

例外を含む多重継承ネットワークにおける継承アルゴリズムについて
 On Inheritance Algorithm
 for Multiple Inheritance Networks with Exceptions

奥村学, 田中穂積 (東京工業大学工学部)

継承アルゴリズムに関する最近の研究は, 人工的に作り出した複雑な継承ネットワークを扱えるような強力な継承アルゴリズムを提案するものである. 本研究では, 記述した継承ネットワークを条件付きリンクや冗長なリンクを含まない等価な継承ネットワークに変換して推論を行う方法を提案する. この方法では, 継承アルゴリズムは比較的単純でよいことになる.

1. はじめに

概念とそれらの間の関係をネットワーク形式で表現する手法は知識表現ではよく用いられる. ここで, 概念はノード, それらの間の関係はリンクで表現される. 概念体系を分類して階層構造に記述する際には, IS-A関係が用いられる. この階層構造では, 下位概念はその上位概念のもつ性質を継承することが原則であり, そのためこの表現形式を継承ネットワークと呼ぶ. 継承ネットワークでは, 性質の継承により知識の記述量に関して経済性が得られる. しかし, 一般にこの性質継承には例外が存在する可能性がある. また, 継承ネットワークが単純な木構造ではなく, 下位概念が複数の上位概念をもつこともありうる. このような多重継承の場合には, 複数の上位概念間の背反性を考慮しなければならない. このようなことから, 例外を含む多重継承ネットワークでは, その継承アルゴリズムはかなり複雑になる.

例外を含む多重継承ネットワークに関する研究としては, [13], [1], [10]などがある. これらの最近の研究は,

- (1) 例外リンクや冗長なリンクを含む多重継承ネットワークにおいて正当な動作をする継承原理を議論する.
- (2) デフォルト論理[9]などの非単調論理を用いて例外を含む継承ネットワークおよびその継承アルゴリズムを定式化する.
- (3) (1)の継承原理に基づいた並列アルゴリズムを開発する.

ものである. その中でいくつかの正当な動作をする継承アルゴリズムが提案されている[13], [14]. しかし, これらの研究は, 人工的に作り出した複雑な継承ネットワークを扱うことができるような強力な継承アルゴリズムを提案するものであり, 提案された継承アルゴリズムは, 実際の知識からみると不要なほど強力であると考えられる.

これらの研究のように, 記述した知識をそのまま推論に用いるのではなく, 本研究では, 知識表現システムを, 知識の記述, 記述した知識のコンパイル, そしてそれを用いた推論の3つの段階として考える. そして, 記述した継承ネットワークを条件付きリンクや冗長なリンクを含まない等価な継承ネットワークに変換して推論を行う方法を提案する. この方法では, 実際に推論を行う継承アルゴリズムは比較的単純でよいことになる.

2. 継承ネットワークの定義

継承ネットワークは非循環(acyclic)とする. また, リンクの始点をsource, 終点をdestinationとよぶことにする. リンクとしては, IS-A, IS-NOT-A, exceptionリンクのみを考え, 性質名をラベルにもつリンクは説明を簡単にするため考えない(この性質名リンクを含むネットワークは, 性質名リンクをIS-Aリンクに, そしてそのdestinationノードを「性質値」から「その性質名に対してその性質値をとる概念」に変更することにより, 等価なネットワークに置き換え可能である).

IS-Aリンク $A \longrightarrow B$

実線矢印で表わす.

IS-NOT-Aリンク $A \dashrightarrow B$

横線付き実線矢印で表わす.

exceptionリンク $A \cdots \cdots B$

点線矢印で表わす. IS-A, IS-NOT-Aリンクと異なり, exceptionリンクは, ノードからリンクへのリンクであり, ノードがそのリンクに対する例外であることを表わす.

IS-A関係は推移的であり, このIS-A関係を通して性質継承は実行される. 単一継承ネットワークでは, 例外は, 下位概念が上位概念の性質と矛盾する性質をもつことに対応するが,

「矛盾する性質が存在しないかぎり, 下位概念は上位概念の性質を継承する」

という原則に基づいて継承アルゴリズムを実現すれば, 下位概念の性質が常に優先されるので, 一般に例外は意識する必要がなく, 従って陽に記述する必要もない. しかし, 矛盾する性質をもつわけではなく, 上位概念の性質の継承を単にブロックする記述を行いたいときや多重継承ネットワークでは, exceptionリンクを用いて例外を陽に記述しなければならないことがある. 例を図1.2に示す. ノードCからリンクB→Dへの点線のリンクがexceptionリンクであり, CがBからDへのIS-Aリンクに対する例外であることを表わす. 従って, 図2で, Aは, Cの性質を継承するため, BがDの性質を継承するのに対し, Dの性質を継承しない. 図1の継承ネットワークでは, exceptionリンクのsourceノードがdestinationリンクからみて下位にあるのに対し, 図2の継承ネットワークでは, exceptionリンクのsourceノードはdestinationリンクと上位/下位関係にない. このため, 図1の継承ネットワークでは, 上に述べた継承の原則をそのまま用いてexceptionリンクが扱えるのに対し, 図

2の継承ネットワークでは、exceptionリンク及びそのdestinationリンクであるIS-Aリンクに対し、(そのIS-Aリンクを用いて推論を行う際、exceptionリンクが成り立つかどうか調べるという)余分な推論を実行しなければならない。この推論の問題点及びその解決法は3.2で述べる。

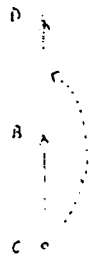


図1 下位ノードから上位リンクへのexceptionリンクを含むネットワーク

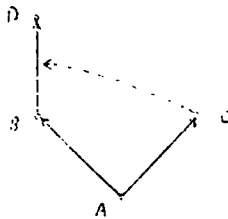


図2 下位ノードではないノードからのexceptionリンクを含むネットワーク

3. 継承ネットワークのコンパイル

3.1. 冗長なリンクの除去

最短パスをたどって継承を行う推論方式が、従来継承アルゴリズムとして用いられることが多かった。しかし、Touretzkyら[13][14]は、この最短パス・アルゴリズムを用いると、冗長なリンクを含むネットワークでは誤った結論を導く可能性があることを指摘し、この問題を解決する継承原理として、パスの長さではなく、継承の途中経路を考慮した推論距離順序(inferential distance ordering)という考え方を提案している。Touretzkyらは、最短パス・アルゴリズムで問題となるネットワークの例として図3に示すものを挙げている。最短パス・アルゴリズムでは、ClydeからElephantへの冗長なIS-Aリンクがあるため、Clydeに関して、Grayでない(Royal elephantがGrayでないことに基づく)、Grayである(ElephantがGrayであることに基づく)という矛盾した結論が導かれてしまう。

しかし、本来知識表現において冗長なリンクは何の役割も果たさないはずである。また、知識は整合的に保たれるべきであることを考えると、冗長なリンクは知識管理機構[5]により除去されるべきである。このような考え方に基づき、冗長なリンクを除去した継承ネットワークで推論を行う立場を主張する。

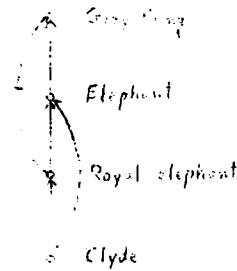


図3 冗長なリンクを含むネットワーク ([14]より)

まず、冗長なリンクを定義する。ノード間を直接結ぶIS-Aリンクを直接リンク、1つ以上のノードを経由してIS-Aリンクで結ばれている時には間接リンクで結ばれているということにする。ノード間に2つ以上のリンクがあるのは次の3つの場合である(図4参照)。

- (1) ノード間に2つ以上の直接リンクが存在する。
- (2) ノード間に直接リンクと間接リンクが存在する。
- (3) ノード間に2つ以上の間接リンクが存在する。

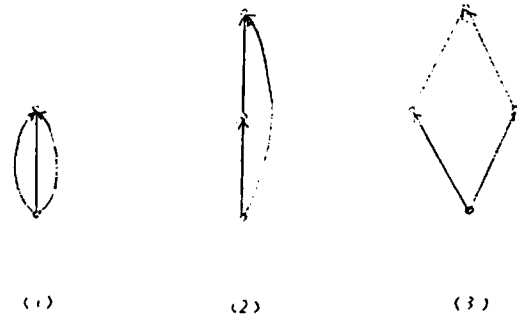


図4 ノード間に複数のリンクがある場合

(1)の場合、すべての直接リンクは等価であるから、1つを除いて残りのリンクはすべて冗長なリンクと判断し除去する。(2)の場合、直接リンクから得られる情報は、間接リンクを通して得られる情報の部分集合になっているので、直接リンクは冗長なリンクとみなし除去する。(3)の場合、それぞれの間接リンクから得られる情報の中にお互いに含まれない部分が存在するので、どのリンクも冗長なリンクとは考えず、多重継承として扱う。

この冗長なリンクの検出・除去は、ユーザが記述する知識の一貫性を動的に保持する知識管理機構によって実行される。我々は、冗長なリンクの除去機能を含むシソーラス作成支援ツールを逐次型推論マシンPSI上に作成している[8]。

3.2. exceptionリンクの展開

図1.2のようなexceptionリンクを含む継承ネットワークをそのまま用いて推論を行う方法が従来用いられている[11]。しかし、[11]で述べられているように、この方法では、図2のようにsourceノードとdestinationリンクの間に上位/下位関係のないexceptionリンク

を含む継承ネットワークでは、2で述べた継承の原則をそのまま用いることができず、本来ごく少数であるはずの例外に対して、継承の過程で例外であるかどうかを必ず調べなければならず、推論の効率を著しく低下させる。

そこで、記述された継承ネットワーク中の図2のタイプのexceptionリンクを図1のタイプのexceptionリンクに変換し、変換後の図1のタイプのexceptionリンクしか含まない等価な継承ネットワークを用いて推論を行う方法を提案する。図1のタイプのexceptionリンクしか含まない継承ネットワークでは、2で述べた継承の原則に基づいて容易に例外を扱うことができる。図2の継承ネットワークは図5の継承ネットワークに変換される。変換の原則は、

exceptionリンクのsourceノードを、exceptionリンクのsourceノード、destinationリンク両方の下位ノード中、最上位のノードに変更する

である。図2では、exceptionリンクはノードCからリンクB→Dへ向かっているのので、そのsourceノードを、Cから、C、B両方の下位ノード中最上位にあるAに変更する(図5参照)。図2の例から明らかなように、変換後のexceptionリンクのsourceノードである、変換前のexceptionリンクのsourceノードとdestinationリンク両方の下位ノード中最上位のノードは、そのexceptionリンクのdestinationリンクの例外となる最も一般的な(most generic)ノードであり、かつこのノードの性質を継承しないノードはすべて、例外とならないことに注意してほしい。すなわち、このノードの性質を継承するノードだけが例外となる。これは、例外となるノードをコンパイル時にcircumscribeし[6]、推論時の負担を軽くしていることになる。

また、exceptionリンクは、[7]でいう制約リンクのうち、負のリンクに相当しており、その観点から見ると、このexceptionリンクの展開は、リンクに付加されている制約を推論実行前に部分計算[3]していることに相当する。

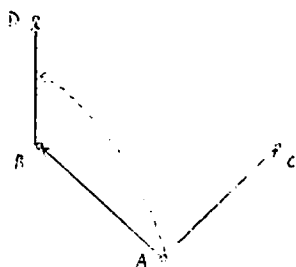


図5 図2のネットワークのコンパイル後のネットワーク

4. 継承アルゴリズム

3で述べた継承ネットワークのコンパイル後に実際に推論を行う継承アルゴリズムについて説明する。3.2で述べたように、単一継承ネットワークの場合には、ネットワークコンパイル後の継承アルゴリズムは、2で述べた原則

「矛盾する性質が存在しないかぎり、下位概念は上位概念の性質を継承する」

に基づいていけば十分である。問題となるのは、多重

継承ネットワークの場合である。1で述べたように、多重継承ネットワークの場合には、複数の上位概念間で得られる結論に矛盾が生じる場合がある。この矛盾として以下の2通りを考える。

(1) 背反な上位概念に到達する。ここで、概念の背反性とは次のようなものである。概念体系を分類する際には、必ずある見方[4]をとることにより下位概念に分類している。その1つの見方で分類した下位概念同士は互いに背反である。

(2) 上位概念同士は互いに背反ではないが、それらのもつ性質の中に矛盾するものが存在する。

このため、多重継承ネットワークを扱う継承アルゴリズムは以下ようになる。

- ① 多重継承するノードが現れたらその上位概念をパス名[12]として分岐する。
- ② それらの上位概念が互いに背反であるかどうか検査する。
- ③ それぞれのパスを上位方向にたどる。
- ④ 拡張(extension)を作成する。
 - a. ②で上位概念が背反な場合
得られる性質集合は、それぞれのパスにより得られる性質集合の選言(disjunction)をとったもの
 - b. 上位概念が無矛盾で、得られる性質にも矛盾するものが存在しない場合
得られる性質集合は、それぞれのパスにより得られる性質集合の和集合
 - c. 上位概念は無矛盾であるが、得られる性質に矛盾するものが含まれる場合
得られる性質集合は、無矛盾な性質集合と、矛盾する性質に関してその選言をとった集合の和集合

以下では、図6の継承ネットワークに関して、上で述べた継承アルゴリズムを用いて推論を行う過程を示す。

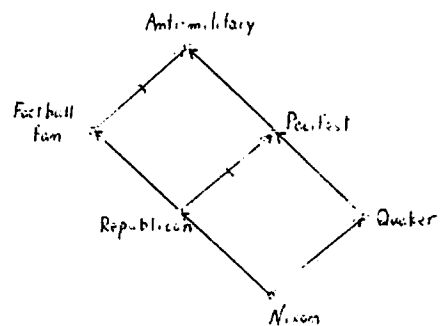


図6 曖昧なネットワーク ([14]より)

- (1) Nixonから継承ネットワークを上に向けてたどる。
- (2) QuakerとRepublicanは互いに背反ではないので、それぞれのパスを上をたどる。その結果、それぞれのパスに対して、次の性質集合が得られる。

Q	P/Q	A/P/Q
R	~P/R	F/R
		~A/F/R

ここで、'~'は否定を表わしており、各概念はそのイニシャルのアルファベットで表現している。また、パスの履歴は、'/'を用いて表現する。このパスの履歴は、命題の成立条件を表わしていると考えられ(例えば、P/Qは、P if Qを表わしている)、次の拡張の計算で用いられる。

- (3) 1. 互いに矛盾しないQ, Rを無矛盾な性質集合に加える.

無矛盾な性質集合 (Q, R)

それぞれのパスの性質集合中の条件からQ, Rを除去する. その結果, それぞれのパスに対する性質集合は次のようになる.

P A/P
 ~P F ~A/F

2. P, ~Pは互いに矛盾する性質が存在するので, 無矛盾な性質集合には付加できない. 次に, Fが, Q, Rと同様に, 無矛盾な性質集合に付加され, 性質集合中の条件から除去される. この時点で, それぞれのパスに対する性質集合は次のようである.

P A/P
 ~P ~A

3. ~Aも矛盾する性質が存在するので, 従ってこれ以上無矛盾な性質集合は拡張することができず, 結果として

無矛盾な性質集合 (Q, R, F),
 矛盾する性質集合 (P, A/P, ~P, ~A)

が得られる.

4. 矛盾する性質集合から無矛盾な最大部分集合の集合を計算する. その結果,

{(P, A), (P, ~A), (~P, ~A)}
 が得られる. そして, 最終的に
 $\{(Q, R, F) \cup \{(P, A), (P, ~A), (~P, ~A)\} =$
 $\{(Q, R, F, P, A), (Q, R, F, P, ~A),$
 $(Q, R, F, ~P, ~A)\}$

が性質集合として得られる. この集合の要素である3つの集合はそれぞれが1つの拡張を表わしている.

このように曖昧な継承ネットワークにおいて, 可能な複数の拡張をすべて得る点で, 信心的(credulous)[14]な立場をとることになる.

5. おわりに

従来のように, 記述した知識をそのまま推論に用いるのではなく, 知識表現体系を, 知識の記述, 記述した知識のコンパイル, そしてそれを用いた推論の3つの段階として考え, 記述した継承ネットワークを条件付きリンクや冗長なリンクを含まない等価な継承ネットワークに変換して推論を行う方法を提案した. この方法では, 実際に推論を行う継承アルゴリズムは, 単一継承ネットワークに関しては,

「矛盾する性質が存在しないかぎり, 下位概念は上位概念の性質を継承する」

という直観に合った原則に基づいていけば十分であり, 多重継承ネットワークに関しても, 4で述べたような, 複数の上位概念に対するパスから無矛盾な拡張集合を計算する手続きを付加するだけでよい.

今後の課題としては, 以下のようなものが挙げられる.

- (1) 3で述べた継承ネットワークのコンパイル・アルゴリズムの定式化, 妥当性の検討, 及びインプリメントを行う.
- (2) 4で述べた無矛盾な拡張集合の計算手続きの定式化を行う.
- (3) 4で述べた継承アルゴリズムを並列に実行するプロ

グラムを開発する. 現在, 並列論理型言語GHC[2]によるインプリメントを検討中である.

また,

- (4) 継承ネットワークのコンパイルは, ある1時点での継承ネットワークに対するものであるため, 時間変化などを含む動的な継承ネットワークには十分対処できない. このような動的な継承ネットワークの研究として[7]がある. これに対し, 4で述べたように, '/'を用いたパスの履歴は, 条件を表わしていると考えられるので, 本継承アルゴリズムも, 動的な継承ネットワークを扱えるように, 容易に拡張可能である.

参考文献

- [1] D.W.Etherington: Reasoning with Incomplete Information, Research Notes in Artificial Intelligence, Pitman, 1988.
- [2] 古川康一, 溝口文雄(編): 並列論理型言語GHCとその応用, 知識情報処理シリーズ6, 共立出版, 1987.
- [3] 二村良彦: "部分計算", 古川康一, 溝口文雄(編), プログラム変換, 知識情報処理シリーズ7, 共立出版, pp.63-80, 1987.
- [4] 井佐原均, 橋田浩一, 石崎俊, 内田ユリ子: "文脈理解のための概念記述法", 情報処理学会自然言語処理研究会, 64-7, pp.53-59, 1987.
- [5] 北上始: "知識獲得システム", 古川康一, 溝口文雄(編), 知識の学習メカニズム, 知識情報処理シリーズ2, 共立出版, pp.71-124, 1986.
- [6] J.McCarthy: "Applications of Circumscription to Formalizing Common-Sense Knowledge", Artif. Intell., Vol.28, No.1, pp.89-116, 1986.
- [7] 毛受哲, 伊藤英則, 森田幸伯: "制約付き性質継承に関する並列アルゴリズム", 人工知能学会第2回全国大会, 4-3, pp.145-148, 1988.
- [8] 望月泰行: シソーラス作成支援ツールに関する研究, 東京工業大学卒業論文, 1988.
- [9] R.Reiter: "A Logic for Default Reasoning", Artif. Intell., Vol.13, No.1,2, pp.81-132, 1980.
- [10] 坂間千秋, 奥村晃: "非単調並列継承ネットワーク", Proc. of the Logic Programming Conference'88, 4.1, pp.65-72, 1988.
- [11] 徳永健伸, 田中穂積: "知識継承における例外の扱いとその応用について", 情報処理学会第35回全国大会, 3P-2, pp.1837-1838, 1987.
- [12] 徳永健伸, 田中穂積: "視点を考慮した概念の同一化", 情報処理学会第37回全国大会, 3H-6, 1988(予定).
- [13] D.S.Touretzky: The Mathematics of Inheritance Systems, Research Notes in Artificial Intelligence, Pitman, 1986.
- [14] D.S.Touretzky, J.F.Horty and R.H.Thomason: "A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems", IJCAI '87, pp.476-482, 1987.