

多重世界機構における Circumscription†

森 辰 則** 中 川 裕 志**

人間の常識による推論を表現するための様々なアプローチの中で J. McCarthy の提唱した Circumscription と呼ばれる推論方法は有望なものの一つである。常識による推論を非単調論理により取り扱うシステムは新しく得られた知識により従来の知識を覆すことが許されているため学習としての側面を持つといえるが、Circumscription は対象を一階述語論理とした推論規則であるために知識を記述するための特別な論理体系を必要としないという特徴を持つ。しかし、Circumscription の対象となる一階述語論理では知識表現で重要視される「知識の構造化」を効率良く実現できないという欠点がある。本稿では、以上の点を踏まえ、Circumscription の対象として Prolog の定義空間を多重化した電総研中島氏による Uranus の多重世界機構を用いて表現された階層構造を持つ知識を扱う方法について考え、Circumscription による概念の再構成の方法について考察する。その結果、多重世界による知識の階層化は例外事象を表す abnormal 述語の最小化の優先順位決定にも利用できることがわかった。また、この方法によれば各世界内の知識と世界間の階層構造を Circumscription という統一した方法で取り扱うことができる。

1. はじめに

人間の常識による推論を表現するための様々なアプローチの中でも J. McCarthy の提唱した Circumscription^{1), 2)} と呼ばれる推論方法は有力なものの一つである。Circumscription は古典論理の枠組みを越えることなく、人間が思考の過程で無意識に行っていると思われる「推論の対象範囲を限定する」ことをシミュレートしている。Circumscription の推論規則自体は一階述語論理の枠を越えているので機械的な推論では実現できなかったが、V. Lifschitz の研究^{3), 4)} により、ある限定された範囲においては機械的に計算可能なことが解明されたことで、Circumscription を計算機上に実現する糸口は示された。

常識による推論を非単調論理により取り扱うシステムは、新しく得られた知識により従来の知識を覆すことが許されているため、学習としての側面を持つと言える。非単調論理の体系としては、Circumscription のほかには、D. McDermott, J. Doyle の Non-monotonic logic¹²⁾ や R. Reiter の default logic¹³⁾ などがあげられるが、いずれも論理的整合性を表す新しい記号 M を古典論理の中に導入することにより非単調性を表現している。これに対して Circumscription は対象を一階述語論理としているために知識を記述するための特別な論理体系を必要としないという特徴を持つ。

ところが、一階述語論理は公理の存在する空間が単

一であって知識表現で重要視される「知識の構造化」を効率よく実現できないという欠点がある。我々が日常行っている推論の効率の良さは、思考の対象を適度に限定していることに由来していると思われる。これは人間の知識の形態がある程度のまとまりを持っており、しかもそれぞれの知識が構造化されているためであると考えられる。それゆえ、知識をモジュール化し、そのモジュール間の構造を明確に表現することが必要である。

本稿では、Circumscription の対象として、Prolog の定義空間を多重化した電総研中島氏による Uranus⁵⁾ の多重世界機構を用いて表現された階層構造を持つ知識を扱う方法について考え、Circumscription を概念学習の Tool として応用する方法について考察する。例外事象を表す abnormal 述語の最小化の優先順位として概念の特殊性が利用できることは、V. Lifschitz⁴⁾ により指摘されているが、多重世界による知識の階層化が概念の特殊性を明確にし、abnormal 述語の最小化の優先順位決定に役立つことがわかった。すなわち概念階層の上下関係を利用することにより優先順位を決定できる。また、この方法によれば各世界内の知識と世界間の階層構造を Circumscription という統一した方法で取り扱うことができる。

なお、システム全体の構成は次のようになっている。まず、ユーザから指定された goal に関して無矛盾性のチェックを行い、矛盾が発見されると、これを解消するために abnormal 述語を導入する。そして、abnormal 述語が、いくつか導入された時点で Circumscription を行い、abnormal 述語を消去、概念階層の再構成を行う。

† Computable Circumscription in the Multiple World Mechanism by TATSUNORI MORI and HIROSHI NAKAGAWA (Department of Electrical and Computer Engineering, Faculty of Engineering, Yokohama National University).

** 横浜国立大学工学部電子情報工学科

2. 矛盾の発見による abnormal 述語の導入

本研究で使用した言語は Uranus をベースとしているが「否定」として Prolog 等で通常利用されている Negation as failure⁶⁾ (これを述語 THNOT で表す。)のほかに、陽に否定知識を表現するための論理否定 NOT を導入している^{7),8)}。このため、ある知識に関する肯定と否定が証明される矛盾が発生することがある。一般に論理型言語に negation as failure でない論理的否定を導入し矛盾の発生を監視するメカニズムを組み込むと効率上問題があることが知られている。しかし、矛盾の発見は新しい知識獲得の契機として位置づけられ、推論上重要な役割を果たしているため、論理的否定を導入することは不可欠であると考えられる。

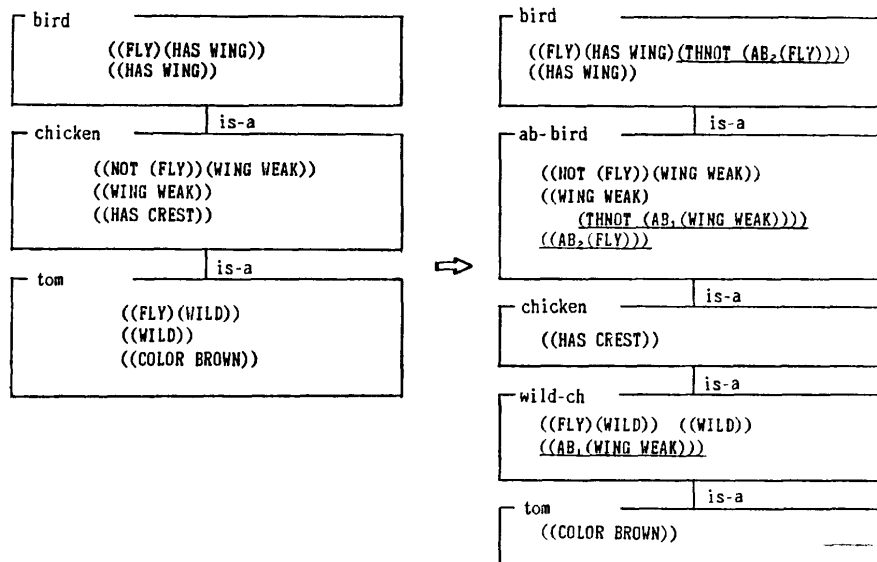
後向き推論を行う Uranus のような論理型言語では矛盾を発見するためには、世界の階層が決定された上で、まずその goal 自体を実行し、それが証明された後にその goal の否定が証明されるかどうかチェックするという方法を用いる。しかし、これを証明過程に現れる、すべての述語について行うと多大なメモリ空間と時間を要し、組み合わせの爆発を起こす。このため、現在、我々は consistency check を行う goal を選択的に指定できるように Uranus にチェック述語 C^{7),8)}を導入しているが、どの述語のチェックを行うかはユーザに委ねている。効果的に矛盾を発見する

ためには矛盾のチェックをどのように行えばよいかという問題は、今後、解決すべき課題である。

ここで、矛盾の解消のプロセスを考えてみる。まず証明された肯定・否定の知識のうち、どちらが(矛盾の発生した世界で)誤りなのかを教師でもあるユーザに選択してもらう。次に矛盾の原因を探るために誤った知識を導出する過程で用いられた知識をユーザとの対話により調べる。その中の一つは必ず矛盾の発生した世界にとって例外的な知識となっている。矛盾の解消のためには、この原因となる知識がその世界では例外であることを示し、その世界では成立しないようにすればよい。これには、McCarthy が Circumscription の対象を統一して表現するために導入した例外事象を表す abnormal 述語を図 1 のように用いることが考えられる。すなわち、矛盾原因の探索法は TMS⁹⁾ の持つ Dependency Directed Backtracking (DDB) と同様だが、矛盾の解消法は abnormal 述語によって行っている点に特徴がある。なお、今後、プログラムの記法は Uranus に準じ S 式表示を用いる。

図 1 (b) において、世界 ab-bird および wild-ch は矛盾解消プロセスがユーザの指示により生成した新しい世界を想定している。これは『矛盾の発見が新概念導入の契機となり得る。』という考えのもとに附加している機能で、矛盾解消機構として本質的なものではない。

以上のように、現在のところ、矛盾のチェック、



(a) tom および chicken で述語 FLY が矛盾

(b) 矛盾解消後

図 1 abnormal 述語による矛盾解消

Fig. 1 The transformation into a consistency knowledge using abnormal predicates.

abnormal 述語を付加する clause の選択および新世界の導入はユーザに委ねているのだが、いずれも解決すべき問題である。以後の議論は、以上のような過程を経て図 1 (a) から (b) が得られるものとして議論を進めることにする。

3. 属性の継承のある階層的な知識表現における Circumscription

矛盾の発見を契機に例外情報を知識に取り込むという方法を第 2 章で提案した。しかし、人間の推論はそこでとどまらず、さらに一歩進んで例外の発見と同時に、それが何であるかを認識していると思われる。Circumscription は正に、この過程をシミュレートしている。

本章では階層構造を持つ知識表現において Circumscription を行う方法について述べる。

3.1 Circumscription

本節では参考文献 4) をもとに Circumscription の定式化、論理型言語への応用について説明するが、詳細は参考文献 1)~4), 10) を参照していただきたい。

AB を述語定数の組、 Z を AB に含まれない関数や述語定数の組とすると、論理式 $A(AB, Z)$ の下で Z の変化を許した場合の AB の Circumscription を $\text{Circum}[A(AB, Z); AB; Z]$ と表し、その定義は文献 4) によれば次のようになる。

$$\begin{aligned} \text{Circum}[A(AB, Z); AB; Z] \\ = A(AB, Z) \wedge \neg \exists ab \exists z. [A(ab, z) \wedge (ab < AB)] \end{aligned} \quad (3.1)$$

ここで、記号 \leq は \supset (含意) に等しく、記号 $<$ は \leq から同値の場合を除いた関係である。また束縛変数 ab, z はそれぞれ AB, Z と同じ形の述語の記号を値とする述語変数である。また、 Z が空の場合は Z を省略して記す。

Circumscription は separable と呼ばれる一階述語論理の部分系に対しては機械的に計算可能であることが V. Lifschitz により示された^{3), 4)}。Prolog などのような論理型言語では Horn 節の連言をプログラムと見なしているので、separable formulas となっている場合には、その特別な形で次のように定義される solitary formulas となる。

AB を述語定数の組とすると、論理式 $A(AB)$ が AB に関して solitary であるとは次のような形をしている場合である。

$$N(AB) \wedge \{U \leq AB\} \quad (3.2)$$

$N(AB)$ は AB が正のリテラルとして出現しない論理式であり、 U は AB を含まない述語の組である。このとき

$$\begin{aligned} \text{Circum}[N(AB) \wedge \{U \leq AB\}; AB] \\ \equiv N(U) \wedge \{U = AB\} \end{aligned} \quad (3.3)$$

が成り立つ⁴⁾。 Z が空でない場合には次式が成立するので空な場合に帰着できる。

$$\begin{aligned} \text{Circum}[A(AB, Z); AB; Z] \\ \equiv A(AB, Z) \wedge \text{Circum}[\exists z. A(AB, Z); AB] \end{aligned} \quad (3.4)$$

ここで再び二階の変数が現れるが $A(AB, z)$ が z に関して solitary, すなわち次式によって表される場合、

$$A(AB, z) \equiv [Nn(AB, z) \quad (3.5a)$$

$$\wedge \{Uu(AB) \leq z\}] \quad (3.5b)$$

次式が成立して、二階の限量子記号は消去できる。

$$\exists z. A(AB, z) \equiv Nn(AB, Uu(AB)) \quad (3.6)$$

具体的には、(3.5b) から得られた z の extension $Uu(AB)$ (Prolog では clause の body に相当) を (3.5a) の z の出現している部分に代入する。これは、Prolog におけるプログラム変換に対応させると、いわゆる unfold¹¹⁾ により (3.5a) 中の goal z を (3.5b) の clause を用いて部分評価することにより正当性を失わずに述語 z を消去していることにあたる¹⁰⁾。

また、それぞれの述語の最小化の結果が互いに矛盾する場合など abnormal 述語のいくつかに最小化に関する優先順位を設けたいことがある。この場合は、prioritized circumscription を用いる。

$$\text{Circum}[A; AB^1 > \dots > AB^k; Z] \quad (3.7)$$

は添字の小さい abnormal 述語が優先されて最小化されることを表す。prioritized circumscription は次のように circumscription の連言によって表すことができる⁴⁾ので実際の計算にはこれを用いる。

$$\begin{aligned} \text{Circum}[A; AB^1 > \dots > AB^k; Z] \\ \equiv \bigwedge_{i=1}^k \text{Circum}[A; AB^i; AB^{i+1}, \dots, AB^k, Z] \end{aligned} \quad (3.8)$$

3.2 世界に関する情報の導入

Circumscription は単一世界に定義された一階述語論理式に対する推論規則である。このため多重世界機構を用いて表現された知識において、ある述語を circumscribe する場合には、一度、単一な世界の知識に表現しなおす必要がある。

そこで、ある概念世界に知識が assert されているとはどのように解釈できるか、その意味を考えてみ

$$(HAS\ WING) \wedge (THNOT(AB_1(FLY)))$$

$$\quad \wedge \underline{bird} \leq (FLY) \quad (3.10)$$

$$\underline{bird} \leq (HAS\ WING) \quad (3.11)$$

$$(WING\ WEAK) \wedge \underline{ab-bird} \leq (NOT(FLY))$$

$$\quad (3.12)$$

$$(THNOT(AB_1(WING\ WEAK)))$$

$$\quad \wedge \underline{ab-bird} \leq (WING\ WEAK) \quad (3.13)$$

$$(WILD) \wedge \underline{wild-ch} \leq (FLY) \quad (3.14)$$

$$\underline{ab-bird} \leq \underline{bird}$$

$$\underline{chicken} \leq \underline{ab-bird}$$

$$\underline{wild-ch} \leq \underline{chicken} \quad \text{etc.}$$

図2 多重世界による知識から単一世界の知識への変換
Fig. 2 The transformation of a multiple world knowledge base into a single world knowledge base.

る。例えば図1のように世界 bird に ((FLY)...) という知識が assert されている場合, ((FLY)...) は世界 bird とその下位の世界 ab-bird ... において成立する。これは ((FLY)...) が「bird に存在する」という事実を前提に持つことを示している。また、属性の継承については世界の階層の上下関係を含意関係と考えることができるので次のような clause が成立するものと解釈できる。

$$\underline{ab-bird} \leq \underline{bird}, \underline{chicken} \leq \underline{ab-bird} \quad \text{etc.} \quad (3.9)$$

以上の考察により、多重世界に assert された知識を次のようにして単一の世界の知識へと変換する。

- ① すべての clause についてその clause の定義されている世界名を前提条件の一つとして body に加え、単一の知識ベースとする。この clause の意味は「すべての前提 (body) が成立し、かつ示された世界に存在する時に帰結 (head) が成立する。」である。
- ② 世界の階層構造を (3.9) のように含意で表現し、知識ベースに加える。

この手続きによれば、図1の知識構造は図2の一階述語論理の体系と等価だと見なせる。(一部省略)

3.3 Circumscription に必要な情報の抽出

前節で述べた方法で単一世界へと変換した知識を circumscribe するためには clause に付加する世界の情報のほかに次のものが必要である^{4),10)}。

- ① circumscribe する abnormal 述語の名前
- ② abnormal 述語を最小化する際の優先順位
- ③ Circumscription の過程で変化を許す述語の名前

①に関しては矛盾の解消時に abnormal 述語を導入するのでチェックするのは簡単であるが他の二つはヒューリスティックな情報を利用しなければならない。

3.3.1 abnormal 述語の最小化の優先順位

abnormal 述語の最小化の優先順位は circumscribe 後の abnormal 述語の定義に影響を与える重要な情報である。この優先順位として fact 型の abnormal 述語の概念階層上の位置を考慮することができる。すなわち、下位の世界に fact 型 abnormal 述語が assert されている abnormal 述語ほど最小化の優先順位が高いとする。fact 型の abnormal 述語が、その関連する矛盾の発生した最も上位の世界を表しているため、より下位の世界に assert されている abnormality ほどその効力の及ぶ範囲に属する概念が少ないことになるので、優先して最小化されるべきであるとするのは自然であろう。例えば、図1の例では述語 AB_1 を AB_2 よりも優先して最小化する。

3.3.2 Circumscription 時に変化を許す述語の選択

Circumscription 時に変化を許す述語には Circumscription により特徴づけたい述語を選択するが、ここでは2.4節で述べた矛盾解消作用を重視する。Circumscription の矛盾解消作用を用いるためには、まず

- ① 矛盾の発生した述語 (直接矛盾を引き起こした clause の述語)

を変化を許す述語に加えなければならない。例えば、図1では述語 FLY が①に相当する。また abnormal 述語が直接矛盾を起こした clause ではなく、矛盾の証明過程に現れる他の clause に付加されている場合には

- ② abnormal 述語が付加された clause から直接矛盾を起こした clause までの証明過程に現れる述語

も変化を許す述語に加える必要がある。述語の消去による矛盾解消効果を abnormal 述語の定義の決定に反映させるためには abnormal 述語を求める時に見かけ上、矛盾の直接の原因となった clause に abnormal 述語が現れるようにしなければならない。そのためには矛盾の直接の原因となった clause の subgoal の一つを unfold により部分評価するという操作を繰り返して subgoal に abnormal 述語が現れるようにすればよい。この効果を得るために変化を許す述語を②のように設定する。これは、3.1節で述べたように、変化を許す述語を消去する過程が unfold と等価であるからである。

例えば図1の世界 ab-bird においては abnormal 述語を矛盾の発生した述語 FLY ではなく、その矛盾

$$(\text{WING WEAK}) \wedge \text{ab-bird} \leq (\text{NOT (FLY)}) \quad (3.12)$$

$$\begin{array}{c} \uparrow \text{unfold} \\ (\text{THNOT (AB}_1 \dots)) \wedge \text{ab-bird} \leq (\text{WING WEAK}) \quad (3.13) \\ \downarrow \end{array}$$

$$(\text{THNOT (AB}_1 \dots)) \wedge \text{ab-bird} \leq (\text{NOT (FLY)})$$

図 3 abnormal 述語が矛盾の証明過程の途中に付加されている場合

Fig. 3 The case that an abnormal predicate is attached to an inconsistency checking process.

の証明過程中に存在する述語 WING に付加している。この場合、図 3 のように述語 WING を unfold することにより見かけ上、述語 FLY の body に abnormal 述語が現れるので矛盾を起こした述語 FLY を消去する際に、その効果を abnormal 述語の定義に反映させることができる。

よって、図 1 の例の場合には Circumscription の過程で変化を許す述語として FLY のほかに述語 WING を選択する。

3.4 Circumscription の結果に基づく世界の再構成

図 2 の horn 節の連言を論理式 A とすると図 1 の世界構造を circumscribe することは、次式と等価である。

$$\text{Circum}[A; \text{AB}_1 > \text{AB}_2; \text{FLY}, \text{WING}] \quad (3.15)$$

すなわち、述語 AB_1, AB_2 を論理式 A の下で circumscribe する。この時、より下位の世界の abnormal 述語 AB_1 を優先して最小化し、その過程で述語 FLY および WING の変化を許す。これを計算するためには、まず、prioritized circumscription を (3.8) を用いて circumscription の連言に変換し、次いで、Circumscription の過程で変化を許す述語を (3.4) を用いて消去する。

$$\begin{aligned} & \text{Circum}[A; \text{AB}_1 > \text{AB}_2; \text{FLY}, \text{WING}] \\ & \equiv A \wedge \text{Circum}[\exists \text{AB}_2 \exists \text{FLY} \exists \text{WING}. A; \text{AB}_1] \\ & \quad \wedge \text{Circum}[\exists \text{FLY} \exists \text{WING}. A; \text{AB}_2] \end{aligned} \quad (3.16)$$

次に、(3.6) により二階の述語変数 (ここでは、FLY, WING, AB_2) を消去する。(3.16) において、

$$\begin{aligned} & \exists \text{FLY} \exists \text{WING}. A \\ & \equiv \exists \text{FLY} \exists \text{WING}. [(\text{HAS WING}) \\ & \quad \wedge (\text{THNOT (AB}_2 (\text{FLY}))) \wedge \text{bird} \leq (\text{FLY}) \\ & \quad \wedge \{(\text{WILD}) \wedge \text{wild-ch} \leq (\text{FLY})\} \\ & \quad \wedge \{(\text{WING WEAK}) \\ & \quad \wedge \text{ab-bird} \leq (\text{NOT (FLY)})\} \end{aligned}$$

$$\begin{aligned} & \wedge \{(\text{THNOT (AB}_1 (\text{WING WEAK}))) \\ & \quad \wedge \text{ab-bird} \leq (\text{WING WEAK})\} \wedge \text{Rest}] \end{aligned} \quad (3.17)$$

ただし、説明に不用な論理式はすべて論理式 Rest に含まれるものとする。変化を許す述語を消去する unfold の準備のために (3.17) を変形する。

$$\begin{aligned} & (\text{WING WEAK}) \wedge (\text{THNOT (NOT (FLY))}) \\ & \quad \wedge \text{ab-bird} \leq (\text{FALSE}) \end{aligned} \quad (3.18)$$

上式のように含意記号を越えてリテラルを移動させると、そのリテラルに THNOT が付加されるので (THNOT (NOT (FLY))) というリテラルが現れる。これは、(FLY) の論理否定が証明できなければ真となる様相演算子である。一般に、系が無矛盾な場合には、(FLY) が真である領域は (THNOT (NOT (FLY))) が真である領域に包含されていなければならない。そこで、変化を許す述語の消去時に、この (THNOT (NOT (FLY))) を (FLY) と同等に扱って unfold することにより述語 FLY に関しての無矛盾性を保つことができるのである¹⁰⁾。Circumscription の矛盾解消作用を利用するためには変化を許す述語として矛盾の発生した述語を加えればよいことになる。述語 FLY の無矛盾性を保つために (3.18) を次式とする。

$$(\text{WING WEAK}) \wedge (\text{FLY}) \wedge \text{ab-bird} \leq (\text{FALSE}) \quad (3.19)$$

論理式 A の中で述語 FLY, WING を unfold して消去し、世界名 subgoal を 3.2 節で述べた含意関係を利用して、まとめると次のようになる。

$$\begin{aligned} & \exists \text{FLY} \exists \text{WING}. A \\ & \equiv [(\text{THNOT (AB}_1 (\text{WING WEAK}))) \\ & \quad \wedge \text{ab-bird} \wedge (\text{HAS WING}) \\ & \quad \wedge (\text{THNOT (AB}_2 (\text{FLY}))) \wedge \text{bird} \leq (\text{FALSE})] \\ & \quad \wedge [(\text{THNOT (AB}_1 (\text{WING WEAK}))) \\ & \quad \wedge \text{ab-bird} \wedge (\text{WILD}) \\ & \quad \wedge \text{wild-ch} \leq (\text{FALSE})] \quad \wedge \text{Rest} \\ & \equiv [(\text{HAS WING}) \\ & \quad \wedge (\text{THNOT (AB}_1 (\text{WING WEAK}))) \\ & \quad \wedge \text{ab-bird} \leq (\text{AB}_2 (\text{FLY}))] \\ & \quad \wedge [(\text{WILD}) \wedge \text{wild-ch} \\ & \quad \leq (\text{AB}_1 (\text{WING WEAK}))] \quad \wedge \text{Rest} \end{aligned}$$

また、 $\exists \text{FLY} \exists \text{WING}. A$ において AB_2 は第 1 項目だけに含まれるので、

$$\begin{aligned} & \exists \text{AB}_2 \exists \text{FLY} \exists \text{WING}. A \\ & \equiv \exists \text{AB}_2. [(\text{THNOT (AB}_1 (\text{WING WEAK}))) \\ & \quad \wedge (\text{HAS WING}) \wedge \text{ab-bird} \leq (\text{AB}_2 (\text{FLY}))] \end{aligned}$$

$$\begin{aligned} & \wedge \{(WILD) \wedge \text{wild-ch} \\ & \leq (AB_1 (WING WEAK))\} \\ & \wedge \text{Rest} \end{aligned} \quad (3.20)$$

となるが、(3.20)の第1項は、

$$(AB_2(FLY)) = (\text{TRUE})$$

の時 [] 内が恒真式となるので、恒真である。よって、(3.20)の AB_2 の存在限量を消去することができて次のようになる。

$$\begin{aligned} & \exists AB_2 \exists FLY \exists WING. A \\ & \equiv [(WILD) \wedge \text{wild-ch} \\ & \leq (AB_1 (WING WEAK))] \wedge \text{Rest} \end{aligned}$$

つまり上位世界の abnormal 述語 AB_2 が成立した状況で AB_1 を計算することに等しい。(3.3)を用いると、

$$\begin{aligned} & \text{Circum}[\exists AB_2 \exists FLY \exists WING. A; AB_1] \\ & \equiv [\{(WILD) \wedge \text{wild-ch}\} \\ & = (AB_1 (WING WEAK))] \\ & \wedge \text{Rest} \end{aligned} \quad (3.21)$$

(3.3)において含意(\leq)を同値(=)にすることが、述語 AB_1 を circumscribe したことになる。同様にして、

$$\begin{aligned} & \text{Circum}[\exists FLY \exists WING. A; AB_2] \\ & \equiv [\{(HAS WING) \\ & \wedge (\text{THNOT } (AB_1 (WING WEAK))) \\ & \wedge \text{ab-bird}\} = (AB_2 (FLY))] \\ & \wedge [(WILD) \wedge \text{wild-ch} \\ & \leq (AB_1 (WING WEAK))] \wedge \text{Rest} \end{aligned} \quad (3.22)$$

得られた(3.21)(3.22)を用いて abnormal 述語 AB_1 , AB_2 を消去すると次のようになる。

$$\begin{aligned} & \text{Circum} [A; AB_1 > AB_2; FLY, WING] \\ & \equiv [(HAS WING) \wedge \text{bird} \\ & \wedge (\text{THNOT } \text{ab-bird}) \leq (FLY)] \end{aligned} \quad (3.23)$$

$$\begin{aligned} & \wedge [(HAS WING) \wedge (WILD) \\ & \wedge \text{wild-ch} \leq (FLY)] \end{aligned} \quad (3.24)$$

$$\wedge [(WILD) \wedge \text{wild-ch} \leq (FLY)] \quad (3.25)$$

$$\wedge [(WING WEAK) \wedge \text{ab-bird} \leq (\text{NOT}(FLY))] \quad (3.26)$$

$$\begin{aligned} & \wedge [(\text{THNOT } (WILD)) \\ & \wedge \text{ab-bird} \leq (WING WEAK)] \end{aligned} \quad (3.27)$$

$$\begin{aligned} & \wedge [\text{ab-bird} \wedge (\text{THNOT } \text{wild-ch}) \\ & \leq (WING WEAK)] \end{aligned} \quad (3.28)$$

$$\begin{aligned} & \wedge [(AB_1 (WING WEAK)) \\ & = \{(WILD) \wedge \text{wild-ch}\}] \\ & \wedge [(AB_2(FLY)) \end{aligned}$$

$$\begin{aligned} & = \{(HAS WING) \wedge (\text{THNOT}(WILD)) \\ & \wedge \text{ab-bird}\} \vee \{(HAS WING) \\ & \wedge \text{ab-bird} \wedge (\text{THNOT } \text{wild-ch})\}] \\ & \wedge \text{Rest} \end{aligned}$$

世界名を含めた clause において abnormal 述語を circumscribe する時には世界名を通常の述語と同様に扱うが、上記の結果をみると (THNOT 世界名) という subgoal が出現している。導入する際に規定した定義より世界名 subgoal を、それを持つ clause がその世界に存在する場合に真となる述語と考えると (THNOT 世界名) は示された世界にそれを body に含む clause が存在することが証明できなければ真と解釈できる。すなわち示された世界にその clause があってはならないことを示しているのである。さらに世界名で示された goal には概念世界間の継承を表現する clause が適用されるので (THNOT 世界名) は示された世界を含む、それよりも下位の概念世界で偽となる。よって、THNOT を伴う世界名 subgoal はそれを body に含む clause の示す知識の継承を示された世界で断ち切らなければならないことを表しているのである。以上の考察により circumscribe 後の世界名 subgoal の解釈は次のようになる。

通常の世界名……その clause が新たに assert されるべき世界を表す。

THNOT を伴う世界名……その clause の継承を打ち切るべき世界を表す。

Circumscription を行った後の clause (THNOT 世界名) が現れた時、それによって示された世界が、通常の goal として示された世界の低位に位置している場合には (THNOT 世界名) で示された世界でその clause の継承を打ち切らなければならない。具体的には図4のように、示された世界で上位概念からの属性継承ルートである is-a リンクを切断、階層を分割しその clause の存在する世界と並列な世界としてしまう。

なお、Circumscription を施す前の論理式(3.10)~(3.14)および Circumscription 後の論理式(3.23)~(3.28)を見るとわかるように、abnormal 述語を含んでいる(あるいは、含んでいた) clause の head には世界名が現れていない。これは、abnormal 述語+THNOT という goal および Circumscription による abnormal 述語消去操作の持つ非単調性は、直接には、世界に影響しないということを表している。つまり、世界の階層構造が変化するのは、世界の中に記述

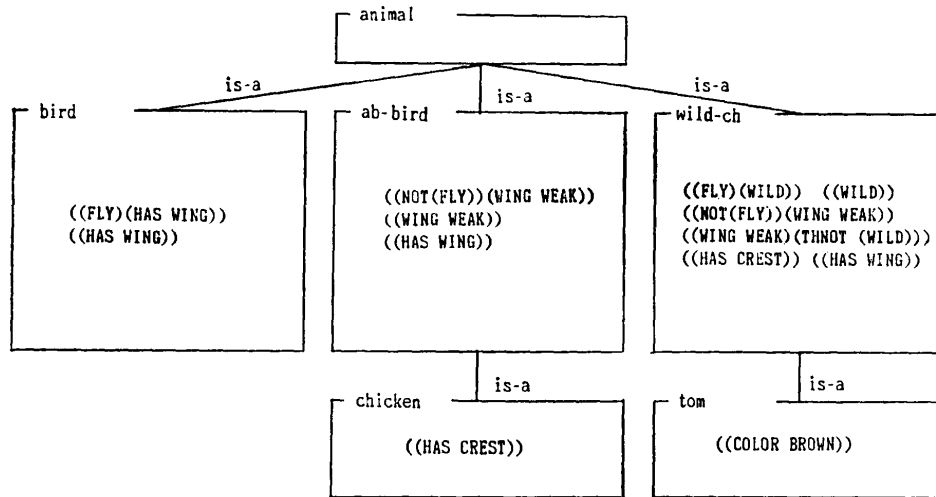


図 4 Circumscription 後の世界の構造
Fig. 4 The structure of the circumscribed knowledge base.

されている知識の持つ非単調性に起因する間接的なものである。また、世界自身が非単調性を持つわけではないので、一度、知識として導入された世界が消滅することはない。

4. 世界の分割の制御

clause 中に THNOT を伴って現れる世界が、その clause の継承を断ち切るべき世界を示しているという解釈を与えた。しかし THNOT を伴って現れる世界すべてについて知識の継承を断ち切るために示された世界で概念階層の分割を行うと場合によっては分割をしすぎてしまう恐れがある。人間の学習を考えた場合でも、概念階層の変更は、できるだけ少なくするように思われる。そこで概念階層の分割を最小限に抑える方法について考えてみる。例えば先ほどの (3.27), (3.28) では、

$$(THNOT (WILD)) \wedge ab\text{-}bird \leq (WING WEAK) \quad (3.27)$$

$$(THNOT wild\text{-}ch) \wedge ab\text{-}bird \leq (WING WEAK) \quad (3.28)$$

いずれも body に ab-bird を持っているので世界 ab-bird における述語 WING の定義を与えている。ところが (3.27) が世界 ab-bird から属性を継承する世界にも共通して成立するのに対して、(3.28) は世界 wild-ch を含むそれより下位の世界では成立しない。図 1 のような世界の階層では世界 wild-ch においては (3.27) の定義を採らざるをえないが、世界 ab-bird ではそのほかに世界間の構造に影響を与える (3.28)

も採用できる。(3.28) を含めると、世界 ab-bird での述語 FLY の定義は body の世界名を取り去った Uranus の program

$$((WING WEAK) (THNOT (WILD))) \quad (4.1)$$

$$((WING WEAK)) \quad (4.2)$$

となるが (4.1) は冗長なので結局は (4.2) のみとなる。このとき (3.28) の body にある (THNOT wild-ch) の解釈により図 4 の世界の階層構造となる。

しかし、次のように考えることもできる。Circumscription の結果として得られる述語 WING に関する abnormal 述語 AB₁ の定義は

$$(AB_1 (WING WEAK)) \equiv (WILD) \wedge wild\text{-}ch$$

となったが、述語 WILD が世界 wild-ch に assert されているので

$$(WILD) \text{ の成立} \equiv wild\text{-}ch \text{ より}$$

$$(AB_1 (WING WEAK)) \equiv (WILD) \quad (4.3)$$

$$\text{or } (AB_1 (WING WEAK)) \equiv wild\text{-}ch \quad (4.4)$$

のいずれかとなる。図 4 の場合は (4.4) を選択した結果と言える。ここで (4.3) を選択すると (3.13) から abnormal 述語 AB₁ を消去して得られる式は (3.27) [(4.1)] のみとなる。この場合は (3.28) の goal (THNOT wild-ch) の影響がないので世界の構造の分割を最小に抑えることができる。

これは述語 WING の abnormality を、その clause のなかに取り込んだ表現となっており、世界 wild-ch の存在による例外は世界 ab-bird 内で表現されている。学習という点から見ると図 4 は新しい知識構造を世界 wild-ch にもたらし、図 5 は世界 ab-bird に対し

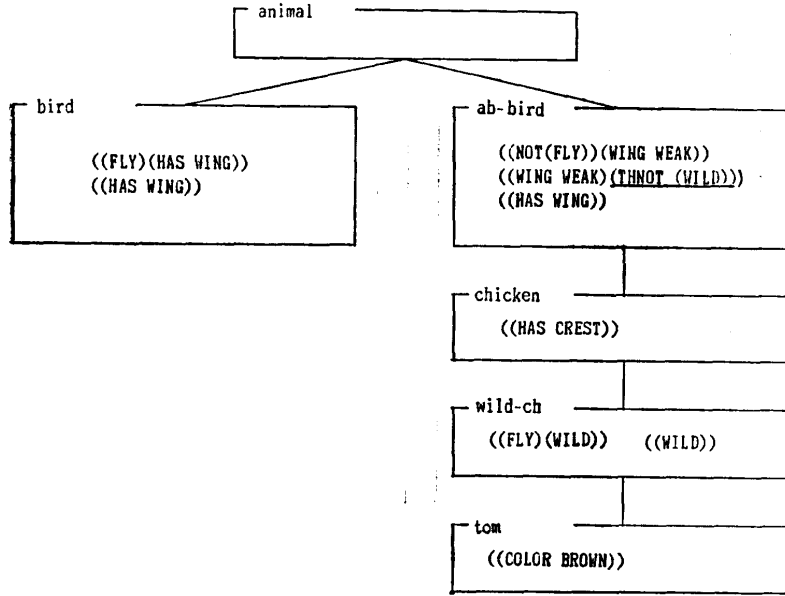


図5 世界の構造の分割を最小に抑える

Fig. 5 Minimization the number of an is-a hierarchy division.

ての新しい rule の学習となっている。

一般に, Circumscription の結果, 得られる abnormal 述語の定義が

$$(AB *) \equiv (P_1 *) \wedge \dots \wedge (P_n *) \wedge \text{world}$$

となる場合, 世界 world の直上の世界 world₁ において質問 $(P_1 *) \wedge \dots \wedge (P_n *)$ が fail するならば

$$(AB *) \equiv (P_1 *) \wedge \dots \wedge (P_n *) \quad (4.5)$$

が成立する。なぜならば, 質問 $(P_1 *) \wedge \dots \wedge (P_n *)$ の成功する世界 world₂ は世界 world と等しいかそれより下位の世界なので

質問 $(P_1 *) \wedge \dots \wedge (P_n *)$ の成功

$$\supset \text{world}_2 \supset \text{world}$$

が成り立つからである。abnormal 述語の定義が(4.5)のようになるときには世界の構造の分割を抑えることができる。

人間は図4, 図5のような二つの方法を適宜使い分けているので推論が速く, かつ効率的な記憶が行われていると思われる。これをシミュレートするためには, 分割を抑えるモードと, そうでないモードを設定により使い分けることが考えられる。

5. むすび

本稿では人間の「常識」による考え方の一側面を推論規則に定式化した Circumscription を多重世界環境下で行う手法について述べてきた。特に, 多重世界に記述された知識のみならず多重世界間の構造の変革

を行うことができることがわかった。また, 知識と, それらが定義されている概念の構造を同一の手法で取り扱うことができる。

一方, Circumscription を概念学習という点から眺めると「負の例(矛盾)からの学習」だけを行っている点に注意しなければならない。概念学習を行うシステムに利用する場合には「正の例からの学習」を積極的に行う, 類推・帰納学習の機能を別に用意すべきであろう。例えば, ホーン節の帰納学習アルゴリズムである MIS (Model Inference System)¹⁴⁾ では正の事実と負の事実の両方から変数を含む事実やルールを学習する方法を与えて

いる。

また, 第2章で述べたように, 現在のところ, 矛盾のチェック, abnormal 述語を付加する clause の選択などはユーザに委ねているのだが, これらを自動的に行う手法については検討中である。さらに, 本システムが扱う知識が大規模になった場合には, 計算量が膨大になると思われる。それは, Circumscription の過程で要する作業量が, 現在のアルゴリズムでは, 関与する clause 数の指数関数のオーダーとなるからだ。このため, 部分的な知識に対して少しずつ Circumscription を適用する手法の開発が望まれる。ここで, 問題となるのは Circumscription を知識に適用する順番に, その結果が依存するかということであるが, この点については別の機会に発表する予定である。

謝辞 日頃, 有益な議論を頂く, 人工知能勉強会 AIUEO 諸氏に感謝いたします。なお, この研究は, 文部省科学研究費『特定研究 多次元情報処理 60210006』並びに『一般研究 C 非単調論理に基づく学習に関する研究 61550258』により予算の援助を得ている。

参考文献

- 1) McCarthy, J.: Circumscription—A Form of Non-monotonic Reasoning, *Artif. Intell.*, Vol. 13, pp. 27-39 (1980).
- 2) McCarthy, J.: Applications of Circumscrip-

- tion to Formalizing Commonsense Knowledge, *AAAI Workshop on Non-Monotonic Reasoning*, pp. 295-324 (1984).
- 3) Lifschitz, V.: Some Results on Circumscription, *AAAI Workshop on Non-Monotonic Reasoning*, pp. 151-164 (1984).
- 4) Lifschitz, V.: Computing Circumscription, in *Proceedings Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, pp. 121-127 (1985).
- 5) 中島, 戸村, 諏訪: Prolog/KR から Uranus へ, 情報処理学会第 36 回知識工学と人工知能研究会資料, AI 36-2 (1984).
- 6) Shepherdson, J.C.: Negation as Failure, *the Journal of Logic Programming*, Vol. 1, No. 1, pp. 51-79 (1984).
- 7) 中川裕志: 論理型言語における explicit な否定知識, 情報処理学会第 27 回プログラミングシンポジウム, pp. 13-21 (1986).
- 8) 岸川, 中川: 多重世界機構を用いた非単調論理, 情報処理学会第 14 回ソフトウェア基礎論研究会資料, SF 14-1 (1985).
- 9) Doyle, J.: A Truth Maintenance System, *Artif. Intell.*, Vol. 12, pp. 231-272 (1979).
- 10) 森, 中川: 論理型言語における Circumscription, 情報処理学会第 46 回知識工学と人工知能研究会資料, AI 46-8 (1986).
- 11) 佐藤, 玉木: Prolog に於けるプログラム変換, *Proc. of The Logic Programming Conference '83*, 6.1, ICOT (1983).
- 12) McDermott, D. and Doyle, J.: Non-Monotonic Logic I, *Artif. Intell.*, Vol. 13, pp. 41-72 (1980).
- 13) Reiter, R.: A Logic for Default Reasoning, *Artif. Intell.*, Vol. 13, pp. 81-132 (1980).
- 14) Shapiro, E. Y.: Algorithmic Program Debugging, ACM Distinguished Dissertations, The MIT Press, Cambridge (1982).

(昭和 61 年 9 月 1 日受付)
(昭和 62 年 2 月 12 日採録)



森 辰則 (学生会員)

昭和 39 年生。昭和 61 年横浜国立大学工学部情報工学科卒業。現在同大学院工学研究科修士課程在学中。人工知能に関する研究に従事。



中川 裕志 (正会員)

昭和 28 年生。昭和 50 年、東京大学工学部電気工学科卒業。昭和 55 年、東京大学大学院博士課程修了。工学博士。昭和 55 年 4 月より横浜国立大学工学部勤務。現在、同大工学部電子情報工学科助教授。人工知能の研究に従事。AIUEO, 電子情報通信学会, 日本認知科学会, 日本ソフトウェア科学会各会員。