



# 並列 AI マシン

古谷立美

電子情報通信学会誌 Vol. 71 No. 2 pp. 165-171 1988 年 2 月

古谷立美：電子技術総合研究所電子計算機部  
Parallel AI Machine. By Tatsumi FURUYA, Nonmember  
(Computer Systems Division, Electrotechnical  
Laboratory, Tsukuba-shi).

## 1. はじめに

人間の知的活動を計算機に代行させるものが人工知能 (AI) である。診断等で人間の判断を助けるエキスパートシステムや、自然言語処理等は良く知られた例である。人工知能の処理では、計算機内に蓄えた大量のデータと知識をうまく使って高速に正しい解を導くことが要求される。このような要求を満たすためになされる二つの工夫は

(1) データや知識を計算機で処理しやすいように上手に整理する。

(2) AI 処理に適した高速計算機を作る。  
である。

前者は知識表現とよばれる分野で、さまざまな知識表現方法が提案されている。後者は AI 専用の計算機を作る試みである。本解説のテーマである並列 AI マシンとは、AI 専用計算機を複数のプロセッシングエレメントを用いて実現するものである。

本稿ではまず AI の知識表現と処理の方法を紹介し、次にそれらをどう並列マシンで実現するかを解説する。

## 2. AI 処理のしくみ

図 1 は AI の応用がどのように実行されるかを示している<sup>1)</sup>。AI の応用は知識表現という形で整理され、これが Prolog や LISP というような AI 言語で記述され計算機で処理される。もちろん応用を直接 AI 言語で書き下し実行することもあるが、大規模な応用では一度知識表現の形に整理するのが一般的である。AI マシンはこれらのどこかに焦点を当てて、それをハードウェア化することにより高速処理を実現するものである。例えば、知識表現を

直接ハードウェア化するものは知識表現指向の AI マシンとよばれ、AI 言語を高速処理するように設計されたものは AI 言語指向マシンとよばれる。現在世の中で AI マシンというと LISP を高速化する LISP マシンである場合が多い。AI 言語指向マシンとしてはその他に Prolog マシン、知識表現指向マシンとしてはセマンティックネットワークマシンやプロダクションシステムマシン等が提案されている。次節では、これらのマシンを紹介するが、ここでは AI になじみのない方のために、知識表現とはどんなものか雰囲気を紹介する。

知識にはさまざまな種類があり、それぞれ特徴があるので目的ごとにいろいろな表現方法が開発されている。次の二つはその代表的なものである。

### 2.1 セマンティックネットワーク

知識を“概念”と“概念間の関係”で表現するものである。図 2 は、動物のセマンティックネットワークであり、概念がノードで、概念間の関係がリンクで表されている。セマンティックネットワークの特徴は上位概念の属性が下位概念に継承されることである。図 2 の例では動物の“餌を食べる”という性質がハチ公

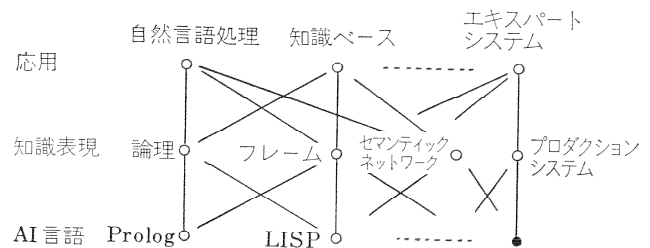


図 1 AI の処理過程 AI の応用は、個々の応用に適した知識表現を用いて整理され記述される。知識表現は AI 言語で書き下され、計算機で処理される。

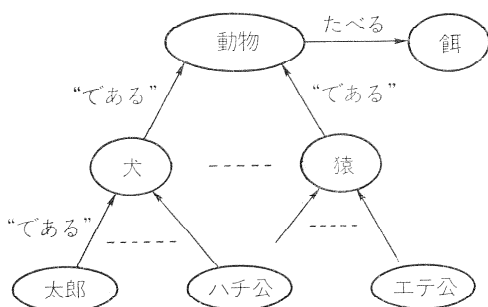


図2 セマンティックネットワークによる知識表現  
ノード(概念)とリンク(概念間の関係)で知識を表現する。上位概念の属性(例えば“餌を食べる”)が下位概念に継承される。

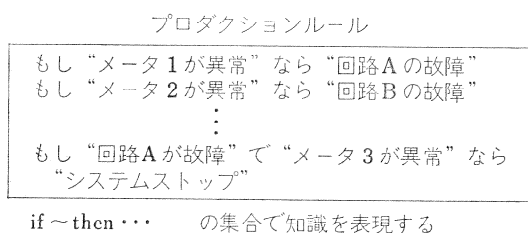


図3 プロダクションシステムの例

にも伝わる。図2に対して“ハチ公は餌を食べるか?”と質問すると、ハチ公には“食べる”と書いてないが、“である”リンクをたどって“ハチ公は餌を食べる”と答えてくれる。

### 2.2 プロダクションシステム

セマンティックネットワークは、“何々は何々である”と明言できる事柄を表すのに都合がよいが、診断などの知識は例えば、“のどが痛くて咳ができれば風邪の症状”、“風邪の症状を示しかつ高熱なら肺炎”というように“もし~ならば...”というルールの集合として表した方が簡単に表せる。図3は回路の故障に関する知識例である。このような表し方はプロダクションシステムとよばれ、現在多くのエキスパートシステムで用いられている。

## 3. 並列 AI マシン

先に述べたように並列 AI マシンとは、AI マシン

を複数のプロセッシングエレメント (PE) で実現するものである。表1はさまざまな AI マシンを示しているが、ここで示したどの AI マシンも複数の PE で実現することは可能であり、いろいろな並列 AI マシンが提案されている。

従来 AI プログラムには LISP が広く用いられてきており、最近では Prolog もよく用いられる。そのためこれらの AI 言語を用いた AI プログラムの蓄積は大きい。LISP マシンや Prolog マシンはそれぞれ LISP と Prolog で書かれたプログラムを高速に実行するように設計されたマシンである。これらのマシンを複数の PE で実現するアプローチは、今までに作られてきた多くの AI プログラムがそのまま並列処理という手段で高速実行され、ユーザへの負担はない。現在各所でこの種の並列実行の試みがなされており、そこではデータフロー、リダクションなどという方法が用いられている<sup>(2),(3)</sup>。これらについては紙面の関係上ここで詳しく述べることはできないが、簡単にいうと、一つの式のうち並列に実行できる箇所を捜して並列実行したり、あらかじめ実行できる部分は空いている計算機が先に実行してしまっておくというようなものであると考えれば良い。但し、これらの AI 言語はもともと1台の計算機での実行を前提としているため、プログラムに内在する並列性が低いことなどから苦勞のわりには効果が少ない。

AI 言語としては、その他に並列処理を前提にした並列 AI 言語も提案され、そのためのマシンの提案もある<sup>(4)</sup>。このような場合にはユーザが並列性を陽に記述するため並列性は高くなるが、プログラムは新たに書かねばならない。

知識表現指向の AI マシンは、知識表現の図式を直接ハードウェア化するもので高い並列性が期待できる。この種のハードウェア化は近年 VLSI の進歩により実現可能になり注目を集めている。以下に現在提案されているセマンティックネットワークマシンと、プロダクションシステムマシンについて、その実行メカニズムとアーキテクチャを紹介する。

表1 並列 AI マシン例

		AI 言語/知識表現	AI マシン	並列 AI マシン例
AI 言語	シリアル AI 言語	LISP Prolog	LISP マシン Prolog マシン	DFM, EM3 PIE
	並列 AI 言語	並列 Prolog	並列 Prolog マシン	PIM
知識表現		セマンティックネットワーク プロダクションシステム	セマンティックネットワークマシン プロダクションシステムマシン	NETL, Connection Machine, IX DADO, NONVON

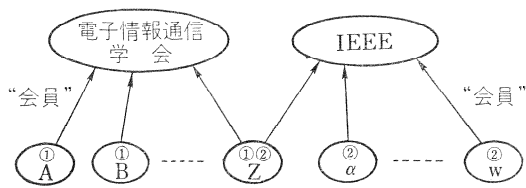


図4 セマンティックネットワークの処理 “電子情報通信学会”と“IEEE”の両方の学会に入っている人を探すには、“電子情報通信学会”から会員にマーカ①を送り(マーカ伝搬)、“IEEE”からマーカ②を送って、最後にマーカ①と②を持っているものを探す(集合演算)。

3.1 セマンティックネットワークマシン

セマンティックネットワークマシンはセマンティックネットワークの基本処理を効率良く実現するように設計されている。本節ではセマンティックネットワーク処理の基本を簡単な例で紹介し、次にそれを効率良く実現するマシンを示す。図4はセマンティックネットワークの例である。このようなネットワークに対して出される典型的な質問は、“電子情報通信学会とIEEEの両方に属する人はだれか?”というものである。この答を求めるために次の三つの基本操作が使われる。まず第1は検索の拠点となる“電子情報通信学会”、“IEEE”二つのノードを捜すことで、これは“連想”とよばれる。第2は“電子情報通信学会”、“IEEE”から下位のノードへリンクに沿ってそれぞれ異なるマーカを流すことで、これは“マーカ伝搬”とよばれる。図4の例では“電子情報通信学会”からマーカ1が、“IEEE”からマーカ2が流される。最後はマーカ1とマーカ2を持つノードを捜すことでこれは二つのマーカのANDをとることから“集合演算”とよばれる。セマンティックネットワークマシンは以上の3基本操作を高速に実行するように設計されている。現在次のようなセマンティックネットワークマシンが提案されている。

(i) NETL<sup>(5)</sup>

NETLは、カーネギメロン大学のS.E. Fahlmanによって提案された最初のセマンティックネットワークマシンである。NETLは多数のシンプルなPEをスイッチングネットワークで結合した形をしている。セマンティックネットワークのノードとリンクは、それぞれ一つのPEに割り当てられる。図5は図2のセマンティックネットワークの一部をNETLにのせた様子を示している。各PEは図5(a)のような構成で、マーカをセットするビットやコマンドインタプリタなどを持ち上記の“連想”と“集

合演算”を実行できる。セマンティックネットワークのリンクはスイッチングネットワークでハードウェア的にセットされる。マーカはセマンティックネットワークのリンクと一対一に対応付けられたライン上を伝搬する。NETLの特徴の一つは、セマンティックネットワークのノードとリンクを直接ハードウェアで実装するネットワークを用意してあることである。このネットワークはハッシュネットとよばれ図5の構造を持ち、図5(b)のセクタセル内のスイッチをセットすることによりノードとリンク(例えば“犬”と“である”)が直接ハードウェア的に結びつくことになる<sup>(6)</sup>。このようなマシンではマーカを送るとき行く先を指定するアドレス情報は不要となる。NETLは提案にとどまり、実現されていない。

(ii) コネクションマシン<sup>(7),(8)</sup>

コネクションマシンはMITのW.D. Hillisによって提案され、現在はThinking Machines社で製造販売されている。このマシンは、16個のPEをメッシュ結合とクロスバ結合でつないだものを1チップに納め、更にこのチップ4,096個をキューブ結合し、計65,536PEを実現している(図6(a))。PEの構成は、図6(b)で1ビットALUと4Kビットのメ

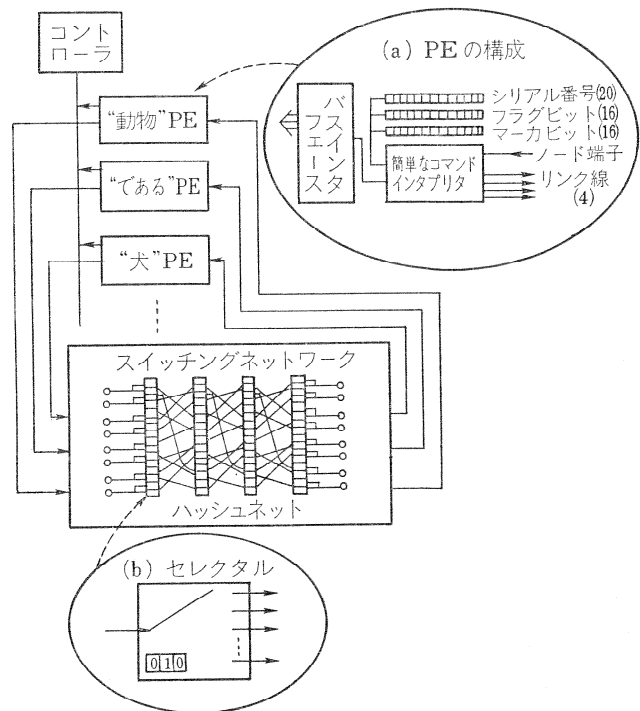
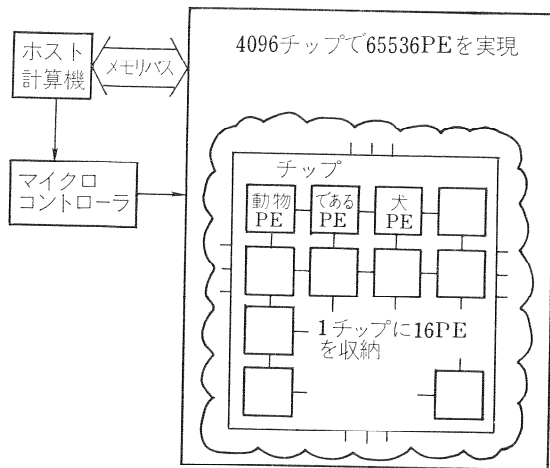
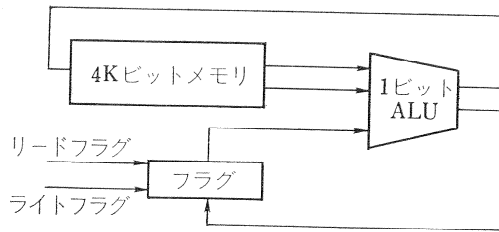


図5 NETLの構成 セマンティックネットワークのノードとリンクは一つずつPEに割り当てられる。PEはハッシュネットとよばれるネットワークで接続される。



(a) コネクションマシンの構成



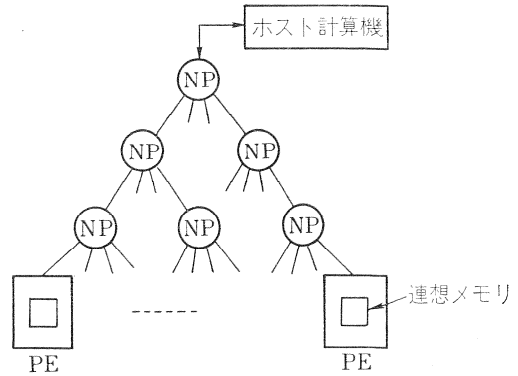
(b) PE の構成

図 6 コネクションマシン 16 PE を収納したチップを 4,096 個結合し 65,536 PE の計算機を実現。各 PE は 1 ビット ALU と 4K ビットのメモリを持つ。

メモリをもつ。このマシンはホスト計算機の制御下で SIMD で動作する。コネクションマシンへのセマンティックネットワークのマップは、セマンティックネットワークの各ノードが 1 PE に割り当てられる。コネクションマシン上でのマーカ伝搬は、ノード間が NETL のようにハードウェア的に一对一に設定されていないため、行き先をもったメッセージとして送られる。このマシンはもともとセマンティックネットワークマシンとして提案されたが、現在は画像処理やシミュレーションなどへの応用が強調されている。

(iii) IX (イックス)<sup>(9)</sup>

電子技術総合研究所で開発中のセマンティックネットワークマシン IX は上記二つと大分異なるアーキテクチャを採用している。各 PE は連想メモリを持ち、セマンティックネットワークは連想メモリに格納される(図7)。連想メモリとは、内容アドレスメモリともよばれ、指定される内容が入っている番地を一命令で捜せるメモリである。これによってセマンティックネットワークの基本操作である連想と集合演算は並列に実行できる。マーカはネットワークを通して伝搬さ



(a) IX マシンの構成

ソースノードフィールド	デスティネーションノードフィールド	リンクフィールド	マーカフィールド				
			0	1	2	--	--
動物	犬	“である”					
犬	ハチ公	“である”					
動物	猿	“である”					

(b) 連想メモリの内容

図 7 IX システム 各 PE は連想メモリを持ち、セマンティックネットワークは連想メモリに格納される。

れるが、このネットワークには特殊な工夫があり、一つのノードから多数のノードへマーカを高速に流すことができるようになっている。

3.2 プロダクションシステムマシン

3.1 にプロダクションシステムの雰囲気を示したが、もう少し正式に書くとプロダクションシステムは図8のように (a) プロダクションルール、(b) ワーキングメモリ、(c) インタプリタという三つの構成要素

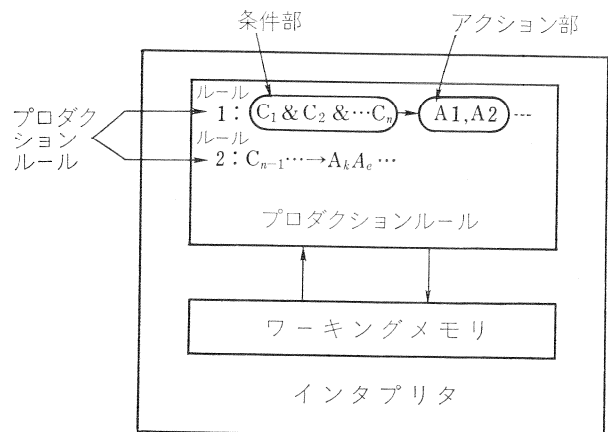


図 8 プロダクションシステム プロダクションシステムはプロダクションルール、ワーキングメモリ、インタプリタからなる。ルールは左辺(条件部)と右辺(アクション部)よりなる。条件部が真のルールの一つが選ばれ、その右辺が実行される。

からできている。図3の“もし～ならば…”という規則はプロダクションルールといわれる部分である。プロダクションルールは図8のルール1, ルール2, …のように左辺(条件部)と右辺(アクション部)よりなる。条件部は“もし～ならば”の～にあたる部分で、この部分が“真”のルールが実行可能なルールとなる。一般に条件部の真偽はワーキングメモリ内のデータによって決定される。プロダクションシステムの実行はまず各プロダクションルールの条件部を調べ、各プロダクションルールに対して“真”, “偽”を決定する。このフェーズはマッチステップとよばれる。実行可能なルールが決まるとインタプリタはその中から実行すべきプロダクションルールを決定する。これは選択ステップとよばれる。最後はプロダクションルールの実行で、選択されたプロダクションルールの右辺、すなわちアクション部が実行される。これはアクトステップとよばれ、一般にはこの実行によりワーキングメモリの内容が書き換えられる。ワーキングメモリの内容が書き換えられると実行可能なプロダクションルールが前と変ることになる。プロダクションシステムは以上に述べた(1) マッチ, (2) 選択, (3) アクトの三つのステップを繰り返すことにより最終的な解にたどりつく。次にプロダクションシステムマシンの例を示す。

(i) DADO<sup>(10)</sup>

DADO はコロンビア大学で開発されたプロダクションシステムマシンである。マシンは図9のように木構造をしており、PM レベルの各 PE に一つずつプロダクションルールが割り当てられる。あるルールが割り当てられている PE より下位木の PE には、そのルールに関係のあるデータ(ワーキングメモリの一部)が格納される。プロダクションシステムの実行ではまず、PM レベルの全 PE が並列にプロダクションルールの左辺のチェックを行い、条件部が真の PE は木の上位へその旨を知らせる。上位木では優先度のチェックが行われ、最上位へ達したルールが、選択されたルールとなる。最後に、選択されたルールのアクション部が全 PE にブロードキャストされ全 PE で並列にアクション部が実行されデータ(ワーキングメモリ)の書き換えが行われる。コロンビア大学ではこのほかに NON-VON とよばれる木構造計算機も提案されプロダクションシステムの実行が議論されている<sup>(11)</sup>。

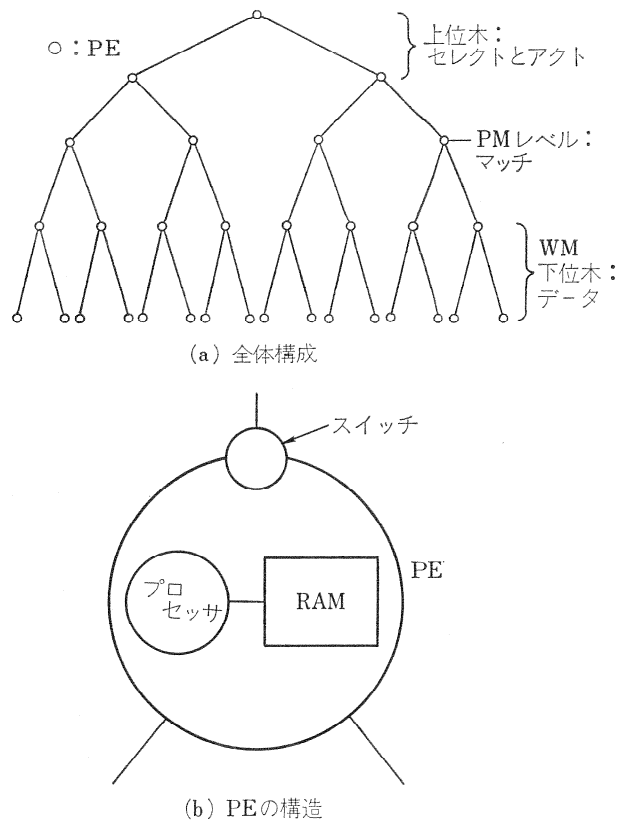
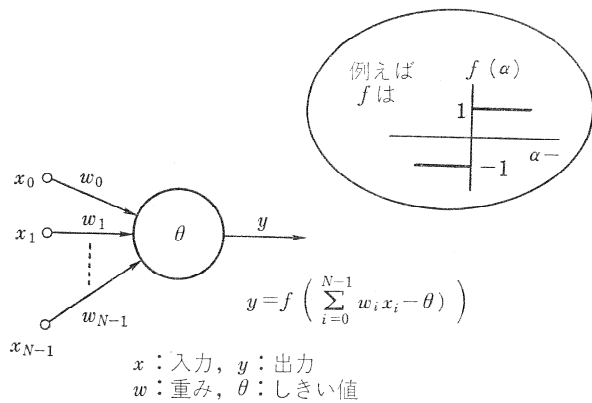


図9 DADOの構成 木構造の計算機である。PM レベルの PE に一つずつプロダクションルールが割り当てられる。

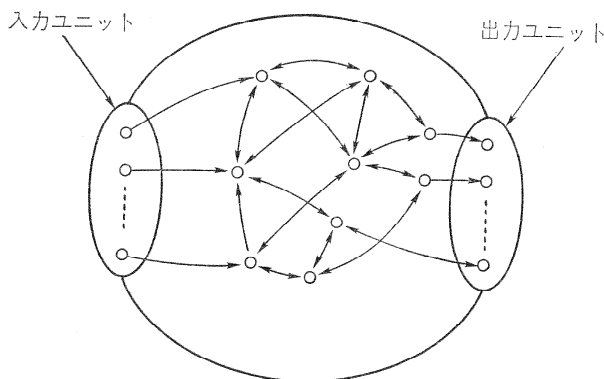
4. コネクショニストアプローチ<sup>(12)</sup>

AI は人間の知的活動を代行するものとして、いくつかの分野で成功を納めてきた。しかし人間の行っている情報処理とは大分異なっていることも確かである。例えば、人間はあいまいな情報や雑音の多い入力から認識を行うのが普通であるが、AI ではそれらの全くないクリーンな情報の下でしか処理が行えない。また人間は大量の知識から現実的な時間内に答を導けるが、AI ではそれが簡単にはいかない。このような状況判断から、AI を従来と違ったもっと人間の処理に似せた方法で処理しようという試みが行われだした。これはコネクショニズムとかニューラルネットワークなどとよばれて、現在各所で研究が開始されている。これは今のところどのように発展し、成功するのか失敗に終るのかもわからないが、一つの新しい動きとしてここに紹介する。

コネクショニストシステムは人間の脳を模して、神経細胞に相当する簡単なユニットと、それらを結合する“重み”をもった“結合リンク”で構成される。各



(a) ニューラルユニットの動作: 重み付けられた入力の総和がユニットのしきい値を越えると出力を出す



(b) 入力ユニットに、ある入力パターンを与えると、出力に対応した出力パターンが現れる。これは連想操作と考えられる。このような回路では、入力パターンが少しぐらい雑音で乱れても出力に正しい答が得られる

図 10 ニューラルネットワーク

ユニットの動作は入力の総和をとり、総和が各ユニットのしきい値を越えたとき、出力を出す。このようなネットワークの“重み”をうまく設定すると、入力ユニットにある入力パターンを入れたとき出力ユニットに対応する出力パターンが出るようになる(図 10 (b))。これは一種の連想メモリである。このようなシステムでは入力パターンに雑音に乗っていても出力に目的の答を出してくれることや、一部のユニットに故障が起こってもシステムに大きな影響を与えないなど、人間とよく似た性能を示す。このような回路を使ったセマンティックネットワークマシンやプロダクションシステムマシンの提案がある<sup>(13)</sup>。図 11 はニューラルネットを用いたプロダクションシステムマシンの提案である。このマシンの扱えるプロダクションルールは左辺が 2 項目に限られ、次のような形式を採る。

$$c_1, c_2 \Rightarrow a_1, a_2, \dots$$

図 11 の WM, C1, C2, R, B の各ブロックはニューラルネットで作成され、それらには相互結合がある。プロダクションルールの条件部 ( $c_1, c_2$ ) は WM のユニットのアクティブパターンとして格納される(例えば、“あるルールの  $c_1$  は WM のユニット 0,1 をオンにする”というように条件が格納される)。そのため、条件の数が増えると WM 内のオンのユニット数が増えていく。C1, C2 は WM と同数のユニットをもち、WM のユ

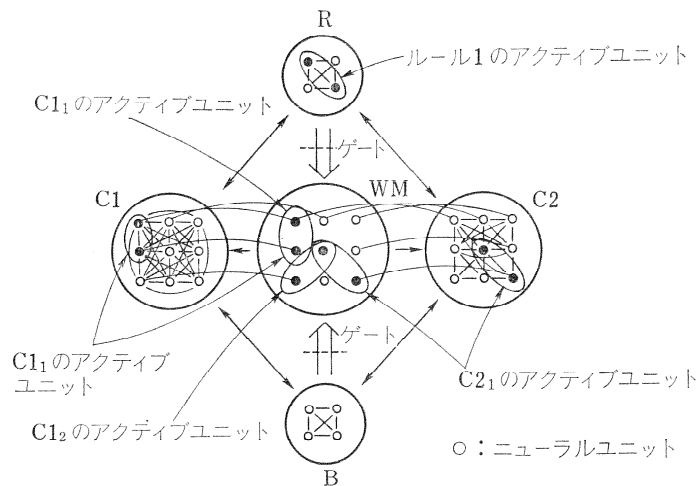


図 11 ニューラルネットを用いたプロダクションシステムマシン マシン構成は五つのニューラルネット(R, C1, C2, B, WM)が相互結合されている。上図は二つのプロダクションルール ( $\langle C1_1, C2_1 \rightarrow A1_1 \rangle, \langle C1_2 \rightarrow A1_2 \rangle$ ) が格納されている様子を示している。プロダクションルールの左辺部は WM にユニットのアクティブパターンとして格納される。C1, C2 のユニットは WM のユニットと一対一に対応し接続されているが、C1, C2 内にはユニット間結合があり、C1 には  $C1_i$  のうちの一つのパターン、C2 には  $C2_i$  のうちの一つのパターンのみがアクティブされる。R のユニットは C1, C2 のユニットと相互結合され、R 内にはルールのうちの一つのパターンがアクティブされる。一つのルールのルールパターンが R に現れ、その左辺部のパターンが C1 と C2 に現れたとき、系が安定しルールが選択されたことになる。選択の後にはゲートを開けてルールの右辺部を実行(一般には WM への書き込み)する。

ユニットと一対一に接続されている。但し C1 内ではユニット間に相互接続があり、WM に多数の条件が格納されていても  $c1$  の内の一つの条件に対応するアクティブパターンのみが現れるようにユニット間の“重み”がセットされている。C2 も C1 と同様でいくつものルールのうち一つの  $c2$  のアクティブパターンが現れる。更に R には常にただ一つのルールパターンしかアクティブにならないようにしておき、C1、C2 と相互結合しておくことで C1、C2 に現れる条件から一つのルールのアクティブパターンが生成される。これにより左辺部を“真”にする一つのルールのみが選ばれることになる。R から C1、C2 への結合もあるため、異なるルールの  $c1$ 、 $c2$  が C1、C2 に生成されていても R からの影響で修正され最終的に一つのルールが選択される。以上でマッチ、選択がすむので、最後に右辺部を WS に書き込めばよい。なお図中の B ブロックは左辺部に変数が有る時に用いられる。

## 5. おわりに

本稿では現在注目を集めている代表的な並列 AI マシンを紹介した。数年前までこの種のマシンは夢物語という感じを与えたが、コネクションマシンが登場してにわかには現実化し、開発に拍車がかかっている。今後 VLSI 技術の進歩と共にこの分野の研究開発は加速されよう。コネクションニズムは従来と異なる AI アプローチとして興味深い。今後の研究の進展が期待される。本稿は専門外の方を対象に、わかりやすさを重視したため多少厳密さに欠ける部分や、単純化を行った部分があることは御理解願いたい。

## 文 献

- (1) Kai Hwang, et al.: "Computer architectures for

artificial intelligence processing", IEEE Computer (Jan. 1987).

- (2) 島田: "データフローマシン", 情報処理, 28, 1 (昭 62).
- (3) 小長谷, 山本: "関数型言語とリダクションマシン", 情報処理, 26, 7 (昭 60).
- (4) K. Murakami, et al.: "Research on parallel machine architecture for fifth-generation computer systems", Computer, 18, 6 (June 1985).
- (5) S.E. Fahlman: "Design sketch for a million NETL machine", Proc. of First Annual National Conf. on AI (1980).
- (6) S.E. Fahlman: "The Hashnet Interconnection Schemes", Technical Report, CMU (1980).
- (7) W.D. Hillis: "The connection machine", MIT Press (1985).
- (8) D.L. Waltz: "Applications of the connection machine", IEEE Computer (Jan. 1987).
- (9) T. Higuchi, et al.: "The IX Supercomputer for Knowledge Based Systems", FJCC (1986).
- (10) S.J. Stolfo and D.P. Miranker: "The DADO production system machine", J. Parallel and Distributed Computing, 3, 2 (June 1986).
- (11) D.E. Shaw: "NON-VON's applicability to Three AI Task Areas", Proc. Ninth International Joint Conference on Artificial Intelligence (IJCAI) (Aug. 1985).
- (12) S.E. Fahlman and G.E. Hinton: "Connectionist architectures for artificial intelligence", Computer, 20, 1 (Jan. 1987).
- (13) D.S. Touretzky and G.E. Hinton: "A Distributed Connectionist Production System", CMU Tech. Rep. CMU-CS-86-172 (1986).



古谷立美

昭 46 成蹊大・工学。昭 48 同大学院修了。同年工業技術院電子技術総合研究所入所。以来、マイクロプロセッサや AI マシンの開発に従事。現在、電子計算機部主任研究官。工博。著書「ソフトウェア指向アーキテクチャ」(共著)など。