

意味記憶システム IX

— 意味ネットワークのIXMマシンへの割り付け —

5M-9

樋口哲也 古谷立美 半田剣一 楠本博之 国分明男  
(電子技術総合研究所)

1. はじめに

筆者らは意味ネットワークマシンIXMを開発中であるが、これまでのシミュレーション結果では汎用機に比べて3桁以上の速度の改善が見込まれている。本稿ではこのような改善の理由について述べると共に、IXMの性能に大きな影響を与える、ハードウェア上への意味ネットワークの割り付けについて考察する。

2. IXMの概要

【実行速度】 表1はMicro VAX II上での意味ネットワークの処理時間と、シミュレーションでのIXMのマシンサイクル数とを、処理対象の意味ネットワークの大きさをかえながら比較したものである。

表1. シミュレーションでの性能予測

	1600リンク	3300リンク	4500リンク
VAX	2分	9分	15分
IXM	2452 MC	2756 MC	3556 MC

MC: Machine Cycle (1MC:100ns)

VAXでの実行時間は、K-Prologで記述した意味ネットワーク言語IXLのインタプリタによってユーザが与えた質問を処理する際の実測時間である。意味ネットワークが大きくなるに従い探索すべき空間が飛躍的に増大するため、実行時間が著しく増えている。

これに対し、IXMでは意味ネットワークが大きくなってもその影響をさほど受けない。これは後述するように、IXMでは意味ネットワーク処理に連想メモリを効果的に利用していること、またマーカ伝搬の並列性を高める機構を導入しているためである。

表1のマシンサイクルは、現在100 nsに収まるように設計を進めており、汎用機に比べて相当の改善が見込まれる。意味ネットワークが大きくなっても実行時間が爆発的にはふえない利点があるため、特に大規模知識ベースに対してIXMの適用は効果的である。

【IXMの構成】 IXMは2種類の連想メモリを内蔵するPE群とそれらを接続するネットワークから成る(図1)。ネットワークの各節点にはネットワークプロセッサが配置される。

IXMの各PEには、意味ネットワークを格納し、かつそれに連想・集合演算を施す連想メモリと、IXLインタプリタを格納する連想メモリがある。後者は命令レベルの

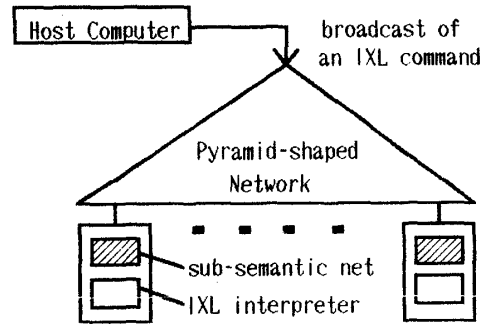


図1. IXMの全体構成

並列性を生かす目的がある。

3. IXMの特徴

【意味ネットワークの基本処理】 意味ネットワークの基本処理には連想演算、集合演算、マーカ伝搬の3つがあり、意味ネットワークの各ノードにマーカと呼ぶ1ビットのフラグを設けることで均一な形で取り扱える。

たとえば図2の2つの集合A, Bの共通要素を求める場合、まず集合Aのルートノードa0を連想演算によって見つけ、a0のマーカ1番をセットする。次にa0からisaリンクを次々に放射状にたどり、集合Aの各要素のマーカ1番をセットする。このようなマーカのセット操作をマーカ伝搬と呼ぶが、ここには並列性が多く存在している(たとえばもし集合Aが百万の要素を持っていたも2進木構造ならば10ステップでマーカ伝搬が終了する)。同様に集合Bの各要素のマーカ2番をマーカ伝搬によってセットし、最後にマーカ1番と2番のANDをとる集合演算を全ノードにブロードキャストすれば答えが求まる。

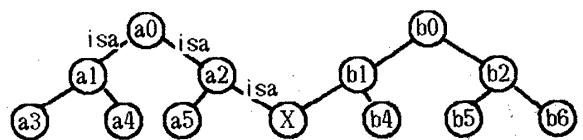


図2. 意味ネットワーク

【IXMでの基本処理の実現法】 IXMでは意味ネットワークをいくつかのサブ意味ネットワークに分割し、連想メモリに格納する。図3は図2の意味ネットワークを分割、格納した例である。

連想演算は、連想メモリの検索レジスタの属性(ID)フィールドに探すべき属性をセットして検索し、その後ヒットした語に対して特定のマーカ番号を書き込みに行けば実現できる。

集合演算も、たとえば図2の場合なら、マーカ1番と

Semantic memory system IX: the allocation of semantic networks on IXM machine  
Tetsuya Higuchi, Tatsumi Furuya, Ken-ichi Handa, Hiroyuki Kusumoto, Akio Kokubu  
Electrotechnical Laboratory

identifier	destination	link	marker bits		
			1	2	...
a 0	a 2	risa			..
a 0	a 1	risa			..
a 1	a 0	isa			..
..	..	..			..

図3. 意味ネットワークの連想メモリへの割り付け

2番だけがセットされているパターンを連想メモリの検索レジスタに入れて検索すれば集合の共通要素が一度に求められる。

これらの連想・集合演算は、意味ネットワークの各1リンクが格納されている連想メモリの各ワードごとに一斉に並列に行われるため、IXM内では連想メモリの全ワード数分の並列性が存在することになる。従って連想・集合演算についてはIXM内のPE台数には無関係であり、通常のマルチプロセッサのようにPE台数分の並列性の増加を議論することは意味がない。

しかしマーカ伝搬については、IXM内のPE台数、およびネットワークPE台数が多いほどその並列性が生かしやすい。但し扱う意味ネットワークの性質によって並列性が著しく異なる。また同じ意味ネットワークでもその意味ネットワークをどのような方法で分割し、さらにそれらをPE間にどのような分布で割りつけるかによってIXMの実行時間が大きく変化する。

このため、意味ネットワークの割り付けの問題はアーキテクチャの設計と同程度に重要である。

たとえば図4(a)の意味ネットワークのノードA0からA1~Anにマーカを伝搬したいとき、論理的には1ステップで伝搬が済むはずであるが、もしこの意味ネットワークが一台のPEに割りつけられていたのでは逐次処理になるのでマーカ伝搬にnステップを要してしまう。

このためファンアウトの大きいノードは図4(b)のようによりファンアウトの小さいノードに分割し、別々のPEに配置したほうがマーカ伝搬の並列性を生かせる。IXMではこれを等価ノード (node of equivalence) と呼ぶ。

但し等価ノードは分割されてはいとも同一の論理的属性を持つので、たとえばある別のノードから等価ノードに属する一つのノードに信号が到着したら、その信号は複製されて、その等価ノードに属する残りのノード群にも伝達されねばならない。この等価ノードの実現のためにネットワークPE内にも連想メモリを用いている。

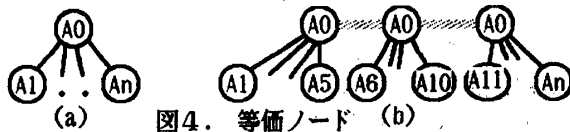


図4. 等価ノード

4. 意味ネットワークの割り付け

意味ネットワークを分割してPEに割り付けるアロケータのアルゴリズムによって、IXM内のデータ交信量、交

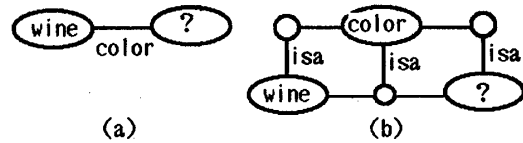


図5. 検索パターン例

信パターン、並列性が大きく変化する。そのような影響を与えるアロケータのアルゴリズムの基本要素としては次の二つがこれまでに明らかになっている。

- (1)ファンアウトの大きいノードをどう分割・割り付けるか?
- (2)パターンマッチをどの程度局所的に行えるか?

(2)のケースはたとえばワインの色を知るために図5(a)の意味ネットワークのパターンマッチングが要求されたときに対応する。図5(a)の意味ネットワークは意味ネットワーク言語IXLでは図5(b)の形に展開されており、このパターンが一台のPE内だけに含まれているときが一番パターンマッチの時間が短い。もしこのパターンがPE間にばらばらに分散していたのでは、ネットワークのデータ交信量やPEの負荷が増大し、実行時間が増える。

表2は3種類のアロケータ (A1,A2,A3)を2種類の意味ネットワーク (SN1,SN2)に適用してIXM上に割り付け、質問を与えたときの応答時間である。

表2. アロケータによる実行時間の違い

	A1	A2	A3
SN1	201	105	152
SN2	105	344	918

A1は意味ネットワークの性質を何も考慮しない単純なアロケータ、A2は等価ノードを生かすアロケータ、A3はA2の逆でファンアウトの大きいノードをなるべく一台のPEに割り付けるものである。SN1は4500リンクのワイン知識ベース、SN2はファンアウトの大きいノードをプログラムで発生した6300リンクのデータである。

表2でA2とA3の比較をすることによって等価ノードの効果を見ることができる。しかしSN2の場合にはA2はA1の単純なアロケータよりも遅くなっている。これは、SN2の場合、A1では上記(2)のように探索すべきパターンが局所的に割りつけられるのに対し、A2では等価ノードの割り付けを優先したためにパターンの局所的分布形態を破壊したためである。

このようにアロケータは適用する意味ネットワークの性質によって適、不適がある。しかしアロケータによっては性能が10倍近く変化するため、きわめて重要な研究課題であり、現在種々のアロケータのアルゴリズムの研究を進めている。

末筆ながら電総研柏木寛電子計算機部長に感謝する。

[参考文献]

- (1) 楠本、他: "意味記憶システムIX - 要素プロセッサ間結合方式のシミュレーション"、本大会。