

記憶の活性化とその手法†

—連想プロセッサ NOAH の設計—

嶋津好生^{††} 田町常夫^{†††}

自然言語理解システムを構築するのに必要な、日常的な知識の表現と利用とに関する活性化意味ネットワーク (ASN)・モデルを提唱し、そのモデルが原理的に含む並列処理能力を実現するために連想プロセッサ NOAH を設計している。一般に意味ネットワーク (SN) は特殊な文脈の下での知識の統合の状態を表現しやすい知識の表現法である。したがって環境から受け入れられた情報を再認するのが SN の役割だといえよう。ASN モデルは知識の表現法として SN を採用し、SN を活性化することによって知識の利用を制御する機構を実現した。SN のいま注目されている部分を励起領域と呼び、励起領域の移動、拡散、集約、転移を励起動態と呼ぶ。知識の利用は SN の励起動態を制御することによって実現される。連想プロセッサ NOAH は分配論理記憶の原理を使って設計されたセル・システムである。一つのセルは SN の一つの節点とその隣接弧のいくつかを格納する。このようなセル・システムにネットワーク構造をうまく引き写して SN データを配置格納する。NOAH セル・システムに格納された SN の励起動態を制御する基本命令系を本稿に与えている。そのなかにはセル格納データと入力データとを照合する match 命令や励起状態のセル間移送を制御する transfer state 命令などが含まれる。また、NOAH セルの回路構成や ASN モデルに沿った NOAH プログラムの具体例を示す。

1. まえがき

自然言語を理解するシステムを構築するには、システムに日常的な知識を賦与する必要がある。筆者らは知識の表現法に意味ネットワークを採用し、日常的知識の範囲の想起や再認や推論に関する記憶のモデルを提唱した。それを活性化意味ネットワーク・モデルと呼んでいる¹⁾⁴⁾。知識は人間によってであれコンピュータによってであれ、いわゆる記憶されているものに違いがない。そして記憶されたものは、必要に応じて、つねにいま必要とされている部分だけが注目されるものである。このことを「記憶の賦活」(memory activation)と呼ぶ。また、このような賦活の機能をもった記憶機構を「活性化された記憶」と呼ぶことにする。記憶の賦活は、記憶の主体の認識作用と行為の発現とを結ぶ機構を考えるうえで重要な枠組みを与える。活性化意味ネットワーク (略して ASN)・モデルは、記憶を活性化することの重要性に着目して考えられた記憶のモデルである。

モデルの要点を述べると次のようになる。知識は多くの要素的概念のネットワークで表現される。概念ネットワークが部分的に注目されることを「励起され

る」といい、また外界からの情報が賦与されることを「特定化される」という。このモデルで「賦活される」といえば、励起されたり特定化されたりすることを総称していつている。ASN モデルは、概念ネットワークの賦活領域の移動集散を制御することによって、日常的知識の想起や再認や推論を実現する。

本稿は、記憶の活性化の必要性とその手法について述べるとともに、実際に概念ネットワークを賦活制御するための特殊な連想プロセッサを設計し、これを NOAH (network-organized association handler) と呼ぶ。NOAH は分配論理記憶⁵⁾の原理によって設計されている。一般に分配論理記憶はセル・システムであり、データは各セルに分割格納される。また各セルは一つのフラッグをもち、そのセルが活性 (active) か休止 (quiescent) かいずれの状態にあるかを表示する。分配論理記憶では全セルが一つの命令で並列に制御されるが、おもな命令では活性なセルのみが制御を受ける。このことが ASN モデルの基本原則にうまく合致している。連想プロセッサ NOAH においてもおもな命令によって活性なセル群のみが操作を受ける。その活性なセル群は ASN の励起領域に対比して考えることができる。

本稿はこのあと、2章において ASN モデルを概説し、記憶の活性化の意義と手法とについて議論する。3章において連想プロセッサ NOAH の基本命令系を示し、NOAH セルの回路設計を行う。4章において ASN モデルに沿ったプログラムの具体例を示す。

† Memory Activation and Its Implementation: Design of the Associative Processor NOAH by YOSHIO SHIMAZU (Faculty of Engineering, Kyushu-Sangyo University) and TUNEO TAMATI (Interdisciplinary Graduate School of Engineering Sciences, Kyushu University).

†† 九州産業大学工学部

††† 九州大学大学院総合理工学研究科

2. 活性化意味ネットワーク・モデル

2.1 ASN モデルに対する留意点

ASN モデルは次の諸点を特徴とする。

1) 意味ネットワーク (SN) は概念のネットワークである。その役割は知識を連合 (associate) し統合 (integrate) するところにある。

2) ASN モデルは、知識の一つの表現法として見ると、フレーム (frame) 理論⁶⁾ のように知識の意味をなすユニットを陽に表現することをしない。ASN モデルにおいては有意味ユニットは励起されないかぎり潜在化しており、記憶の賦活過程においてはじめてときの経過とともに徐々に顕在化する。このような意味において、ASN モデルは知識表現に関する徹底した過程説だといえるのである。

3) ASN を賦活制御する実際の手続きを実現しなければならない。たとえば、ASN に埋め込まれているパターンをなす部分ネットを識別したり、あるリンクを経由して励起状態を移動させたりする。ASN モデルはこれらの手続きを、いわゆる、ローカル・アルゴリズムで実現する。ローカル・アルゴリズムとは筆者らの定義⁴⁾ によるもので、知能グラフ⁷⁾ オートマトンのネットワーク⁸⁾ セルラ・グラフ・オートマタ⁹⁾ あるいは分配論理記憶などに共通している、体系的な、アルゴリズムの構成法を意味する。

4) 人工知能の自発的な処理機構 (spontaneous computation) を実現するのに一般によくプロダクション・システム (production system, 以下 PS と略す) が採用される。記憶の賦活に関する諸過程も自発的に発現するから、その実現に PS の採用が考えられるのは自然である。ASN モデルは、概念のネットワークのなかにプロダクション・ルールを埋め込み、そのことによって PS を包摂することができる。SN はおもに外界からの情報を認知 (cognition) あるいは再認 (recognition) するために必要な知識を表現する手段である。このことと ASN モデルが PS を包摂することとを考え合わせれば、ASN モデルによって PS に認知的拡張がほどこされたことになる。

以下、上記の諸点について少し詳細な説明を加えたい。

2.2 意味ネットワークの役割

意味ネットワーク (SN) には複雑な概念を表現する能力がないという評価は再検討する必要がある。SN は図的表現法であることを除き、とりたてて特徴があ

るわけではない。したがって、他の方法で表現できる知識ならば SN にも表現できるはずである。むしろ、SN への期待が強まり必然的に SN が複雑な表現形式を含むようになったにもかかわらず SN を積極的に利用する機構の開発が手がけられていないところに問題があるのではないだろうか。

SN は一般的な推論規則なども表現できるが、むしろ特殊な文脈を反映した日常的な事象を表現するのが役目である。すなわち、変数を異にする推論規則を一つ一つ個別に貯えることもできるが、それに加えて、それらに共通の変数を与えてよく起こる特殊な推論の道筋すなわち因果関係の連鎖をつくり、大きく連合した知識のユニットとして貯えることもできるのである。意味をなすユニットは、要素的概念が格構造などを表すリンクで結合され、一つのまとまりを形成している。そしてそういった有意味ユニットがいくつか集まり、それぞれは内部構造を変えないで、節点の変数あるいは変数値を共通のものにそろえて相互に節点をかさね、因果関係を表すリンクなどでユニット同士結合されていく。このようにして SN は知識を連合し統合していく。知識の統合は個々の文脈の下でこそ可能となる。したがって、SN の役割はおもに外界からの情報に含まれる特殊な文脈の再認にあるといえる。

2.3 有意味ユニットの取扱い方

さきに、SN が有意味ユニットを陽に表現しないと述べた。このことについて簡単な例を使って説明しよう。いま、次のような文章で表現される知識について考える。

「春彦は史彦にテニスラケットをゆずった。

だからそのラケットはいま史彦のものである。」

これを R. C. Schank のいう概念依存表現¹⁰⁾ で表すと図1のようになる。ただし、図1がこの文章の意味する内容と同じになるには、図1の各節点があつスロット (slot) を次のように満たす手続きを経由しなければならない。

P1=春彦 P2=史彦 X1=テニスラケット

Schank によれば、言語レベルの動詞で表現される行為は数少ない要素的な行為に分解でき、したがって、概念レベルの表現はそれらの要素的な行為の合成によって表現されるとしている。ATRANS はそのような要素的行為の一つであり所有権や支配権の委譲を表す。概念の統辞規則として Schank があげた典型的なものが概念化構造 (conceptualization) と呼ばれるもので、それには行為を表すものと状態を表すものとの2種類

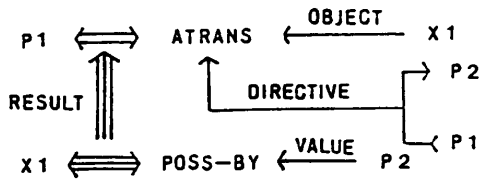


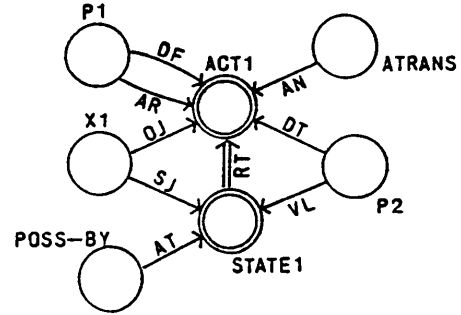
図1 R. C. Schank の概念依存表現
 「P1はP2にX1をゆづった、だからX1はいまP2のものである。」
 Fig. 1 Conceptual dependency representation by R. C. Schank.
 "P1 gave X1 to P2, and so X1 is now possessed by P2."

が区別される。図1に、前者は2重線両方向矢印で示された部分に、後者は3重線両方向矢印で示された部分に、それぞれ現れている。行為の概念化構造は格構造をそのおもな構成要素とする。また図1に現れている POSS-BY は被所有の状態を表す概念である。

図1が表現する同じ内容を節点と有向弧とから成る典型的な SN 表現に書き直すと図2のようになる。この表現では同一のラベルをもつ節点が2箇所以上のところに重複して現れることを禁じている。またこの図で節点を1重丸と2重丸のものに、有向弧を1重線と2重線のものに区別しているが、この区別は説明の便宜のためであり、グラフ構造として見れば同格である。さて図1で明瞭に表現されていた概念化構造が図2では、たしかに、明示されなくなっている。このように潜在化しているものを改めて顕在化する方法は賦活の手続きすなわち注目するための動的過程以外にはない。図2をさらに図3のように書き直してみればこのことがより明らかになるであろう。ここで方形で囲った部分は、一つの節点とその隣接弧のすべてから成るユニットであり、節点セルと呼ばれる。各隣接弧は、その節点からその隣接弧を経由する隣接節点へのポインタを伴う。SN は節点セルの集合として形式的に表現される。ASN モデルにおいて陽に表現されるユニットは節点セルだけである。いうまでもなく、節点セルは有意義ユニットではない。ASN モデルが備えている ASN の賦活制御命令系に対して、節点セルはその操作対象の単位とされている。ASN の賦活過程の流れにしたがって順次励起される節点セルの部分集合が、知識の有意義ユニットとして顕在化するのである。

2.4 賦活制御命令系の機能

分配論理記憶はすべての記憶セルに同じ操作を同時にほどこす。この手法を採用して、すべての節点セル



AN : ACTION AR : ACTOR OJ : OBJECT
 DF : DIRECTIVE-FROM DT : DIRECTIVE-TO
 AT : ATTRIBUTE VL : VALUE
 SJ : SUBJECT RT : RESULT

図2 典型的な意味ネットワーク表現
 Fig. 2 In this figure, the content expressed by Fig.1 is represented by the typical method of the semantic-network-representation. This method can not state clearly the conceptualizations. They hide themselves among the graph-structure.

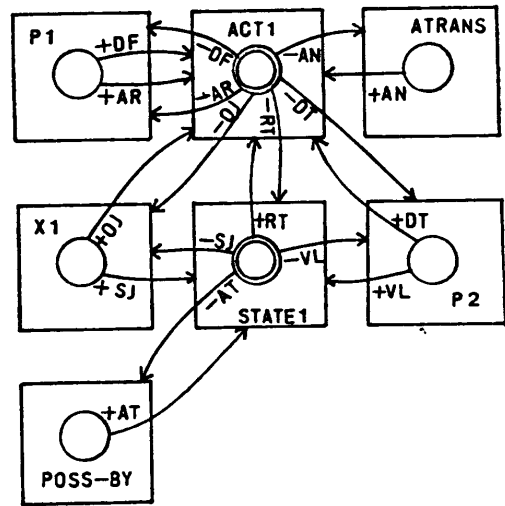


図3 節点セルの集合
 Fig. 3 A unit composed of one node and all of its adjacent arcs is called a node-cell. The arcs are accompanied by the pointers to the nodes adjacent via themselves. The semantic network can be formally represented by the set of the node-cells.

に同じ操作を同時にほどこす命令系を ASN モデルに与えている。ASN を賦活制御する手続きはこの命令系を使って実現する。ASN モデルでは、注目された節点セルを「励起された節点セル」と呼ぶ。「励起された節点セル」は分配論理記憶における「活性な記憶

セル」に対比して考えることができる。われわれが設計する分配論理連想プロセッサ NOAH においては ASN モデルにおける用語のほうをそのまま踏襲する。

図2の例に表現された内容は全体の SN 記憶のなかの一部分である。いま、

「春彦は史彦にテニスラケットをあげた。」

という文章を受けると、記憶の主体はこの文章を概念の表現に変換する手続きを経由したうえ、その概念表現を使って P1, P2, X1, ATRANS, ACT1 の部分を励起し、またそれらのスロット (slot) を次のようなフィラ (filler) で埋める。

P1=春彦 P2=史彦 X1=テニスラケット

この状態から、

「そのテニスラケットは史彦のものである。」

ことが推論される手順は次のとおりである。すべてが賦活制御プログラムによって実行される。さきに述べた五つの節点が現在励起状態である。まず、この励起領域のなかから代表節点 (ACT1) だけ励起状態のまま残し、他の節点は励起状態でなくす。このことをもって「励起領域を代表節点に集約した」と表現することもできる。続いて RESULT リンクを通してその励起状態を節点 STATE1 へ移す。そして最後に、概念化構造を構成するリンクを通して節点 STATE1 の励起状態を節点 X1 や P2 や POSS-BY へ拡散する。ASN モデルは以上のような過程を映画の画面を作る要領で実現する。すなわち、節点セルの全集合が画面を構成し、一連の賦活制御命令が各節点セルに変化を起して画面を作り出す。

以上の説明でも明らかなように、ASN モデルは知識を節点単位で識別する。したがって、SN にどんなに複雑な有意味ユニットを埋め込むことになっても、その構造が認知されているかぎり一定の手続きで識別することができる。

2.5 ASN モデルによる PS の認知的拡張

ASN モデルは記憶を活性化することによって、外界から受け入れた情報を再認する機構を実現している。そのうえさきに述べたように、ASN はそのなかでプロダクション・ルールを埋め込むことができる。また、その部分が励起されると PS が発現するような機構を構成するのは容易である。すなわち ASN モデルは、宣言的知識と手続き的知識とを ASN のなかと同じ表現形式で混在させ、しかもその記憶機構を活性化することによって認識と行為とを連合しようとする

試みである。

知識の利用には古い知識を賦活し新しい知識を生成するという二つの局面がある。PS はプロダクション・ルールによって生成の局面を効果的にした結果、賦活の局面が十分反映されていない。プロダクション・ルールはルールが起動される条件の集合とその条件が満たされたとき実行される動作の集合との対から成る。ここで前者はあくまで後者が発現する条件を与えたものであり、一般的には心理学的現象である本来の認知あるいは再認などの役割をはたしているものとはみなしがたい。また PS の制御機構はルール適用の優先順位やバックトラッキングなど単純なものにとどまっている。したがって、複雑多岐にわたる認知や再認の機構を表現することはむずかしく、認識システムを実現するには不適當である。一方、ASN モデルは知識の賦活の局面を十分に活かしているから認識システムの実現に適應性を示すとともに、PS を包摂して知識の生成の局面についても考慮されている。

また、ASN モデルは概念ネットワーク記憶を意識の座とする仮説を立てた。意識を概念ネットワーク記憶の励起領域だと解釈したことが PS のもう一つの認知的拡張となっている。たとえば連想のような記憶の事象を PS で実現するとすれば、プロダクション・ルールの実行によって生起させることになるが、このとき疑問がのこる。適用されたプロダクション・ルールは意識されているのだろうか。また作業記憶の内容は全部がいつも意識されているのだろうか。つまり PS は意識の座については考慮外のこととしてきたのである。

3. 連想プロセッサ NOAH の設計

3.1 連想プロセッサ NOAH の必要性

NOAH は活性化意味ネットワーク (ASN) ・モデルを実現するために設計された分配論理連想プロセッサである。ASN モデルが原理的に含む並列処理能力を実現するには分配論理記憶の原理を採用するのが適當と判断される。ASN モデルは当初、自然言語を理解するために必要な日常的な知識を表現し利用するための一つの知識情報処理モデルであった。しかし考察が進むにつれて、ASN モデルが、記憶を活性化するという新しい枠組みのなかで PS の認知的拡張をはたし、もっと一般的な知識の利用形態を制御する機構を与えうることが明らかになった。ASN モデルを従来のコンピュータ・リソースで実現する場

合、現状ではプログラム言語 LISP を採用することが考えられる。このとき、各賦活制御命令は LISP 関数として定義される。しかし LISP には並列処理能力が備っていないから、節点セルの集合の次の状態を作り出すのにすべての節点セルに同じ操作を繰り返して実行しなければならない。概念ネットワークの賦活制御機構が広範な知識情報処理システムへ応用できることを考慮すれば、あえて、すべての節点セルに並行して命令を実行できる分配論理記憶を開発することは有意義であると考えられる。NOAH はセル・システムであり、一つのセルには一つの節点とその隣接弧のいくつかのラベルを格納し、NOAH 基本命令系によって全セルを single instruction stream multiple data stream (SIMD) のモードで並列に制御する。

3.2 NOAH セル・システムへの ASN の格納

図4に NOAH セル内部のレジスタ構成を示す。1セルに節点ラベルが一つとその節点に隣接する弧のラベルが m 個格納される。図4は $m=3$ の場合である。節点ラベルは NSS (node storage space) と呼ばれるレジスタに、弧ラベルは ASS (arc storage space) と呼ばれるレジスタに格納される。NSS や ASS をただストレージ・スペースと呼ぶこともある。

図5に NSS や ASS のデータフォーマットの一例を示す。一般に1ストレージ・スペースを n ビットとする。図5は $n=24$ の場合である。各ビットの位置を指すには D1, D7 などと書く。

ストレージ・スペースの特定の部分がセル回路の他

の部分と関わり特別な働きをする。NSS と ASS とに共通して D1 と D2 は励起痕跡ビット (excitation saving bits) と呼ばれる。ASS において、D6~D3 の4ビットは肢選択ビット (limb selection bits), D8 と D7 は弧方位ビット (arc direction bits) と呼ばれる。以下それぞれの働きについて述べる。

1) 励起痕跡ビット

ストレージ・スペースが励起されたあと、その痕跡を残すのに使われる。ストレージ・スペースの励起状態は、それぞれのストレージ・スペースに一つずつ備わった制御用フラグがセットされることによって示される (これらのフラグは励起状態を表示するとともに、記憶セルをさまざまに制御するのに使われる。詳細は3.4.2項セルの回路構成のところ述べる)。このフラグがリセットされるとストレージ・スペースが励起状態でなくなる。しかし、たったいままで励起されていたことを記録することができれば、それを賦活制御プログラムのなかでうまく利用し多彩な賦活動態を実現することができる。したがってそのことをストレージ・スペースのなかの励起痕跡ビットに記録するのである。

2) 肢選択ビット

ASS に格納されている弧ラベルが割り付けられるべき肢 (limb) を指定する。一つのセルから周辺のセルへ伸ばせる肢は、図6に示すように、たとえば16本とすることができる。図6ではセルがたてよこ等間隔になるように描いてある。このようにセル・システムを描いたとき、途中さえぎるものない直線で結ば

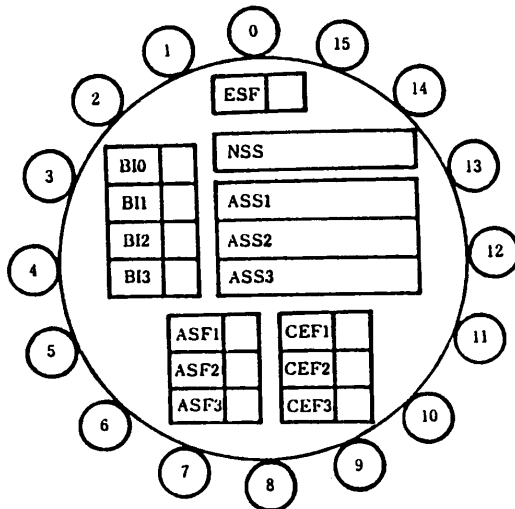


図4 NOAH セルのレジスタ構成

Fig. 4 Register-configuration of the NOAH-cell. The number of the arc-storage-spaces (ASS) is generally m . However $m=3$ here.

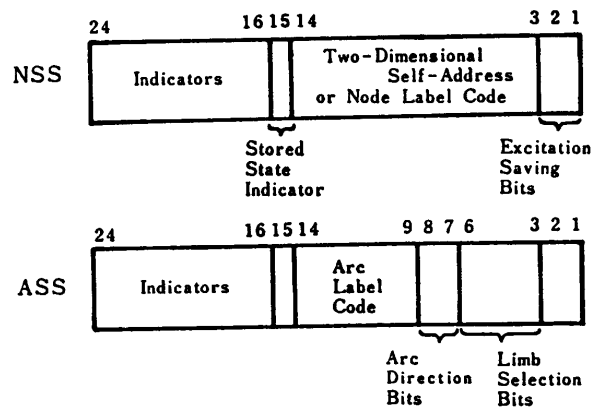


図5 NOAH セル・データ・フォーマットの一例

Fig. 5 An example of the NOAH-cell-data-format, where the usage-purposes of D1 and D2 in the node-storage-space (NSS), and D1~D8 in the arc-storage-space (ASS), are determined in the NOAH system.

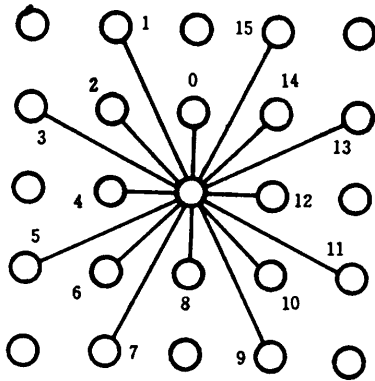


図 6 一つのセルから伸ばせる肢とその識別番号 ($l=4$ の場合)

Fig. 6 The number of limbs able to let grow out from a cell is generally 2^l . However it is 16 ($l=4$) here. In the figure, the identification-number is attached to each limb. The network-data are stored in the cell-system to have their arcs arranged on the limbs.

れるセル間にのみ肢が存在しうると考える。一般に、肢選択ビットを l ビットとすれば、セルは肢を 2^l 本もつことができる。ASSに弧ラベル・コードを格納するとき、同時に、その弧ラベルが割り付けられるべき肢を指定する l ビット・コードを肢選択ビットへ書き込む。このように、ネットワーク・データをセル・システムに割り付ける情報は各 ASS の肢選択ビットに格納されている。肢の選択について気を配らねばならないのはネットワーク・データを格納するときのみである。いったん格納されてしまえば、ASN を賦活制御するため NOAH 命令によってセル間で励起状態を移送する場合でも、肢について意識することなく、弧ラベルだけ指定すればよい。

3) 弧方位ビット

ASS に格納された有向弧の方向を表すのに使われる。二つの節点とその間を結ぶ有向弧とから成る三つ組構造を格納するには、肢で結ばれた二つの記憶セルが必要である。このとき二つの節点セルがそれぞれその有向弧に対してどの位置にあるか示さなければならない。(D8, D7) に格納される 2 ビット・コードがその位置を示す。

図7に、ネットワーク・データをセル・システムに実際に配置格納した例を示す。データは図3と同じものを使っている。一般に一つの節点セルの全体を単一の記憶セルで格納できないことがある。図7では1セルの ASS の個数 m を 3 としているが、節点セル ACT1 の隣接弧が 6 個であるから 1 セルに格納でき

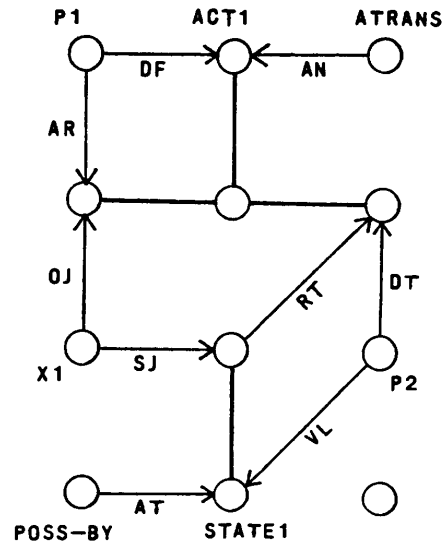


図 7 ネットワーク・データのセル・システムへの配置格納 ($m=3$ の場合の ASS の確保)

Fig. 7 An example of network-data is arranged on the cell-system. Let us consider the case that the number of the ASSs in one cell is 3. The node ACT1 is stored spending 4 cells to secure 6 necessary ASSs.

る弧の個数を超過している。図中の太線のリンクは、節点セル ACT1 の隣接弧のすべてを格納するため、肢で結ばれた四つのセルを使って ASS の必要数を確保したことを表している。太線の肢で結ばれたセル同士がそれぞれ一つの ASS を提供し、必要に応じて励起状態を互いに共有し合うことを表す弧ラベルを格納する。NOAH 命令によってこの弧ラベルのところを励起共有設定すると、賦活制御プログラムの実行中、太線の肢で結ばれたこれらのセルは励起状態を共有し同じ制御を受け、あたかも単一の記憶セルのようにふるまう。このことについては、3.3 節 NOAH 基本命令系や 4.2 節 “簡単なプログラミング” のところで詳述する。

3.3 NOAH 基本命令系

表1に各 NOAH 基本命令の機能を示す。この命令系によって、ASN の有意味ユニットを識別しながらネットワーク構造を自由自在にたどる手続きを表現し、その実行によってセル・システムに格納された ASN を賦活制御することができる。表1の選択性ビットについては、3.4.2 項の 5) NOAH 命令のマイクロプログラムや 4.1 節 NOAH 命令の記法のところで説明する。以下、表1を参照しながらおもな NOAH 命令について概略説明を加える。

transfer state 命令は 励起状態のセル間移送を制御

表 1 NOAH 基本命令系の機能
Table 1 The system of the NOAH fundamental commands and their functions.

| 命 令 | 選択性ビット | 命 令 の 機 能 |
|--------------------------|---|--|
| activate | (ET 1, ET 2) | その位置が明らかにされている特別に定められたセルの節点を励起する。 |
| store | (NSB, ASB) | 励起セルの節点や弧に入力データを格納する。 |
| put | (NSB, ASB) | 励起されているいかにかわらず、すべてのセルの節点や弧に入力データを格納する。 |
| add | (NSB, ASB) | 励起セルの節点や弧の格納データに入力データを加算する。 |
| read | (NSB, ASB) | 励起セルの節点や弧の格納データを出力データ線に読み出す。 |
| match | (NSB, ASB) (EQT, GRT, LET) (ET 1, ET 2) | 全セルの節点や弧の格納データを並列に、入力データと照合する。照合条件に合致した節点あるいは弧が励起される。match 命令の場合に限り弧の励起は同時にその弧を含むセルすなわち節点の励起状態をもたらす。 |
| clear excitation | (NSB, ASB) | 励起状態を解消する。 |
| set logical mode | (NSB, ASB) (CMB, DMB) | 励起痕跡の論理条件を設定する。 |
| clear logical mode | (NSB, ASB) | 励起痕跡の論理条件を解消する。 |
| transfer state | (EQT, GRT, LET) (ET 1, ET 2) | 入力データと全セルの弧の格納データとを照合し、その結果適合し励起された弧を通して、その弧を含むセルの節点の励起状態を隣接セルの節点へ移送する。励起状態を送り出した節点は他のセルから励起状態が送られてこない限り励起状態ではなくなる。 |
| propagate | (ET 1, ET 2) | 命令実行前にすでに励起共有設定されている弧を通して節点の励起状態を伝播する。ただし、propagate 命令一つで一つの弧だけしか伝播しない。transfer state 命令と異なり、励起状態を送り出した節点はもとの励起状態のままにとどまる。この命令はおもに励起共有設定の下で励起命令の働きを補佐するのに使われる。 |
| set sharing-excitation | (EQT, GRT, LET) (ET 1, ET 2) | 入力データと全セルの弧の格納データとを照合し、その結果適合した弧を励起するとともに、その弧を通じた隣接セルとの間を励起共有設定する。 |
| compose | | 命令実行前にすでに存在する励起弧のところで励起共有する。 |
| clear sharing-excitation | | 励起共有を解消する。 |

する。ASS の必要数を確保したり有意味ユニットの識別を容易にするため set sharing-excitation 命令と clear sharing-excitation 命令が使われる。前者が指定された弧ラベルのところを励起共有設定し、後者はそれを解除する。activate, match, transfer state, set sharing-excitation, propagate の 5 命令は選択性ビット (ET 1, ET 2) をもっている。これらの命令が実行されたあとには、通常、ASN の励起領域に変化が見られる。その意味でこれらの命令を励起命令と呼ぶ。さて、励起痕跡を残す手法は ASN を賦活制御する有効な手段となるが、それを行う役割を ASN に新たな励起状態をもたらす励起命令自身にもたせる。たとえば、励起痕跡を利用した論理探索を次のようにプログラミングする。プログラム中にブロックを設け、そのなかにあり励起痕跡を残すことを指定された励起命令が個別に ASN を励起する結果を累積して最終的にそれらの AND 条件あるいは OR 条件で励起痕跡が残るようにする。set logical mode 命令と clear logical mode 命令とでブロックの前後を挟み、前者で AND 条件あるいは OR 条件を指定する。

3.4 NOAH セルの回路設計

3.4.1 セルの諸操作

NOAH セル・システムの制御にはマイクロプログラム制御方式が採用される。図 8 にその制御部の構成を示す。セル・システムを制御する NOAH 命令はレジスタ CCR (cell control register) へ送られ、そこから直接セル・システムへ制御信号を送る。NOAH 命令には入力データを必要とするものが多い。IPR (input pattern register) は入力データを容れるレジスタである。入力データを必要とする命令の場合、プログラム上その命令が実行される直前に IPR を適切な入力データで設定する必要がある。

NOAH セルは次のような諸操作を実行する。

- 1) 入力データの格納
- 2) 入力データとすでに格納されているデータとの照合およびその他の算術・論理演算
- 3) 格納データの出力
- 4) 励起状態や励起共有の設定
- 5) 隣接セルへの励起状態の移送
- 6) 励起痕跡を残すこと

目下のところ NOAH セルの初期設計の段階として、入出力データや照合データがビット直列にセルに

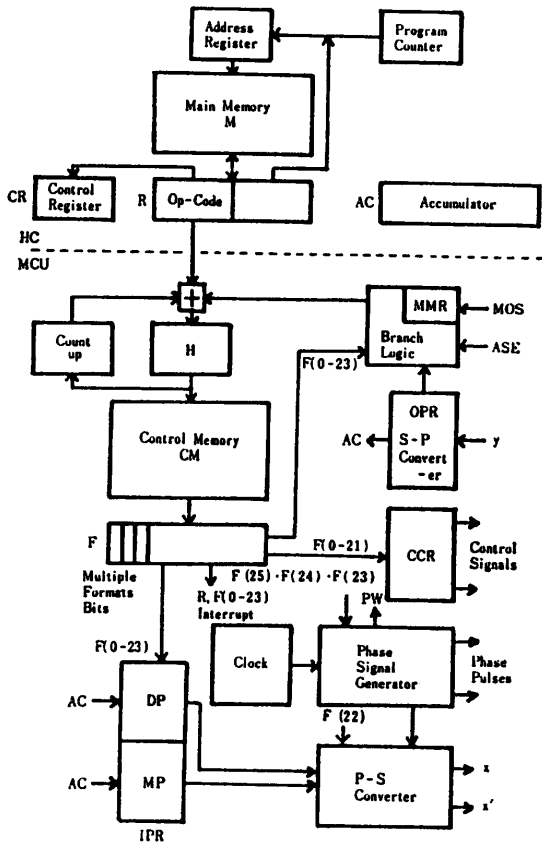


図 8 NOAH セル・システムの制御部

HC: ホスト・コンピュータ, MCU: マイクロプログラム・コントロール・ユニット

Fig. 8 The control-unit of the NOAH cellular system.

HC: host computer, MCU: microprogram control unit.

入出力されるように設計している。しかしこうすれば、入出力データ線を少なくできるし、また照合・算術・論理演算のすべてをもっとも簡単な回路構成で実行できる利点がある。便宜上、上記の 1) 2) 3) の諸操作を直列処理と呼んでいる。セルからの出力データはレジスタ OPR (output pattern register) へ送られる。直列処理のあとは上記の 4) 5) 6) の諸操作が順を追って実行される。それぞれがタイミング・パルス PX (励起状態や励起共有の設定), PY (励起状態の移送), PZ (励起痕跡を残す) によって起動する。

3.4.2 セルの回路構成

図 9 に NOAH セル全体の構成を示す。一つのセルのなかに 3 種類のモジュール、直列処理部 (serial processing section, SPS), 状態移送部 (state-transfer section, STS), 励起痕跡部 (excitation-saving section,

ESS) が含まれる。また図 9 に、弧ラベルを格納するときの ASS の自動選択回路を示している。以下、3 種類のモジュールと ASS 自動選択回路とを順を追って説明し、最後に各 NOAH 命令のマイクロプログラムを示す。

1) 直列処理部 SPS

前述したセルの諸操作のうち 1) 2) 3) を実行する。各 SPS はストレージ・スペースを一つずつ合んでいる。入力にマスク・データを入れることができ、したがって直列処理はストレージ・スペースの注目する部分だけにしばって実行できる。

match, transfer state, set sharing-excitation の 3 命令は照合操作を含む。入力データと格納データとの照合条件は選択性ビット (EQT, GRT, LET) によって指定する。これら三つの制御信号は順を追ってそれぞれ大小関係の $=, >, <$ を意味する。したがってたとえば、照合条件を格納データ \geq 入力データとしたときは $EQT=1, GRT=1, LET=0$ とすればよい。照合操作は制御信号 MAT によって起動する (表 2)。照合結果を示す信号 MS (matching signal) は各ストレージ・スペースの一つずつ備えられた制御用フラッグに記録される。すなわち、節点の SPS からはフラッグ ESF (exciting state flip-flop) へ、弧の SPS からはフラッグ ASF(i) (arc-selection flip-flop) へそれぞれ送られ記録される。とくに ESF がセット状態のとき、その記憶セルに格納されている節点セルは「励起状態である」という。ESF は SPS による照合結果や隣接セルからの励起状態の移送によってセット状態になる。また、弧への match 命令や transfer state 命令や set sharing-excitation 命令が実行されたとき、それらの命令の照合条件に合致した ASS のフラッグ ASF(i) ($i=1, 2, \dots, m$) がセット状態になる。そしてそのうち弧への match 命令のときのみ、セット状態の ASF(i) が存在する記憶セルではその ESF もセット状態になる。

2) 状態移送部 STS

transfer state 命令を受けて STS が、節点セルの励起状態を、指定の弧ラベルが割り付けられた肢を通して隣接セルへ移送する。励起状態の移送は制御信号 TRS によって起動する (表 2)。入力データと弧ラベルとを照合して条件に合致すればその弧の ASF(i) がセット状態になる。STS はその結果を受けて指定の弧ラベルが割り付けられた肢の励起状態移送路を開き、その弧ラベルが属する節点セルが励起状態であれ

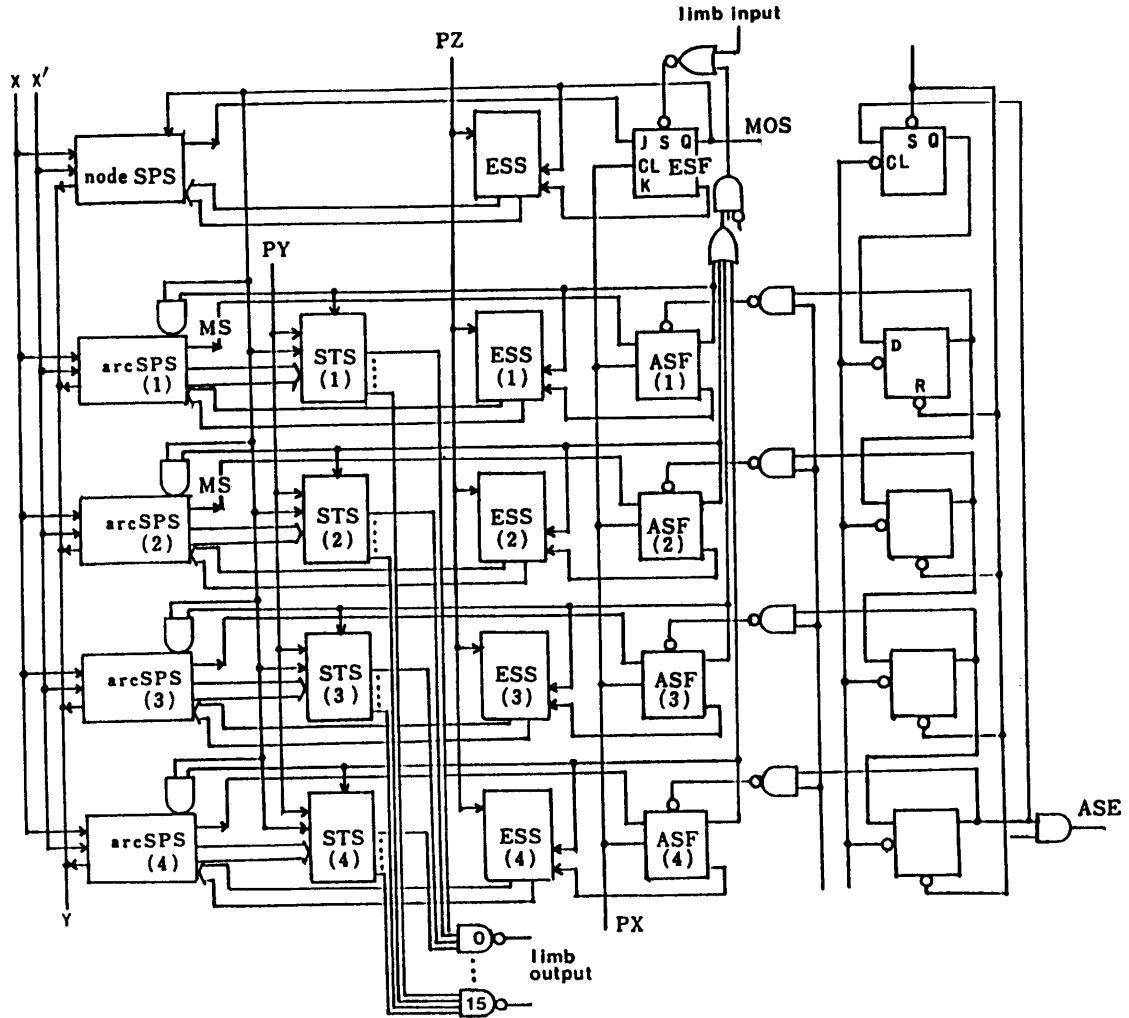


図 9 NOAH セルの回路構成 (m=4 の場合)

Fig. 9 Circuit configuration of the cell of the associative processor NOAH. In this case, it includes 4 ASSs.

ば、それを隣接セルへ移送する。弧の SPS に一つずつ ASS が含まれているが、その ASS に一つずつ STS が接続される。すなわち、記憶セルに含まれるすべての ASS に対応してそれぞれ 16 本 1 組となって励起状態移送路が出てくる。記憶セル全体としてはそのすべてを OR ゲートで集め 16 本の肢出力 (limb output) として 16 個の隣接セルに接続する。

set sharing-excitation 命令によって弧ラベルを指定し励起共有設定するのも STS の役割である。この命令の下で、指定の弧ラベルを入力データとする照合操作が起り、ASF(i) がセット状態になると、これを受けて STS は制御信号 SSE (表 2) の存在の下で、STS のなかにある別のフラッグ (図には示していない) をセット状態にする。その結果、その後の励

起命令の下で制御信号 MAT があるたびにこの弧を通して自動的に励起状態の伝播が起こるように、回路構成されている。

3) 励起痕跡部 ESS

励起命令によってストレージ・スペースが励起状態になったとき、これもその同じ励起命令の指示によってその痕跡を残すことができる。その役割をはたすのが ESS である。前述したように、この励起痕跡はストレージ・スペースの一部、励起痕跡ビット D1 や D2 に記録される。この励起痕跡を利用して論理探索を容易に実行できる。set logical mode 命令によって論理条件が設定されると、プログラム上そのあと何回か、指定された励起命令の下で励起痕跡を残す手続きが実行されるだけで、自動的にそれらの論理的な累

りの書き方がある。これは弧ラベル DF の弧方位ビットが (0, 1), (1, 0), (1, 1) の場合を表し、それぞれ有向弧の矢尾側、矢頭側、両側の位置を占める記憶セルの ASS を指摘する。また各命令の選択性ビットは、1 にセットされる制御信号の名称を括弧に囲んで表現する。

4.2 簡単なプログラミング

次に ASN の簡単な賦活制御プログラムを実際に書いてみよう。セル・システムに格納された ASN の具体例として図7を使う。まず、節点 ACT1 に代表される概念化構造を ASN 上で同定する手続きを考えよう。

```
clear excitation (NSB, ASB)
[L 0]→IPR
set sharing-excitation (EQT)
clear excitation (ASB)
```

ここで、(NSB, ASB) は選択性ビットであり、前者は節点を選び後者は弧を選ぶ。また2行目の [L 0] は、L 0 が ASS を確保するための弧ラベルを表し、この弧に対して両方向励起共有設定するための入力データを与えている。さてこの4行のプログラムで、節点 ACT1 を構成する四つの記憶セルが励起共有設定された。次に節点の論理探索モードを AND 条件に設定する。

```
set logical mode (NSB, CMB)
```

次に節点 P1 と関係 AR にあるものとして節点 ACT1 を識別する手続きを考える。ただし、この時点でその節点のラベルが ACT1 であるとわかっているわけではない。

```
P 1→IPR
match (NSB, EQT)
+AR→IPR
transfer state (EQT)
propagate
propagate (ET 1)
clear excitation (NSB, ASB)
```

これで P1 と AR の関係にあるものとして、励起共有設定された ACT1 の記憶セル群を識別したことになる。結果は #1 タイプの励起痕跡で示されている。なおこの場合、propagate 命令を三つ以上続けても結果は同じである。これと同様に ACT1 はほかにも、節点 P1 と関係 DF、節点 ATRANS と関係 AN、節点 X1 と関係 OJ、節点 P2 と関係 DT をもっている。それぞれの関係は、上記の P1 との関係 AR の

場合と同様の手続きで確かめられ、しかも AND 条件で次々に識別条件を付加していく。こうして ASN 上で節点 ACT1 が同定される。この例に見られるように、外界に注目しより多くの情報を収集するにつれて、内部の記憶への注目もしだいに収束していく。

次に節点 ACT1 で代表される概念化構造からリンク RT を通じて励起状態を移送することにより一種の推論を実現する手続きを考えよう。図7に描かれた範囲では、節点 STATE1 で代表される概念化構造が励起される。まず、概念化構造の内部において代表節点から構成要素節点の方向へ励起共有設定する。

```
clear excitation (NSB, ASB)
-L 1→IPR
set sharing-excitation (EQT)
clear excitation (NSB, ASB)
```

ここで、2行目の -L1 は、L1 が概念化構造の構成要素を示す弧ラベルを表し、この有向弧において矢の方向とは反対の方向に励起共有設定するための入力データを与えている。いま、節点 ACT1 に #1 タイプの励起痕跡が残されているものとする。

```
<D 1=1>→IPR
match (NSB, EQT)
-RT→IPR
transfer state (EQT)
propagate (ET 2)
```

ここで、1行目に出てくる <...> はビット単位で入力データを与える記法である。この5行のプログラムが実行された結果、節点 STATE1 に代表される概念化構造が励起されており、またそこに #2 タイプの励起痕跡を残すことができた。

以上、その片鱗しか示すことができないが、このような NOAH プログラミングによって、複雑な文脈を指定された場合も容易に ASN の目的の部分にアクセスできるし、また励起領域の移動、拡散、集約、転移などを自由に制御し任意の賦活動態を実現できる。具体的な応用的課題を考えると、たとえば次のような課題が考えられるであろう。われわれ人間の場合、「連想的準備 (associative priming)」¹¹⁾ という心理現象がある。すなわち、ネットワーク構造に連合された知識に基づいて知識の無意識の準備が起り、たとえば談話理解の場面で、不正確な情報が入ってきたときその欠けた情報を補い、また次の話を聞かないうちにその内容を予測したりすることができる。このような「連想的準備」も適切な概念ネットワークのデー

タと賦活制御プログラムによって実現可能である。

5. む す び

筆者らは認識システムを実現しやすくするため PS に代わる知識の表現法を模索して、意味ネットワークを活性化することの着想を得た。活性化意味ネットワーク (ASN)・モデルは記憶を活性化することによって想起や再認や推論の機構を実現する。ASN モデルは PS の認知的拡張を行っている。したがって、概念ネットワークのデータや賦活制御プログラムの開発を行うことが今後の応用的課題となる。

本稿は ASN モデルの機能とその意義を述べるとともに、ASN モデルが原理的に内蔵している並列処理能力を実現するために特殊な連想プロセッサの設計を試みたものである。それを NOAH と呼んだ。NOAH セル・システムにネットワーク・データがいったん格納されると、自在なデータ・アクセスを実行する賦活制御プログラムが容易に得られる。しかし、格納するときの煩雑さについてはまだ十分な評価を行っていない。いかにしてその煩雑さを克服し実用的な ASN メモリ・システムを実現するか今後の課題である。

NOAH セル・システムのネットワーク・データ格納法は、データ構造の隣接性を直接物理的な隣接性に引き写している点を特徴とする。記憶容量を補うため、NOAH セル・システムを2次記憶と結合し必要に応じてデータの入れ換えを行う方法を考えるとき、このことが NOAH の有利な条件となる。NOAH と2次記憶との結合方式については稿を改めて報告する。

参 考 文 献

1) 嶋津, 田町: 概念記憶の意味ネットワークモデル, 九州大学大学院総合理工学研究科報告,

Vol. 3, No. 2, pp. 95-107 (1981).

- 2) 嶋津, 田町: 意味ネットワークの静態構造論, 情報処理学会論文誌, Vol. 23, No. 1, pp. 50-57 (1982).
- 3) 嶋津, 田町: 意味ネットワークの賦活動態を制御するプログラミング言語, 情報処理学会論文誌, Vol. 23, No. 3, pp. 226-234 (1982).
- 4) 嶋津好生: 概念記憶システムの研究, 九州大学出版会, 福岡 (1982).
- 5) Lee, C. Y. and Paull, M. C.: A Content Addressable Distributed Logic Memory with Applications to Information Retrieval, Proc. IEEE, Vol. 51 (1963).
- 6) Minsky, M.: A Framework for Representing Knowledge, in Winston, P. H. (ed.): *The Psychology of Computer Vision*, McGraw-Hill, New York (1975).
- 7) Rosenstiel, P., Fiksel, J. R. and Holliger, A.: Intelligent Graphs: Networks of Finite Automata Capable of Solving Graph Problems, in Read, R. C. (ed.): *Graph Theory and Computing*, Academic Press, New York and London (1972).
- 8) Fiksel, J. R.: A Network-of-Automata Model of Question-Answering in Semantic Memory, Stanford University, Ph. D. (1973).
- 9) Wu, A. and Rosenfeld, A.: Cellular Graph Automata I. Basic Concepts, Graph Property Measurement, Closure Properties, II. Graph and Subgraph Isomorphism, Graph Structure Recognition, *Information and Control*, Vol. 42, pp. 305-353 (1979).
- 10) Schank, R. C.: *Conceptual Information Processing*, North-Holland/American Elsevier, New York (1975).
- 11) Anderson, J. R.: *Cognitive Psychology and Its Implications*, W. H. Freeman and Company, San Francisco and London (1980) (富田他 (訳): 認知心理学概論, 誠信書房, 東京 (1982)).

(昭和59年1月9日受付)

(昭和59年7月19日採録)