の方が提案手法2よりもメッセージの送信数が少ないことが確認できた。

今回行った比較は一例であり、今後の課題はパーティションの構成によって問題を解く際の演繹数とメッセージの送信数がどのように変化するか調べることである。演繹数とメッセージの送信数がともに小さくなるようなパーティションの構成を発見することが必要である。また関連研究として、より戦略的に問題を解決する方法や分散SAT問題への適用などが挙げられる。これらについては[7]を参照のこと。

## 文 献

- [1] Chang, C.-L. and Lee R. C.-T : *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973.
- [2] Eyal Amir, Sheila McIlraith : Partition-based logical reasoning for first-order and propositional theories, *Artificial Intelligence*, 162, pages 49-88, 2005.
- [3] Hidetomo Nabeshima, Koji Iwanuma, and Katsumi Inoue : SOLAR : A Consequence Finding System for Advanced Reasoning. *Marta Cialdea Mayer and Fiara Pirri(eds.), Automated Reasoning with Analytic Tableaux and Related Methods:Proceedings of the 11th International Conference(TABLEAUX 2003),Lecture Notes in Artificial Intelligence*, 2796, pages 257-263, Springer, 2003.
- [4] Katsumi Inoue : Linear resolution for consequence finding, *Artificial Intelligence*, 56, pages 301-353, 1992.
- [5] Katsumi Inoue and Koji Iwanuma : Speculative computation through consequence-finding in multi-agent Environments. *Annals of Mathematics and Artificial Intelligence*, 42, pages 255-291, 2004.
- [6] Pierre Marquis : Consequence finding algorithms. In: Dov M. Gabbay and Philippe Smets, editors, *Handbook for Defeasible Reasoning and Uncertain Management Systems*, 5, pages 41-145, Kluwer Academic, 2000.
- [7] 岡本孝之 : メッセージ通信を用いた分散型結論発見に関する研究, 神戸大学大学院自然科学研究科修士論文, 2005.