

Situated ATMS

飛鳥井 正道

キヤノン(株) 情報システム研究所

junet: ask@canon.co.jp

不完全な知識に基づいて推論を行なうためのひとつの方法として仮説推論がある。de Kleer により提案された ATMS は仮説推論を行なうときに知識ベースの管理を行なうための機構である。しかしながら、ATMS ではすべての仮説を同等に扱うために、多数の仮説を扱うような問題では仮説の組合せ的爆発の問題が不可避である。本論文では、状況という概念を導入することにより、仮説が導入された状況を扱うように ATMS を拡張した **Situated ATMS (SATMS)** を提案する。SATMS は従来の多重コンテキストによる推論と ATMS を用いた仮説推論とを包含した枠組みをも与えている。また、SATMS を仮説間に制約を記述できるように拡張するとともに、このときのラベル計算の無矛盾性と完全性を維持するための Hyperresolution について述べる。

A Situated Assumption-based Truth Maintenance System

Masamichi ASUKAI

Canon Inc., Information Systems Research Center

890-12, Kashimada, Saiwai-ku, Kawasaki, Kanagawa 211, Japan

Hypothetical reasoning is a powerful reasoning technique to draw a conclusion from incomplete knowledge. The ATMS proposed by de Kleer is a maintenance system of knowledge base in problem solving with hypothetical reasoning. However, a problem of combinatorial explosion is unavoidable as the number of assumptions increases, since the ATMS handles all assumptions equally. In this paper, a **situated ATMS(SATMS)** is proposed, which is an extension of the ATMS to handle situated assumptions. The SATMS gives a framework to subsume both reasoning with multiple contexts and hypothetical reasoning with the ATMS. When assumptions are introduced in a situation, the SATMS adds the information of the situation to the assumptions. By dividing assumptions into each situated assumptions, the risk of combinatorial explosion is reduced. Extended hyperresolution to handle disjunctions of situated assumptions is also described.

1 まえがき

問題解決の過程では、対象についてすべてを知っていることはまれであり、通常は限られた知識に基づいて推論しなければならないことが多い。不完全な知識に基づいて結論を導き出すためのひとつの方法は、不完全な知識を補うためになんらかの仮説を立てて推論を行なうことである。

仮説に基づいて推論を行なうときに、知識ベースの管理を効率的に行なうための機構として de Kleer の提案した ATMS [1] がある。ところが、ATMS はすべての仮説を等しく扱うために、多数の仮説を扱うような問題では仮説の組合せ的爆発の問題が不可避である。本論文では、状況という概念を導入することにより、仮説が導入された状況を扱えるように ATMS を拡張した **Situated ATMS (SATMS)** を提案する。SATMS では、異なる状況で導入された仮説を区別することにより、組合せ不可能なそれぞれの状況で導入された仮説の組合せを回避するなど、探索空間の縮小をはかるための柔軟な枠組みを提供している。また、SATMS は従来の多重コンテキストによる推論 [2] と ATMS を用いた仮説推論とを包含した枠組みを与えている。

SATMS と同様に仮説に状況を導入したものとしては、董らの研究 [3] がある。董らは同じ内容ではあるが異なった状況で導入された仮説を区別することにより、行為・時間などの順序にかかわる問題の解決を目的としている。しかしながら、状況を取り扱うのは基本的には問題解決機構であり、ATMS は状況間および環境間の順序関係によるラベルの環境などの極小性を維持している以外は、状況を含めた仮説を命題として扱っている点では ATMS と同様である。従って、状況ごとに導入された仮説の総数が増大することによる仮説の組合せ的爆発の問題は回避できない。

本論文では、まず第 2 章で SATMS の基礎となっている de Kleer の ATMS について紹介するとともに、ATMS が本質的に持つ組合せ的爆発の問題を解決するための従来の手法について述べる。次に、第 3 章では、状況化された仮説を扱えるように ATMS を拡張した SATMS を提案する。SATMS における基本概念を説明するとともに、ラベル計算アルゴリズムを中心とした SATMS の動作について述べる。さらに、仮説間の制約を記述できるように拡張するとともに、このときのラベル計算の無矛盾性と完全性を維持するための Hyperresolution について述べ

る。第 4 章では、SATMS を用いた簡単な例題を示し、第 5 章では、他の研究との関連性や今後の課題について考察する。

2 背景

2.1 de Kleer の ATMS

ATMS を用いた推論システムは、問題解決機構 (problem solver) と ATMS とから構成される (Fig.1)。問題解決機構は、推論の実行ごとに推論の結果を理由付け (justification) として ATMS に与える。ATMS は与えられた理由付けを記録するとともに、これらの理由付けからデータがどのコンテキストで信じられているかという信念 (belief) を計算して問題解決機構に返す。ATMS では、問題解決機構のデータはノードとして表現される。理由付けは、あるノードが他のノードからどのようにして導出されたかを、後件と複数の前件により、

$$x_1, x_2, \dots \Rightarrow n$$

と表現される。また、環境 (environment) は仮説の集合として表現される。 m 個の仮説 $A_1, A_2, \dots, A_m$ から構成される環境 E は、

$$E = \{A_1, A_2, \dots, A_m\}$$

と表現される。理由付けの集合 J を持つノード n が環境 E のもとで信じられているとは、

$$E, J \vdash n$$

が成立することをいう。より直観的には、

$$J \vdash E \rightarrow n \quad (\rightarrow \text{は含意記号})$$

のように、問題解決機構から与えられた理由付けの集合 J から、 $E \rightarrow n$ となるようなノード n に対する環境 E が計算できることである。言うまでもなく、 $E \subset E_i$ となる環境 E_i では $E_i \rightarrow n$ が必ず成立する。従って、このようにして計算された環境の集合のうち、極小の環境の集合がラベルとしてノードに記録され、ノードがどの環境で信じられているかを表現する。ここで、極小な環境 E とは、 $E_j \subset E$ かつ $E_j \rightarrow n$ を満足する環境 E_j が存在しないことを言う。このように、理由付けの集合からノードが信じられている極小な環境の集合を求めることをラベル

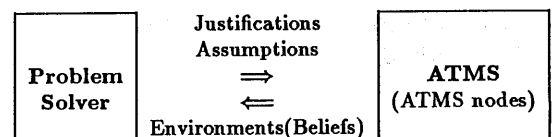


Fig. 1: Problem Solver and ATMS

計算という。また、

$E, J \vdash \perp$

のように矛盾 $\perp$ を導出する環境は、すべてのノードのラベルから取り除かれるとともに、極小な矛盾環境として矛盾データベースに記録される。

2.2 ATMS の拡張

ATMS において N 個の仮説を扱う場合には、最大で 2^N 個の環境からなる探索空間を考慮しなければならない。従って、多数の仮説を扱うような問題では仮説の組合せ的爆発の問題が不可避である。このような大規模な探索空間から解を効率的に発見するための従来の手法としては、探索の制御と探索空間の縮小とがある。

探索の制御としては、問題解決機構が探索の焦点となる環境 (focus environment) を制御することにより、効率的な探索を行なう de Kleer らによる方法 [4] が考えられる。さらには、問題解決機構と ATMS との間にインタフェースを設けることにより探索の制御を行なう井上の方法 [5] など [6, 7] や、仮説に不確実性の情報を付加することにより探索の制御などを行なおうとする董らの方法 [8] や劉らの方法 [9] など [10, 11, 12] がある。

探索空間の縮小としては、de Kleer により拡張された ATMS [13] を利用して仮説間に制約を記述する飛鳥井らの方法 [14] や EBL により矛盾の学習を行なう矢沢らの方法 [15] もある。

3 Situated ATMS

本章では、状況化された仮説に基づく Truth Maintenance System である SATMS を提案する。SATMS を用いた推論システムは、ATMS と同様に問題解決機構と SATMS とから構成され (Fig.2), SATMS は問題解決機構から与えられた理由付けを記録するとともに、理由付けからラベル計算を行なう。

ただし、SATMS では仮説に、その仮説が導入された状況が情報として付加されており、SATMS はこの状況に関する状況演算も合わせて行い、結果としてその理由付けがどのような状況のどのような環境のもとで成立するかを状況依存環境として返す。

3.1 基本概念

【状況】

状況とは、情報が不完全であるために知識の意味が

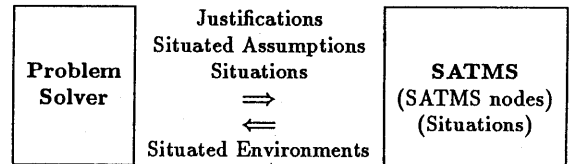


Fig. 2: Problem Solver and SATMS

異なったりするときの文脈、時間的に対象の状態が変化するときの時刻、信念や知識に基づいて推論を行なうときの信念や知識の状態などに対応する。そして、個々の命題¹の真偽値は状況に依存して決定され、「状況 s において命題 p が真である」ことを、

$$s \models p$$

と表現する。

【状況演算】

SATMS では、宣言された状況の集合 S を以下に定義する順序関係「 $\preceq$ 」が成立する半順序集合と考える。ここで、状況間の順序関係は問題解決機構により宣言され、SATMS が記録する。また、状況間の順序関係に基づく推論規則および状況演算を導入する。

【定義1】状況間の順序関係「 $\preceq$ 」

状況間の順序関係 $\preceq$ とは、反射法則、反対称法則、推移法則を満足する順序関係である。

□

【定義2】推論規則

$$s_i \preceq s_j$$

$$s_i \models p$$

$$\hline s_j \models p$$

□

この推論規則は状況の順序に関する単調性を保証しており、状況 s_i において真であるすべての命題は状況 s_j においても真となる。

【定義3】状況演算「 $\sup$ 」

状況の集合 S の任意の部分集合 S_{sub} に対して、状況演算 $\sup S_{sub}$ は集合 S_{sub} の上限 (最小上界) の状況を返す。ただし、上限の状況が状況の集合 S に存在しない場合には空を返す。

□

¹問題解決機構が扱うのは命題論理の命題に限定する必要があるが、SATMS は ATMS と同様に常にこれらを命題論理の命題として扱う。

【状況依存仮説, 状況依存環境】

ある状況において真偽値が確定しない命題は, その状況において仮説として扱うことができる。仮説はあくまでもある状況においての仮説であり, 別の状況においては真であり, また別の状況においては偽であるかも知れない。SATMSで扱うこのような仮説を「状況依存仮説(situated assumption)」と呼ぶ。

また, 「状況依存環境(situated environment)」は, 状況演算により計算される状況 s と環境 E の対として, $[s, E]$ と表現される。さて, 命題のラベルや矛盾データベースを極小な状況依存環境で管理するために, 状況依存環境の順序関係, および極小状況依存環境を定義する。

[定義4] 状況依存環境の順序関係「 $\subseteq, \subset$ 」

2つの状況依存環境 $SE_1 = [s_1, E_1]$, $SE_2 = [s_2, E_2]$ に以下に示す関係が成立するとき, $SE_1 \subseteq SE_2$ であると定義する。

$$s_1 \subseteq s_2 \quad \text{かつ} \quad E_1 \subseteq E_2$$

また, 次の関係が成立するとき, $SE_1 \subset SE_2$ であると定義する。

$$SE_1 \subseteq SE_2 \quad \text{かつ}$$

$$s_1 \neq s_2 \quad \text{または} \quad E_1 \neq E_2$$

□

【定義5】極小状況依存環境

状況依存環境の集合において, $SE_i \subset SE$ を満足する SE_i が存在しないとき, SE は極小状況依存環境である。

□

【矛盾状況依存環境】

ATMSと同様に, 矛盾を後件に持つ理由付け,

$$x_1, x_2, \dots \Rightarrow \perp$$

において, 前件の集合から計算される状況依存環境 $[s, E]$ の集合は, すべての命題のラベルから取り除かれるとともに, 矛盾状況依存環境,

$$snogood[s, E]$$

として矛盾データベースに記録される。この矛盾状況依存環境は,

$$s \models (E \rightarrow \perp)$$

を意味する。また, 矛盾データベースでは, 極小な矛盾状況依存環境が管理される。

【SATMS ノード】

SATMSでは, 命題はSATMSノードとして取り扱われる。SATMSノードはその命題の内容(datum)とラベル(label)と理由付けの集合(justifications)

とから,

$$\langle datum, label, justifications \rangle$$

$$label = \{SE_1, SE_2, \dots\}$$

$$SE_i = [s_i, E_i]$$

$$E_i = \{A_{i1}, A_{i2}, \dots\}$$

のように構成される。内容と理由付けはATMSと同様であり, 問題解決機構から与えられる。ラベルは, 理由付けからSATMSがラベル計算により計算したものであり, 状況依存環境 SE の集合として表現される。

状況依存環境 SE は状況 s と環境 E の対として表現され, そのノードがどのような状況のどのような環境のもとで成立するかを表現する。従って,

$$s_i \models (E_i \rightarrow datum)$$

すなわち,

$$s_i \models (A_{i1} \wedge A_{i2} \wedge \dots \rightarrow datum)$$

を意味する。また, 状況 s における状況依存仮説 A はSATMSノードとして,

$$\langle A, \{[s, \{A\}], \dots\} \rangle$$

と表現され, 状況 s において常に真である命題 p は,

$$\langle p, \{[s, \{\}], \dots\} \rangle$$

と表現される。

3.2 SATMS の動作

問題解決機構はSATMSに対して, 状況および状況間の順序関係の宣言, 状況依存仮説の宣言などを行なうことができ, 宣言された状況依存仮説はSATMSノードとして記録される。

さて, 問題解決機構は推論の実行ごとに推論の結果を理由付け,

$$x_1, x_2, \dots \Rightarrow n$$

としてSATMSに与える。SATMSは理由付けを記録するとともに, 理由付けの後件の命題がどのような状況のどのような仮説の組合せのもとで成立するかを, 理由付けの前件の命題の集合から状況依存環境として計算して問題解決機構に返す。このときに計算された状況依存環境 $[s, E]$ の集合を後件のラベルといい, 後件とともにSATMSノードとして記録される。ここで, 状況 s は前件の状況依存環境の状況部分から状況演算により計算され, 環境 E は前件の状況依存環境の環境部分から集合演算により計算される。

【例3.1】SATMS の動作

状況として s_1, s_2, s_3 を考え, 以下に示す順序関係が成り立つものとする。

$$s_1 \preceq s_2$$

$$s_1 \preceq s_3$$

さらに、命題として「 $x + y = 1$ 」, 「 $x = 1$ 」, 「 $y = 0$ 」が存在して、以下に示す SATMS ノードが生成されているものとする。

$$\gamma_{x+y=1} : \langle x + y = 1,$$

$$\{[s_1, \{A, B\}], [s_2, \{B, C, D\}], \dots \rangle$$

$$\gamma_{x=1} : \langle x = 1, \{[s_2, \{A, C\}], [s_3, \{D, E\}], \dots \rangle$$

$$\gamma_{y=0} : \langle y = 0, \{\}, \dots \rangle$$

また、状況 s_1 において環境 $\{A, B, E\}$ が矛盾であるとする。

$$snogood[s_1, \{A, B, E\}]$$

ここで、問題解決機構が「 $x + y = 1$ 」と「 $x = 1$ 」とから「 $y = 0$ 」を推論した場合、問題解決機構から SATMS に送られる理由付けは、

$$\gamma_{x+y=1}, \gamma_{x=1} \Rightarrow \gamma_{y=0}$$

である。ラベル計算の結果、

$$\{[s_2, \{A, B, C\}],$$

$$[s_3, \{A, B, D, E\}],$$

$$[s_2, \{A, B, C, D\}]\}$$

となる。ここで、

$$\text{包含関係} : [s_2, \{A, B, C\}] \subseteq_s [s_2, \{A, B, C, D\}]$$

$$\text{矛盾関係} : [s_1, \{A, B, E\}] \subseteq_s [s_3, \{A, B, D, E\}]$$

から「 $y = 0$ 」のラベルは、

$$\gamma_{y=0} : \langle y = 0,$$

$$\{[s_2, \{A, B, C\}], \{(\gamma_{x+y=1}, \gamma_{x=1})\} \rangle$$

となる。

3.3 ラベル計算アルゴリズム

SATMS が問題解決機構から理由付け j を与えられたときの、ラベル計算アルゴリズムを説明する。ここでは、de Kleer による ATMS のラベル計算アルゴリズム [16] をもとに拡張を行なっている。

SATMS は問題解決機構から理由付け j を与えられたとき、 $SPROPAGATE(j, \phi, \{\{\}\})$ を実行する。 $SPROPAGATE(j, a, I)$ は、ノード a のラベルに状況依存環境の集合 I が付加されたときに理由付け j を伝播させる。SATMS におけるラベル計算アルゴリズムは、基本的には ATMS のラベル計算アルゴリズムにおいて、極小という環境間の順序関係 $\subseteq$ を、SATMS における極小という状況依存環境間の順序関係 $\subseteq_s$ に置き換えればよい。SATMS のラベル計算アルゴリズムが ATMS と大きく異なる点は、合成状況依存環境の計算を行なう $SUNION$ において、まず状況演算が行なわれ、その結果合成状況が

存在しないような状況依存環境は生成されないことである。従って、本来ならばあり得ないような状況の組合せによる不要な環境の生成が回避される。ラベル計算アルゴリズムの概略を以下に示す。

• $SPROPAGATE((x_1, \dots, x_k \Rightarrow n), a, I)$: ラベルの更新の伝播

1. [ラベルの計算]

$L = SWEAVE(a, I, \{x_1, \dots, x_k\})$ により、理由付けの前件の集合からラベル L を計算する。 L が空ならば終了する。

2. [ラベルの更新]

$SUPDATE(L, n)$ により、ノード n のラベルを更新する。

• $SUPDATE(L, n)$: ラベルの更新

1. [矛盾の記録]

$n = \perp$ ならば、 $SE \in L$ である状況依存環境 SE に対して $SNOGOOD(SE)$ を実行して、終了する。

2. [ラベルの極小性の維持]

L からノード n のラベルの状況依存環境の上位状況依存環境をすべて除く。ノード n のラベルから L の状況依存環境の上位状況依存環境をすべて除く。 L の状況依存環境をすべてノード n のラベルに付加する。

3. [ラベルの更新の伝播]

ノード n が前件に含まれているすべての理由付け j に対して、 $SPROPAGATE(j, n, L)$ を実行する。

• $SWEAVE(a, I, X)$: ラベルの計算

1. [終了条件]

理由付けの前件の集合 X が空ならば、状況依存環境の集合 I を返す。

2. 前件の集合 X の先頭要素を h , 残りを R とおく。

3. [ラベルの重複計算の回避]

$h = a$ ならば、 $SWEAVE(\phi, I, R)$ を返す。

4. [ラベルの計算]

I の状況依存環境 SE_i とノード h のラベルの状況依存環境 SE_j のそれぞれの状況依存環境のすべての組合せに対して $SUNION(SE_i, SE_j)$ により計算された状況依存環境の集合を I' とおく。

5. [ラベルの極小性と無矛盾性の維持]

状況依存環境の集合 I' から重複状況依存環境、矛盾状況依存環境、および他の状況依存環境の上位状況依存環境を除く。

6. $SWEAVE(a, I', R)$ を返す。

• $SUNION(SE_i, SE_j)$: 合成状況依存環境の計算

1. 状況依存環境 $SE_i = [s_i, E_i]$, $SE_j = [s_j, E_j]$ とおく。

2. $s_k = \sup\{s_i, s_j\}$ により、合成状況を求める。

3. s_k が空ならば、空を返す。

4. $E_k = UNION(E_i, E_j)$ により、合成環境を求める。

5. 状況依存環境 $SE_k = [s_k, E_k]$ を返す。

• $SNOGOOD(SE)$: 矛盾の記録

1. 状況依存環境 SE を矛盾として記録する。

2. すべてのノードのラベルから状況依存環境 SE とその上位状況依存環境を除く。

3.4 仮説間の制約の記述

探索空間の縮小をより効果的に行なうために、ATMSにおける仮説間の制約の記述 [14] と同様に、SATMSにおいても状況依存仮説間に以下に示す制約の記述を実現している。

nogood: 矛盾関係
choose: 選言関係
oneof: 排他的選言関係

ただし、*oneof*は*nogood*と*choose*の組合せとして表現される。選言 (disjunction) の導入により失われるラベルの無矛盾性と完全性を維持するために、ATMSのHyperresolution [13] を拡張している。まず、SATMSでは矛盾または選言として宣言された状況依存仮説の集合からラベル計算を行ない、状況依存環境を求める。すなわち、状況依存仮説 $A_1, A_2, \dots$ が矛盾または選言として、

$nogood\{A_1, A_2, \dots\}$
 $choose\{A_1, A_2, \dots\}$

ただし、 $\langle A_i, \{\{s_i, \{A_i\}\}, \{(A_i)\} \rangle$ のように宣言されたとき、

$snogood[s_{sup}, \{A_1, A_2, \dots\}]$
 $schoose[s_{sup}, \{A_1, A_2, \dots\}]$

ただし、 $s_{sup} = \sup \bigcup_i \{s_i\}$ のように変換され、それぞれ矛盾データベースまたは選言データベースに極小状況依存環境として記録される。

【無矛盾性の維持】

ラベルの無矛盾性とは、ラベル計算アルゴリズムにより得られたラベル内の状況依存環境に矛盾状況依存環境が含まれないことを言う。選言の導入により失われる無矛盾性を維持するために、以下に示すNegative Hyperresolutionを導入する。

$$\frac{\begin{array}{l} schoose[s, \{A_1, A_2, \dots\}] \\ \forall_i snogood[s_i, \{A_i\} \cup \alpha_i] \\ \text{ただし, } \forall_j A_j \notin \alpha_i \end{array}}{snogood[s_{sup}, \bigcup_i \alpha_i]} \\ \text{ただし, } s_{sup} = \sup[\{s\} \cup \bigcup_i \{s_i\}]$$

ここで、状況 s_{sup} が存在しない場合には、このNegative Hyperresolutionは起動されない。論理的な意味を以下に示す。

$$\frac{\begin{array}{l} s \models A_1 \vee A_2 \vee \dots \\ \forall_i s_i \models \neg A_i \vee \beta_i \end{array}}{s_{sup} \models \beta_1 \vee \beta_2 \vee \dots} \\ \text{ただし, } s_{sup} = \sup[\{s\} \cup \bigcup_i \{s_i\}] \\ \beta_i = \bigvee_{B_{ij} \in \alpha_i} \neg B_{ij}$$

〔例 3.2〕ラベルの無矛盾性

状況として s_1, s_2, s_3 を考え、以下に示す順序関係が成り立つものとする。

$$s_1 \preceq s_2 \\ s_2 \preceq s_3$$

ここで、状況 s_2 で仮説 A が、状況 s_3 で仮説 B が、状況 s_1 で仮説 C が宣言されたとき、以下に示す SATMS ノードが生成される。

$$\begin{array}{l} \langle A, [s_2, \{\{A\}\}], \{(A)\} \rangle \\ \langle B, [s_3, \{\{B\}\}], \{(B)\} \rangle \\ \langle C, [s_1, \{\{C\}\}], \{(C)\} \rangle \end{array}$$

また、選言関係 $choose\{A, B\}$ が宣言された場合、

$$schoose[s_3, \{A, B\}]$$

として選言データベースに記録される。さらに理由付けとして、

$$\begin{array}{l} A \Rightarrow a \\ B \Rightarrow b \\ C \Rightarrow c \\ a, c \Rightarrow \perp \\ b, c \Rightarrow \perp \end{array}$$

が与えられた場合、以下に示す5つのノードが生成される。

$$\begin{array}{l} \langle a, [s_2, \{A\}], \{(A)\} \rangle \\ \langle b, [s_3, \{B\}], \{(B)\} \rangle \\ \langle c, [s_1, \{C\}], \{(C)\} \rangle \\ \langle \perp, [s_2, \{A, C\}], \{(a, c)\} \rangle \\ \langle \perp, [s_3, \{B, C\}], \{(b, c)\} \rangle \end{array}$$

矛盾を表現する2つのノードは、

$$\begin{array}{l} snogood[s_2, \{A, C\}] \\ snogood[s_3, \{B, C\}] \end{array}$$

として矛盾データベースに記録される。このとき論理的には、

$$\frac{\begin{array}{l} s_3 \models A \vee B \\ s_2 \models \neg A \vee \neg C \\ s_3 \models \neg B \vee \neg C \end{array}}{s_3 \models \neg C}$$

のように状況 s_3 で $\neg C$ が導出される。ところが、Hyperresolutionを行なわない SATMS では、

$snogood[s_2, \{A, C\}]$ と $snogood[s_3, \{B, C\}]$ が導出されるのみであり, $snogood[s_3, \{C\}]$ は導出されない。すなはち, 本来矛盾であるはずの状況依存環境 $[s_3, \{C\}]$ を持つノードが存在することになり, ラベル計算アルゴリズムの無矛盾性が保証されない。そこで, 上記の Negative Hyperresolution を導入することにより, ラベルの無矛盾性を維持する。

$$\frac{\begin{array}{l} choose[s_3, \{A, B\}] \\ snogood[s_2, \{A, C\}] \\ snogood[s_3, \{B, C\}] \end{array}}{snogood[s_3, \{C\}]}$$

【完全性の維持】

ラベルの完全性とは, 論理的に導出されるべきラベルがすべてラベル計算アルゴリズムにより得られることを言う。選言の導入により失われる完全性を維持するために, 以下に示す Hyperresolution を導入する必要がある。

$$\frac{\begin{array}{l} choose[s, \{A_1, A_2, \dots\}] \\ < \beta, \lambda, \delta > \\ \forall_i snogood[s_i, \{A_i\} \cup \alpha_i] \text{ または } [s_i, \{A_i\} \cup \alpha_i] \in \lambda \\ \text{ただし, } \forall_j A_j \notin \alpha_i \end{array}}{< \beta, \{[s_{sup}, \bigcup_i \alpha_i] \cup \lambda^*}, \delta >}$$

ただし, $s_{sup} = sup[\{s\} \cup \bigcup_i \{s_i\}]$
また, λ^* は λ から $[s_{sup}, \bigcup_i \alpha_i]$ のすべての上位状況依存環境を除いたもの

ここで, 状況 s_{sup} が存在しない場合には, この Hyperresolution は起動されない。

【例 3.3】ラベルの完全性

状況として s_1, s_2 を考え, 以下に示す順序関係が成り立つものとする。

$$s_1 \preceq s_2$$

ここで, 状況 s_1 で仮説 A が, 状況 s_2 で仮説 B が宣言されたとき, 以下に示す SATMS ノードが生成される。

$$\begin{array}{l} < A, [s_1, \{\{A\}\}], \{(A)\} > \\ < B, [s_2, \{\{B\}\}], \{(B)\} > \end{array}$$

また, 選言関係 $choose[A, B]$ が宣言された場合,

$$choose[s_2, \{A, B\}]$$

として選言データベースに記録される。さらに理由付けとして,

$$\begin{array}{l} A \Rightarrow a \\ B \Rightarrow b \\ a \Rightarrow c \end{array}$$

$$b \Rightarrow c$$

が与えられた場合, 以下に示す 3 つのノードが生成される。

$$\begin{array}{l} < a, \{[s_1, \{A\}]\}, \{(A)\} > \\ < b, \{[s_2, \{B\}]\}, \{(B)\} > \\ < c, \{[s_1, \{A\}], [s_2, \{B\}]\}, \{(a), (b)\} > \end{array}$$

このとき論理的には,

$$\begin{array}{l} s_1 \models \neg A \vee a \\ s_2 \models \neg B \vee b \\ s_1 \models \neg a \vee c \\ s_2 \models \neg b \vee c \\ s_2 \models A \vee B \\ \hline s_2 \models c \end{array}$$

のように状況 s_2 で c が導出される。

ところが, Hyperresolution を行なわない SATMS では,

$$< c, \{[s_1, \{A\}], [s_2, \{B\}]\}, \{(a), (b)\} >$$

となるのみであり,

$$< c, \{[s_1, \{A\}], [s_2, \{\}]\}, \{(a), (b)\} >$$

とはならない。すなはち, 本来導出されるべき状況依存環境 $[s_2, \{\}]$ が計算されないことになり, ラベル計算アルゴリズムの完全性が保証されない。そこで, 上記の Hyperresolution を導入することにより, ラベルの完全性を維持する。

$$\frac{\begin{array}{l} choose[s_2, \{A, B\}] \\ < c, \{[s_1, \{A\}], [s_2, \{B\}]\}, \{(a), (b)\} > \\ [s_1, \{A\} \cup \{\}] \in \{[s_1, \{A\}], [s_2, \{B\}]\} \\ [s_2, \{B\} \cup \{\}] \in \{[s_1, \{A\}], [s_2, \{B\}]\} \end{array}}{< c, \{[s_1, \{A\}], [s_2, \{\}]\}, \{(a), (b)\} >}$$

4 例題

4.1 例題 1: 副問題への分割

ある問題が n 個の副問題 $P_1, P_2, \dots, P_n$ に分割可能であり, 副問題 P_i の探索空間はそれぞれ m_i 個の仮説 (AS_i) により表現されるものとする。まず, 副問題 P_i を状況 s_i に対応させる。それぞれの副問題を解く場合, 任意の状況 s_i と $s_{j \neq i}$ の間には順序関係が与えられていないため, SATMS は与えられた理由付けからラベル計算を行なうときに, 状況 s_i の仮説と状況 $s_{j \neq i}$ の仮説との間のラベル計算を行なう必要がない。探索のとき最悪の場合に考慮しなければならない環境の数の ATMS と SATMS との比較を以下に示す。

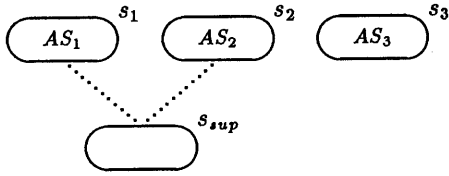


Fig. 3: Example 1

$$\text{ATMS} : 2^{\sum_i m_i}$$

$$\text{SATMS} : \sum_i 2^{m_i}$$

例えば、 $n = 3$ かつ $m_1 = m_2 = m_3 = 3$ の場合、ATMS が 512 個の環境からなる探索空間を考慮しなければならないのに対して、SATMS では 24 個の状況依存環境からなる探索空間を考慮すれば良い。

さて、不要な仮説の組合せを回避する方法としては、仮説間に制約を記述する方法も考えられる。この問題の場合には、副問題間の仮説の組合せをすべて回避するために矛盾として宣言する必要がある、最終的に記録される矛盾の数は、

$$(\sum_i \sum_{j \neq i} m_i \times m_j) / 2$$

となる。前述の例では矛盾の数は 27 個となり、ラベル計算時に余分な矛盾の検査が必要となる。

また、制約条件の存在などにより副問題が互いに従属関係にある場合、副問題を結合して制約条件の検査を行なうことも可能である。SATMS を用いる場合には、従属関係にある副問題を表現する状況の集合 S_{sub} に対して、状況 $s_{sup}(= sup S_{sub})$ を導入することにより、状況 s_{sup} において制約条件の検査を行なうことができる (Fig.3)。

4.2 例題 2: 矛盾の追加

仮説の集合 $AS = \{A_1, A_2, \dots, A_n\}$ を考える。ここで、ある状況では A_i が矛盾となり、さらにある状況では A_i および A_j が矛盾になるとする。これは SATMS では、3 つの状況 s_0, s_1, s_2 と状況間の順序関係として、

$$s_0 \preceq s_1, s_1 \preceq s_2$$

を導入することにより、

$$s_1 \models \neg A_i, s_2 \models \neg A_j$$

と表現される。実際には、矛盾状況依存環境として、

$$snogood[s_1, \{A_i\}], snogood[s_2, \{A_j\}]$$

が記録される (Fig.4)。

これに対して ATMS では、状況を導入するかわりに状況に相当する 2 つの仮説 S_1, S_2 を新たに導入することにより、矛盾環境として、

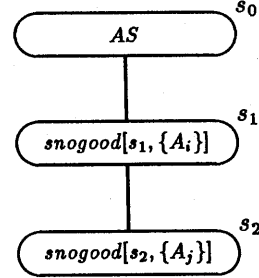


Fig. 4: Example 2

$nogood\{S_1, A_i\}, nogood\{S_2, A_i\}, nogood\{S_2, A_j\}$ が記録される。また ATMS では、仮説の集合 $\{A_1, A_2, \dots, A_n\}$ とこれらの仮説の真偽値を決定するために導入された仮説の集合 $\{S_1, S_2\}$ とが等しく扱われ、本来の仮説の集合により構成される探索空間が不明確となる。また、新たに 2 つの仮説を導入したことにより、導入前の 4 倍の環境を考慮しなければならない。

5 考察

5.1 状況理論との比較

自然言語では文脈に依存して文の意味が異なる場合が多い。このような文の意味の文脈依存性を扱うための意味論として Barwise と Perry により状況意味論 [17]、およびこれを支える理論として状況理論 [18] が提案されている。状況理論を論理的に解釈すると、状況に依存して命題 (インフォンの) 真理値 (極性) が変化すると捉えることができる。SATMS において仮説の集合が空集合の場合が状況理論に対応する。状況理論との相違点としては、SATMS ではすべての論理式が命題論理の枠組みで取り扱われることである。また、命題 p が状況 s_i で成立するという、

$$s_i \models p$$

を SATMS ではノードとして、

$$\langle p, \{[s_i, \{\}]\}, \dots \rangle$$

のように、状況が明示的に命題の内容として表現されていないために、

$$s_j \models (s_i \models p)$$

のような論理式を表現できないことである。また、状況理論においては状況間の順序関係はインフォンの包含関係により定義される。従って、順序関係の存在する 2 つの状況が存在する場合、下位の状況

で成立するインフォンは上位の状況でも必ず成立する。これに対してSATMSでは状況間に任意の順序構造を定義することができる。現在のSATMSは定義2に示したような単調な推論規則を用いているために、状況理論と同様に下位の状況で成立する命題は上位の状況でも必ず成立する。しかしながら、次節の時間の表現で述べるような非単調な推論規則を用いる場合には、この関係は必ずしも成立しない。

5.2 時間の表現

例題2のような状況の集合は、それぞれの状況がある時点に対応付けることにより、時間的な変化を表現することも可能である。ただし、定義2の推論規則は状況の順序に関する単調性を保証している。すなわち、ある状況で成立する命題はその上位の状況でも必ず成立する。従って、時間的な変化により非単調となる命題を表現するためには、現在の推論規則では不十分であり、状況の順序に関して非単調な命題を扱えるように推論規則を拡張することが必要である。このような非単調な推論規則としては、

$$\begin{array}{l} s_i \preceq s_j \\ s_i \models p \\ : Ms_j \models p \\ \hline s_j \models p \end{array}$$

というデフォルト論理風の推論規則などのように、 s_j において p の成立を否定されていない限り p は成立するというものが考えられる。

例えば、

$$p \wedge q \rightarrow \perp$$

であり、

$$\begin{array}{l} s_1 \preceq s_2 \\ \langle p, \{[s_1, \{\}]\}, \dots \rangle \\ \langle \neg p, \{[s_2, \{\}]\}, \dots \rangle \\ \langle q, \{[s_1, \{\}]\}, \dots \rangle \end{array}$$

の場合、状況 s_1 では p と q が成立するため矛盾となり、

$$snogood[s_1, \{\}]$$

が矛盾データベースに記録される。ところが、状況 s_2 では $\neg p$ が成立するため、推論規則から p が導出されない。従って、状況 s_2 は矛盾とはならず、矛盾にも非単調性を導入する必要がある。このため矛盾データベースを極小な矛盾として管理することができず効率的な実現方法が課題となる。

6 まとめ

本論文では、状況という概念を導入することにより、仮説が導入された状況を扱えるようにde KleerのATMSを拡張したSATMSを提案した。SATMSは探索空間の縮小をはかるための柔軟な枠組みを提供するとともに、従来の多重コンテキストによる推論とATMSを用いた仮説推論とを包含した枠組みを与えている。SATMSにおいて、仮説の集合が空である場合が多重コンテキストによる推論に対応する。また、状況がただひとつしか存在しない場合はATMSに一致する。

今後の課題としては、SATMSを論理的に定式化して状況理論などとの関連を明確にすること、推論規則を拡張することにより時間の表現を可能とすること、および具体的な問題に適用して有効性を実現することなどがある。

謝辞 本研究をまとめるにあたり有益なコメントを下さり、快く議論に応じて下さったICOT第5研究室の井上克己氏、本研究の機会を与えて下さった知能工学研究部の田村秀行部長に深く感謝致します。

参考文献

- [1] de Kleer, J.: An assumption-based TMS, *Artificial Intelligence* 28, pp.127-162 (1986)
- [2] Sussman, G.J. and McDermott, D.V.: From PLANNER to CONNIVER - A genetic approach, *Proc. Fall Joint Comp. Conf.*, pp.1171-1179 (1972)
- [3] 董, 中川: assumptionとsituation, 日本ソフトウェア科学会第4回大会論文集E-1-1 (1987)
- [4] de Kleer, J. and Williams, B.C.: Back to backtracking: Controlling the ATMS, *Proc. AAAI-86*, pp.910-917 (1986)
- [5] 井上克己: 導出を用いた仮説探索, 情報処理学会第35回全国大会1M-2 (1987)
- [6] Forbus, K.D. and de Kleer, J.: Focusing the ATMS, *Proc. AAAI-88*, pp.193-198 (1988)
- [7] 西岡, 堀, 池田, 溝口, 角所: 言語処理システムにおけるATMSの利用法について, 情報処理学会第37回全国大会6C-7 (1988)

- [8] 董, 中川: 不確実な知識における ATMS, 情報処理学会知識工学と人工知能研究会資料 52-1 (1987)
- [9] 劉, 西田, 堂下: 不確実な情報による ATMS の制御, 日本ソフトウェア科学会第4回大会論文集 E-1-3 (1987)
- [10] 青山, 石塚: 仮説の重み利用による高速化機構を備えた仮説推論システム, 人工知能学会第2回全国大会 2-6 (1988)
- [11] D'Ambrosio, B.: Truth maintenance with numeric certainty estimates, *Proc. 3rd Conference on Artificial Intelligence Applications*, pp.244-249 (1987)
- [12] Laskey, K.B. and Lehner, P.E.: Belief maintenance: An integrated approach to uncertainty management, *Proc. AAAI-88*, pp.210-214 (1988)
- [13] de Kleer, J.: Extending the ATMS, *Artificial Intelligence 28*, pp.163-196 (1986)
- [14] 飛鳥井, 村田, 森沢, 浅野: エキスパートシステム構築ツール CHORUS (2) - 仮説推論機能 -, 情報処理学会第37回全国大会 5G-7 (1988)
- [15] 矢沢, 篠原: 仮説に基づく推論における機械学習方式について, 電子情報通信学会人工知能研究会資料, AI88-28 (1988)
- [16] de Kleer, J.: A general labeling algorithm for assumption-based truth maintenance, *Proc. AAAI-88*, pp.188-192 (1988)
- [17] Barwise, J. and Perry, J.: *Situation and Attitudes*, MIT Press (1983)
- [18] 向井国昭: 数学的側面からみた状況理論, ICOT Technical Report TM-622 (1988)