

ICOT Technical Memorandum: TM-0515他

TM-0515他

89-1 人工知能学会第2回全国大会

June, 1988

©1988. ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

- TM 0515 制約付き性質継承に関する並列アルゴリズム
毛受 哲, 伊藤英則, 森田幸伯
- TM 0523 プロセス指向データベースの実行管理
庄司 功, 伊藤英則
- TM 0534 CILプログラミング環境
天沼敏幸, 鈴木隆之(三菱電機), 奥西稔幸, 向井国昭
- TM 0535 論理型文法における制約解析
杉村領一
- TM 0540 協調型論理設計エキスパートシステムにおける仮説推論に
基づく再設計機構
角田多苗子, 丸山文宏, 松永裕介, 箕田依子, 川戸信明(富士通), T. KAKUDA
- TM 0541 MENDELにおける意味ネットを用いた部品結合
伊藤美香子, 内平直志(東芝 システム・ソフトウェア)

制約付き性質継承に関する並列アルゴリズム

A Parallel Algorithm for Inheritance Hierarchies with Constraints

毛受 哲、伊藤英則、森田幸伯

Satoshi Menju, Hidenori Itoh and Yukihiro Morita

(財)新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology

This paper introduces the concept of constraints in multiple inheritance hierarchies with exceptions. We add constraints to the links of inheritance networks, then consider links with satisfied constraints only, ignoring links with unsatisfied constraints. This method can increase the expressive power, making it possible to represent exclusive inheritance and preference of inheritance, for example. In this paper, constraints are restricted to self-constraints, that is, constraints from the same network.

This paper also describes a parallel algorithm for inheritance hierarchies with self-constraints. It terminates in $O(n)$ -time, where n is the length of the longest path of an inheritance network including self-constraints. The algorithm obtains one of the solutions produced by the credulous reasoners of Touretzky and Etherington. We also implement the algorithm in a parallel logic programming language, Guarded Horn Clauses (GHC).

1. はじめに

例外のある多重継承に関して今まで種々の研究がされてきたが[1~4]、それらはリンクがあれば無条件に推論できるものと考えていた。しかし、動的な情報や時制を表現しようとするには、それでは不十分である。そこで、本論文では制約の概念を継承階層ネットワークに導入する。

論理プログラミング言語の分野では、制約に関して、すでに新しいパラダイムを築きつつある。そして、制約を利用した協調問題解決のモデル化、及び、それによる並列計算機構の研究成果も期待されている。

ここでは、リンクごとに制約をつけ、指定された条件が成り立たない場合はリンク自体が無いものと考えことにする。これにより、時間等の制約をリンクにつけることができ、知識の表現力を強めることができる。これは、新しい知識

表現言語として発展する可能性を秘めている。

知識に関しては、表現力だけではなく、効率良く扱えることも重要である。そこで、本論文では継承階層ネットワーク及び制約付き継承階層ネットワークについて述べた後、後者に対する並列推論アルゴリズムを示し、並列論理型言語 GHC [5] による実現について述べる。これは、Touretzky らの定義する信心的な解の1つを求めるものの拡張である。

2. 継承階層ネットワーク

継承階層ネットワークは、個体と述語の集合と、その間のリンクで $\langle p, +q \rangle$ 、 $\langle p, -q \rangle$ の形をした順序対の集合 Γ から成る。ここで、 p は個体か述語、 q は述語を表わし、 $\langle p, +q \rangle$ は is-a、 $\langle p, -q \rangle$ は is-not-a を意味する。また、 p から q へのリンクがあれば、 p は q の子、 q は p の親であると言う。

グラフで表わすときは、個体を白丸(ノード)、述語を黒丸(ノード)、is-a リンクを矢印、is-not-a リンクを横線付矢印で示す。継承階層ネットワークには、同じノード間に存在するリンクは高々1つ、サイクルは存在しない、という制限がつくものとする。

さらに、 $\langle x_i, +x_{i+1} \rangle$ ($1 \leq i \leq n-1$) があるとき、 $\langle x_n, +y \rangle$ があるなら $\langle x_1, \dots, x_n, y \rangle$ を正のパス、 $\langle x_n, -y \rangle$ があるならそれを負のパスと呼ぶ。

継承階層ネットワークにおいて、p から q へのリンクがあるとき、それが is-a なら p は q (肯定的)、is-not-a なら p は q ではない(否定的)、と推論する。さらに、p は r であると推論できるとき、 $\langle r, +q \rangle$ があるなら p は q (肯定的)、 $\langle r, -q \rangle$ があるなら p は q ではない(否定的)、と推論する。正負の異なる2つのパスから肯定的な推論と否定的な推論ができるときは、Touretzky らの定義する inferential distance order [1] に従って優先するパスを決定する。ただし、彼等のオーダーは全順序ではないので、優先するパスを決定できない場合もある。

このような定義の下で問い合わせノードを1つ指定したとき、それから肯定を推論するノード、否定を推論するノード、どちらも推論しないノード、の集合の組合せを解と呼ぶ。しかし、パスの優先順位が決まらない場合は、解も一意に定まらない。このとき、可能な解すべての集合を求めることを信心的推論(credulous reasoning)と呼ぶ[1, 2]。筆者らは、[4]で信心的推論で求まる解の一部を求める並列アルゴリズムを提案し、その時間計算量がネットワークの最長パスの長さの線型オーダーで抑えられることを示した。

他に、優先順位が決まらない矛盾するパスがあるノードに対してはどちらも推論しない懐疑的推論(skeptical reasoning)もあり、このときは解は一意に定まる[2]。

3. 制約付き継承階層ネットワーク

制約付き継承階層ネットワークでは、リンクを次のように書く。

$$\langle p, @q, [C_1, \dots, C_n] \rangle$$

ここで、 $@ \in \{+, -\}$ であり、+ なら正のリンク、- なら負のリンクを示す。また、 C_i は制約を表わし、 $C_i = (+, s)$ なら状態 s が成り立つとき、 $C_i = (-, s)$ なら状態 s が成り立たないとき制約を満足する。そして、すべての制約 C_i が満足されていれば、p から q へのリンクはあるものと考え、1つでも満足しない制約があれば、そのリンクは無いものとする。ここでは、s が成り立つか否か不明のときも、 $C_i = (-, s)$ の制約は満足すると考えることにする。この s は、「朝である」のような時制や、その他様々な状態を表わし、それらが成り立つか否かは、あらかじめ与えられているか、問い合わせのとき一緒に与えられる。ただし、すべての制約に対し状態の成否を与えられるとは限らないので、制約が成り立つか否かに対しての推論も必要であり、また、それでも決定できない場合もあると考えられる。

本論文では、制約に使う状態をノードの状態、すなわち「問い合わせノードから、肯定的な推論ができる」に限定する。この場合は、制約のための情報を別に与える必要が無く、継承の推論はそのまま制約のための推論にもなる。この制約をグラフで書くときは、ノードからリンクへの点線リンクで表わす。区別のため、このリンクを制約リンク、継承を表わすリンクを継承リンクと呼ぶことにする。制約リンクの元(tail)のノードの状態が、その制約に対する状態である。制約も2種類あるので、+ の制約は点線矢印で、- の制約は横線付点線矢印で表わす。

この制約では、例えば次の2つの機能が実現できる。

- (1) 優先 a は、b でないときは c である (図1-a)
- (2) 排反 a は b 又は c だが、b, c 両方を推論することはできない (図1-b)

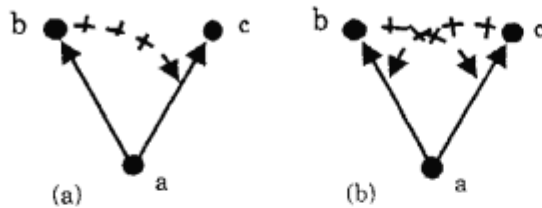


図1 制約付き継承階層の機能

また、Etherington & Reiter [3] のように、デフォルト論理風に継承階層を扱うこともできる (図2)。

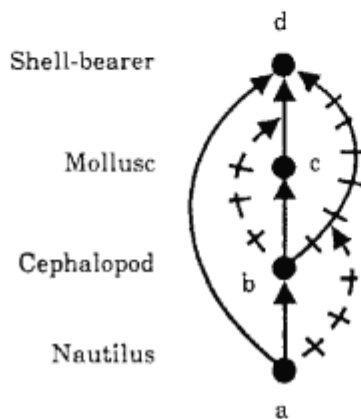


図2 デフォルト論理風継承階層

次節以降で、この制約付き性質継承に対する並列アルゴリズムを示す。

4. 並列アルゴリズム

制約付き性質継承ネットワークに対し、信心的な推論による解の1つを求める並列アルゴリズムを示す。これは、[4]において示したアルゴリズムの拡張であり、各ノードにマーカー M と距離 L のペア (M, L) を並列に伝播していく。マーカーは3種類使い、is-a を推論するノードに tm マーカー、is-not-a を推論するノードに fm マーカー、どちらとも推論しないノードには qm マーカーを置く。

アルゴリズムを示すために、まず、問い合わせノードから各ノードへの距離を定義する。

- 問い合わせノードの距離は、0である。
- 各ノードは、子供のノードの中に、マーカー tm を持ち、その間の継承リンクの持つ制約がすべて満足されているようなものがあれば、それらのノードが持つ距離の最大値に1を加えた距離を持つ。
- 上の2つの条件に合わない各ノードは、無限大の距離を持つ。

アルゴリズムでは、ノードは各々プロセスに対応し、入力リンクから送られてきたペアの中で、距離が最小であるペアのマーカーを自分のマーカーとし、出力リンクへリンクのタイプに合ったペアを送る。リンクの持つ制約がすべて満足されれば、そのペアはそのままリンクの先(head)のノードへ送られるが、1つでも満足されない制約があれば、代わりに (qm, ∞) が送られる。

実際には、次の様になる。ただし、ネットワークには $root$ という特別なノードと、 $root$ から子供を持たないすべてのノードへのリンクを付け加えるものとする。

Step1.

問い合わせノードにペア $(tm, 0)$ 、 $root$ ノードにペア (qm, ∞) を置く。

Step2.

(1) ペア (tm, L) を持つノードから出ている is-a リンクに対し、制約をすべて満たすなら、 $(tm, L+1)$ をリンクの先のノードへ送る。

(2) ペア (tm, L) を持つノードから出ている is-not-a リンクに対し、制約をすべて満たすなら、 $(fm, L+1)$ をリンクの先のノードへ送る。

(3) ペア (M, L) を持つノードから出ているリンクに対し、1つでも満足しない制約があるなら、 (qm, ∞) をリンクの先のノードへ送る。

(4) ペア (fm, L) か (qm, ∞) を持つノードから出ているリンクに対し、 (qm, ∞) をリンクの先のノードに送る。

Step3.

各ノードに対し、入って来るすべてのリンクからペアが来て、 tm または fm をマーカーとして持つペアがあるなら、それらの距離の最大値を L 、最小の距離を持つペアのマーカーを M とし、 $(M, L+1)$ を自分のペアとする。

入力ペアがすべて (qm, ∞) のときは、 (qm, ∞) を自分のペアとする。

Step4.

Step3 で新たにペアが置かれたノードがあれば、Step2 へ。無ければ、まだペアの置かれていないノードに (qm, ∞) を置いて終了。

アルゴリズム上の各操作は、該当するノード又はリンクが複数あれば、それらに対し同時に実行する。また、Step2 の(1)~(4)は並列に実行する。

ネットワークにサイクルが無ければ、このアルゴリズムの時間計算量は、制約リンクも含めたネットワークの最長パスの長さの線型オーダーで抑えられる。そして、得られた解が制約付き性質継承ネットワークに対する信心的推論の解の1つであることも、容易に示すことができる。

5. 並列論理型言語GHCでの実現

上の制約付き性質継承の並列アルゴリズムは、並列論理型言語GHC [5] によって容易に記述することができる。

図2の構造は図3のように書ける。

```
gen(Ms,N) :- true|
  node(a,Ms,Na,[],TMa,_),
  node(b,Ms,Nb,[(TMa,[])],Tmb,Fmb),
  node(c,Ms,Nc,[(Tmb,[])],Tmc,_),
  node(d,Ms,Nd,[(TMa,[]),
    (Fmb,[(Na,Na)],(Tmc,[(Nb,Nb)])],_),
  N=[Na,Nb,Nc,Nd].
```

図3 GHCによるプログラム例

genによってノードに対応したnodeというプロセスを立ち上げる。nodeは、第1引数が名前、第2引数が問い合わせの

メッセージ、第3引数がノードの持つマーカー、第4引数が制約も含めた入力リンクのリスト、第5引数がis-aリンクへ送るマーカーと距離、第6引数がis-not-aリンクへ送るマーカーと距離である。nodeは、継承や制約の情報が入って来るたびに評価を行ない、すべての入力が出て自分のマーカーを求めた後、マーカーと距離を送り出す。

6. おわりに

知識管理言語への基礎として、制約の概念を継承階層ネットワークに導入し、解を効率良く求める並列アルゴリズムを示した。これをGHCで自然に実現できることも示した。本論文での制約は、同じネットワークの内部から受けるものを対象としている。これにより、優先や排反、デフォルト論理風の表現ができることを示した。さらにより高度な制約の記述法及び処理法についても研究中である。例えば、ネットワークが複数あり、互いに制約し合えば、これは協調問題解決のモデルである。並列、制約、協調は相関があるので、これは自然なアプローチと言えよう。その他、他の分野の制約との関連も重要な課題である。

[参考文献]

- [1] Touretzky, D., "The mathematics of inheritance systems", Morgan Kaufmann, 1986.
- [2] Horty, J. F. et al., "A Skeptical Theory of Inheritance in Nonmonotonic Semantic Networks", Proc. of AAAI, 1987.
- [3] Etherington, D. W. and Reiter, R., "On Inheritance Hierarchies With Exceptions", Proc. of AAAI, 1983.
- [4] 毛受、森田、伊藤、例外のある性質継承に関する並列アルゴリズム、情報処理学会第36回全国大会論文集、1988.
- [5] 淵(監修)、古川、溝口(編集)、並列論理型言語GHCとその応用、共立出版、1987.