

# 並列 A I マシン Prodigy の応用 (2)

3N-5

— 並列意味ネットワーク向き外延的知識の表現方法 —

鈴岡 節      藤田 純一      小柳 滋  
(株) 東芝 総合研究所

## 1. はじめに

知識の表現方法として意味ネットワークが広く用いられており、これに伴い高速な意味ネットワーク上の検索が求められている。この要求に応じて現在までにいくつかの並列意味ネットワークマシンが提案されている[1][2]。これらは、小規模の処理装置を多数用いて、マーカ伝播のような単純な処理を高(超)並列に実行することにより、高速性を実現している。マーカ伝播によるモデルは単純でSIMD式制御下で効率良く動作する。しかし、マーカ伝播は意味ネットワーク上の比較的単純な検索を対象としており、複雑な検索は不可能か高速ではない。例えば、マーカの情報は少なくマーカの伝播の履歴を持たないので、検索パターンが閉路を持つ場合や、検索されたパターン(解)の間に交わりがある場合には、ユーザにとって自然な形で解が得られない。そこで我々は既存の並列意味ネットワークよりも高機能な意味ネットワーク操作のための外延的知識の表現方法について提案する。論旨を明確にするために、内包的知識については触れず、外延的知識についてのみ述べる。

## 2. 方針

意味ネットワークをより柔軟にかつ高速に検索するための外延的知識の表現方法の提案が目的である。具体的主眼点を述べる。

- ① 双方向性: AがBに対してrの関係にあるならば、BはAに対してrの逆の関係にある。このとき、rの逆の関係をユーザが新たに定義することなく、逆の関係を高速に検索したい。勿論pure prolog では可能であるが、一般に記述上自然な方向に比べて逆は10倍くらい遅い。
- ② 高並列性: AND/OR並列性を活用する。
- ③ 高機能性: マーカ伝播の問題は、情報がマーカだけであったためにAND分岐の情報が欠落していた点にあった。このために可能な解は出せても、解の分離ができなかった。そこで、AND分岐時の情報をメッセージに付加して伝播させる。これによりメッセージパッシング型の通信モデルを採用する。

"Prodigy" Parallel AI Machine Applications(2)

- Extensional Knowledge Representation for Parallel Semantic Networks -

Takashi Suzuoka, Sumikazu Fujita, Shigeru Oyanagi  
TOSHIBA R&D Center

## 3. 対象とする意味ネットワーク

ここでは、ネットワークを単にラベル付き有向グラフとして捉え、概念(ノード)や基本関係(リンク)の意味については規定しないことにする。

[定義] 概念  $\mathbb{C}$

概念はネットワークのノードに相当し、ユニークな名前を持つ。概念の全体集合を記号 $\mathbb{C}$ で表す。□

[定義] 基本関係  $\mathbb{R}_p$

基本関係はネットワークの有向リンクに相当し、概念間の関係を表す。基本関係の全体集合を記号 $\mathbb{R}_p$ で表す。□

## 4. 関係の定義

[定義] 関係( $\mathbb{R}$ )のsyntax

以下から合成されるもののみが関係である。関係の全体集合を記号 $\mathbb{R}$ で表す。

- $r \in \mathbb{R}_p \rightarrow r \in \mathbb{R}$
- $q, r \in \mathbb{R} \rightarrow q \ r \in \mathbb{R}$  (連接)
- $q, r \in \mathbb{R} \rightarrow q \cap r \in \mathbb{R}$  (積)
- $q, r \in \mathbb{R} \rightarrow q \cup r \in \mathbb{R}$  (和)
- $r \in \mathbb{R} \rightarrow r^n \in \mathbb{R}$  (定数回反復)
- $r \in \mathbb{R} \rightarrow r^* \in \mathbb{R}$  (不定回反復)
- $r \in \mathbb{R} \rightarrow r^+ \in \mathbb{R}$  (不定回反復)
- $r \in \mathbb{R} \rightarrow r^{-1} \in \mathbb{R}$  (逆関係) □

[定義] 関係( $\mathbb{R}$ )のsemantics

関係は概念の集合から概念の集合への写像である。

$$\mathbb{R} : \mathbb{C} \rightarrow \mathbb{C}$$

概念  $c \in \mathbb{C}$  を関係  $r \in \mathbb{R}$  に作用させるときには、 $r \{c\}$  と書くのが正しいが、簡単のため  $r \ c$  と略記する。

$p \in \mathbb{R}_p, c \in \mathbb{C}, r \in \mathbb{R}, C \subset \mathbb{C}$  のとき、

$p \ c = \{ \text{概念 } c \text{ からリンク } p \text{ を順方向に辿った先の概念} \}$

$$p^{-1} c = \{ x \mid x \in \mathbb{C}, c \in p \ x \}$$

$$r \ C = \bigcup_{c \in C} r \ c$$

$$q \ r \ c = \bigcup_{x \in r \ c} q \ x$$

$$(q \cap r) \ c = q \ c \cap r \ c$$

$$(q \cup r) \ c = q \ c \cup r \ c$$

$$r^n = \underbrace{r \ r \ \cdot \ \cdot \ \cdot \ r}_{n \text{ 個}} \quad (n >= 0)$$

$$r^{-n} = (r^{-1})^n$$

$$r^* = \bigcup_{n \geq 0} r^n$$

$$r^+ = \bigcup_{n \geq -1} r^n = r^* r$$

$r$  の逆関係を表す  $r^{-1}$  は  $r^{-1}$  に対し、以下の規則を適用して得られたものと等価である。

$$q, r \in \mathbb{R}$$

$$(r^{-1})^{-1} = r$$

$$(r \circ q)^{-1} = q^{-1} \circ r^{-1}$$

$$(r \cap q)^{-1} = r^{-1} \cap q^{-1}$$

$$(r \cup q)^{-1} = r^{-1} \cup q^{-1}$$

$$(r^n)^{-1} = (r^{-1})^n$$

$$(r^{-n})^{-1} = r^n \quad \square$$

ここに次の定理が成立する。これは概念  $x$  が概念  $y$  と関係  $r$  にあるならば、概念  $y$  は概念  $x$  と関係  $r^{-1}$  ( $r$  の逆関係) にあり、逆に、概念  $y$  が概念  $x$  と関係  $r^{-1}$  にあるならば、概念  $x$  は概念  $y$  と関係  $r$  にあるとを示している。

[定理] 逆関係

$$\forall r \in \mathbb{R},$$

$$\forall x \in \mathbb{C}, \forall y \in r^{-1} x, x \in r^{-1} y$$

$$\wedge \forall y \in \mathbb{C}, \forall x \in r^{-1} y, y \in r^{-1} x \quad \square$$

(例) 基本関係と既定義の関係とから、新たな関係を合成していくことができる。例えば、

$$r = p^2 a$$

$$p = (d \ c) \cap b$$

のように定義されているとき、概念  $x$  に対して、

$$y = r^{-1} x$$

を求めることは、図1のパターンとネットワーク上でマッチする  $y$  を見つけることである。

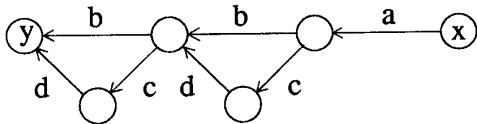


図1

また、

$$q = ((d \ c) \cap b)^2 a$$

のとき、 $q$  の逆関係は以下のように求まる。

$$q^{-1} = (((d \ c) \cap b)^2 a)^{-1}$$

$$= a^{-1} (((d \ c) \cap b)^{-1})^2$$

$$= a^{-1} ((c^{-1} \ d^{-1}) \cap b^{-1})^2$$

### 5. 実行モデル

ネットワーク上の検索とは、ある概念とある関係にある概念を調べることでありとする。具体的には関係に従ってリンクを辿ることである。ここで検索を行なう上での実行モデルの要点を列挙する。

ハードウェアモデル： マルチプロセッサ方式のハードウェアを仮定する。結合方式については規定しない。

割り付け： ネットワークの概念(ノード)を処理装置に割り付ける(概念数が処理装置数よりも多い場合は、一つの処理装置は複数個の概念を受け持つ。一つの概念は必ず一つの処理装置内に置かれる)。

処理装置の持つ情報： 自分の担当する概念に関する情報(概念と繋がっている基本関係の情報)、関与する可能性のある定義された関係の内容の情報を持つ。

並列性： AND 並列, OR並列を共にこなす。並列性は記述された関係の中にある  $\cap$  と  $\cup$  の明示的並列性と、一つの概念から複数の同じ基本関係を調べる場合と不定回反復時の暗示的並列性との両方を利用する。バックトラックはしない。

メッセージの持つ情報： 検索識別子, 現在の制御の位置, AND/ORで分岐した場合の分岐点情報, 定数回反復時の反復回数, 関係の呼び出し時の戻り位置の情報を持つ。

関係の展開： 新しい関係を他の関係を用いて定義できる(再帰は認める)ので、ある関係について検索する時にその内に含まれている関係を展開していく必要があるが、lazy evaluation で展開する。

### 6. おわりに

提案した外延的知識の記述法を用いることにより、与えられた関係から上記の定義の逆関係にある関係が機械的に生成できる。これにより、ユーザが関係毎に逆関係にある関係を定義する必要はなくなり、逆関係の定義の誤りも生じ得ない。逆関係を用いて柔軟な検索ができる。また、意味ネットワークの検索の中に含まれる並列性を素直に記述し引き出すことができる。更にメッセージパッシングに基づいて検索を行なうが、この際にAND分岐の情報をメッセージに付加することにより解の質を高めることができる。

現在、上記の形式の外延的知識表現を有する意味ネットワークを並列AIマシン実験機[3]上にC言語を用いてインプリメントし、その評価を行なっている。

### 参考文献

- [1] S.E.Fahlman, "NETL: A System for Representing and Using Real-World Knowledge", The MIT Press. (1979)
- [2] D.I.Moldovan, "An Associative Array Architecture Intended for Semantic Network Processing", Proc.acm'84 Annual Conf. The 5th Gen. Challenge, pp.212-221. (1984.Oct)
- [3] 小柳, 藤田, 中村, 鈴岡, "超並列AIマシンの構想" データフローワークショップ '87, pp.159-166. (1987.Oct.)