

概念記憶の並列処理

PARALLEL PROCESSING OF CONCEPTUAL MEMORY

連想プロセッサによる活性化意味ネットワークモデルの実現

Implementation of the Activated-Semantic-Network Model
by means of an Associative Processor

嶋 津 好 生

Yoshio SHIMAZU

九州産業大学 工学部

Faculty of Engineering Kyushu-Sangyo University

[1] はじめに

本稿は言語理解過程に必要とされる概念記憶処理の実現法について考察する。

たとえば、

Mary said she had killed herself. (1)
という文章を受けたとき、どう応答すればこれを理解したことになるのか。

この文章の内容は矛盾している。Maryが自殺したのは、そのことを自から報告している以前のことである。したがって死んだ状態でSayという動作を起していることになる。これは不可能だ。この文章を理解したと言えよにはこのような判断が成立しなければならぬ。これが成立するには、どんな知識をまもってもっていなければならないか、その知識はどのように表現され、どのように利用されるのか。このような問題に対して、R.C. SchankやC. Rieger等の貢献は大きい。彼等は知識の結合原理を探究し、それを利用して意味推論の機構を実現しようと試みた。本稿では、彼等の実現方法から今日支配的な手法をよみとり、それを批判し、新しい実現方法の提案を行う。引き続き2章では、R.C. Schank等の実現手続であり、昨今の支配的な傾向でもある手続き依存型の批判を行ない、データ依存型の推論機構実現の理論的背景として活性化意味ネットワークモデルを紹介する。3章ではこのモデルにしたがって意味推論の実現を実例によって示し、活性化意味ネットワークの制御言語によるプログ

ラム例を示す。4章で、そのような制御言語を実現する機構の設計論に入る。大規模分布論理記憶に関する考察、新しい連想プロセッサNOAHの紹介、それを採用した概念記憶装置のシステム設計へと続く。最後に5章で、言語やハードウェアシステムについて考察をくわえる。活性化意味ネットワークモデルの背景、活性化意味ネットワークの制御言語と従来のグラフ処理言語との比較、使用コンピュータシステムとNOAH概念記憶システムとの比較などを述べる。

[2] 手続き依存型推論批判

言語理解に関する研究は、R.C. SchankやC. Rieger等の研究にみられる如く、まず概念の依存関係(Conceptual Dependency)¹⁾をはじめとして、より高度の知識の結合原理(Connectivity of Knowledge)²⁾を探究し、その成果を利用して理解が成立するために必要な、概念の深層における推論機構を再構成する。

Conceptual Dependency
Verb. Template
Normal Function Template
Connectivity of Knowledge
Causal Chain
Script, Plan, Goal

結合形態の知識は本来データ依存であり、事実の集積であって何か公理的な規則に集約されるという性質のものではない。しかるに、

彼等は概念推論を実現するために、知識の結合形態を手続き型知識に翻案し、プロダクションとして実現する。いつまでも未定であることもよくなくされるこの種のシステム開発において、個々のプロダクションの独立性は歓迎されるべき性質であるが、この利点とはうらはらに結合形態の知識の利点である知識内の広範囲の依存関係を松減し、その結果、システムの実行に際し、プロダクションの番番探索に頼らざるをえなくなった。

言語理解のために概念記憶の上で行われる推論を意味推論という。意味推論の実現に、RiegerはInferential Module³⁾と称するプロダクションをLISP言語PROG形式M式で表現している。現行のコムピュータシステムとLISP型言語、手続き型知識への依存という三者の関係図式は、今日支配的な勢力であり、この実現法が必然の帰結であることは認めざるを得ないのであるが、しかし、本来データ依存型の知識を手続き型知識に翻案することなく利用する方法はないのだろうか。それにはデータモデータとして、データの探索過程によって推論を実現できなくてはならない。そのためには上記の実現法や三者関係図式から根本的に離反した手段を探究する以外に方法はない。

本稿はデータ依存型推論の実現のために提唱された活性化意味ネットワークモデル⁴⁾を紹介し、連想プロセッサによるその実現法について議論する。意味ネットワークは人間の連想機能も模倣する知識表現方法⁵⁾と言われてきた。このモデルはこのことを形式的に追究することからはいった。活性化された意味ネットワーク(Activated Semantic Network, 以後ASNと略称する)は連想という動きのある現象をみずから上に表現できる。ASNの節とやがは、原子概念ともいわれるデータを格納する記録媒体であると同時に、小規模な処理装置の働きをもっている。このようなASNモデルは明らかに連想プロセッサによって実現するのが最もふさわしい。

[3] 意味ネットワークの活性化動態による意味推論の実現

3-1 实例、Enabling Inference

励起動態には否定と肯定の二つの様態があることから説明をはじめる。否定・肯定は概念素項の調整子に表現される。活性化では符号的安定化項に、蓄積項では調整子に現われる。

Mary must not practice any action,
if she is dead. (2)

この文章(2)で表現される事実は、

$\neg \{ \text{DEAD}(\text{MARY}) \rightarrow \text{ACT}(\text{MARY}) \}$ (3)

$\text{DEAD}(\text{MARY}) \rightarrow \neg \text{ACT}(\text{MARY})$ (4)

(3)か(4)の二通りに表現できる。前者では、ある時MARYが死んでいる事実から、その時何かの行為を示すことが推論されえないことが表現されている。後者は、ある時MARYが死んでいることからそのときMARYはいかなる行為を示しえないことが推論されることが表現されている。CDグラフ¹⁾ではそれぞれ、

MARY1 $\rightleftarrows$ HEALTH(-10)
 ↑
 ENABLE (5)

MARY1 $\rightleftarrows$ ACT1
 ↑
 ENABLE (6)
MARY1 $\rightleftarrows$ ACT1

と表わされる。スラッシュ / は調整子における否定とCDグラフ表現したものである。

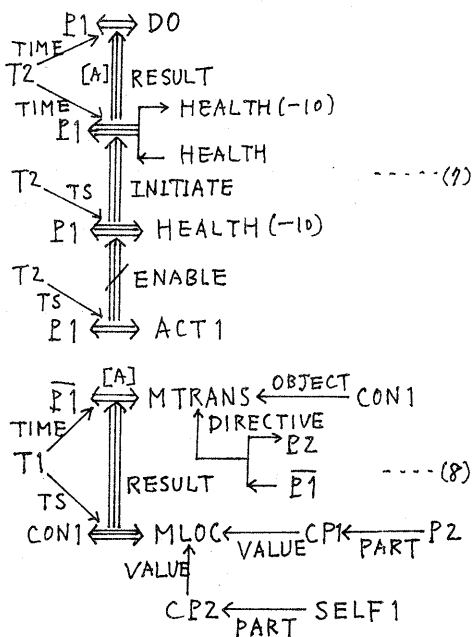
(5)、(6)の両表現においてMARY1 $\rightleftarrows$ ACT1からENABLEリンクを通じて励起状態の移送が起これ、(5)の場合はENABLEリンクが否定されている事を経由した結果、MARY1 $\rightleftarrows$ HEALTH(-10)は否定の符号的安定化項を附加されて活性化された。

このことを否定の活性化(Negative Activation)という。一方、(6)の場合は、いままで特別に意識せずに使ってきた肯定の励起状態の移送によって、はじめから否定の調整子を有するCON, MARY1 $\rightleftarrows$ HEALTH(-10)が励起される。

このことを肯定の活性化(Affirmative Activation)という。1章で示した文章(1)の理解過程を実現する意味ネットワーク(以後、SNと略記する)の活性化動態を例示する際には、上記の否定の活性化を利用する。現実には全く内包的にしか意味のなる否定的情報はたゞ主張的連鎖の否定を経由して、一時的に否定の活性化が起ったCONによって表現される。

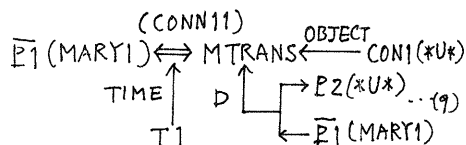
CDグラフ表現はSN表現に変換可能である。
 そのことを保障として、この章ではCDグラフ表現
 によって説明する。

SNの、例(1)に係属する部位を次に示す。



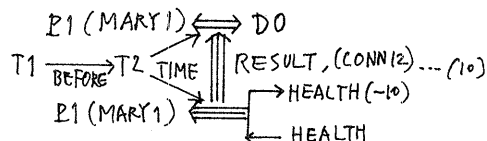
SNの節点や弧には原子概念が添付されている。
 原子概念は活性化されており、小規模の処理装
 置とみなしてよい。そしてSingle-Instruction-
 Stream-Multiple-Data-Streamのモードで
 並列制御される。したがって、以下、ASNの活
 性化動態の説明に当り、実行される操作は、
 すべてSN全体に掛っていることを忘れてはな
 さない。

1. 入力文中、saidに融発されて、(8)の前件[A]が励起される。
2. 入力情報による励起された部位の特殊化が行われる。結果は次のようになる。



3. 活性化域のXU*, すなわち入力情報のみで
 は特殊化を失敗したものに代り、すでに内部
 化されている旧知識による同定が行われる。
 今の例では、CON1もP2も同定できない。

4. 続いて副文中のhad killed herselfに融発されて、(9)の[A]の部位が励起される。
5. 入力情報にもとずいて特殊化が行われた結果は次のようになる。



6. 旧知識による同定はこの場合は必ずない。
7. 本義文論的関係 Say (that) ~ から、(9)のCON1 (XU*)は(10)に同定される。同定の結果は(9)において、

CON1 (XU*) → CON1 (CONN12)

のようにCON1へ絶対同定はCONN12の繋り入れによって表現される。

(注1) (9)や(10)のように励起され特殊化された概念化構造、複合概念化構造には、(必要ならば)新しく代表部点を設け、絶対同定を添付する。

(注2) CON1やCONN11は記述と同定を併せて表現されるものの便宜的な表現法であり、CON1は相対同定子、CONN11は絶対同定子を表わす。

CON1 ← CONCEPTUALIZATION, con1

CONN11 ← CONCEPTUALIZATION, CON11

8. 以上、入力文の内容に該当するSNの部位が特殊化された後は、1 pitch resultative inference へ赴く。この場合、主文、副文それぞれに対して実行される。まず主文に対して、

- ① 励起痕跡#1を残している(9)の部位を励起状態に戻し、マージンを実行する。
- ② CAUSEリンクの矢印の逆方向に励起状態の移行を行う。

(注3) CAUSEリンクとはRESULT, INITIATE, ENABLEなどすべての因果関係連鎖をいう。

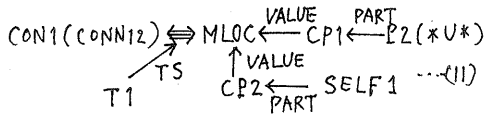
- ③ 再び励起痕跡#1の範囲を照し、これを基底のCON2群に分解し、それぞれのCONの同定子を除く構造によってSN全体に照合し、それと同じ構造をもつ不整合なCONを励起する。
- ④ 新しく励起されたCONに、#1痕跡域の活性化を移植する。
- ⑤ CAUSEリンクの矢印の逆方向に励起状態の

移送を行う。

④ 代表節点の活性化項を移植する。

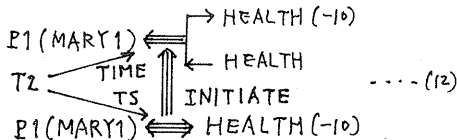
⑦ 以前の#1を消し、新しい励起域に#1を張して励起状態を解消する。

③④⑦の結果はこの例の範囲では取り扱わないうとしてこの結果は、(8)を参照して次のようになる。



(注) ここで SELF1 の意味することは、語られた内容 CON1 と他の誰か P2 と共に聞いた MEMORY 自身が語りかけられた一人として自身を自覚していることを意味している。

9. 続いて副文について pitch resultative inference が実行される。結果は、(7)から



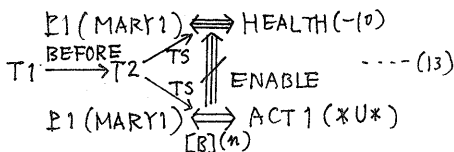
以上までのところで、各種項の範囲では不連続だった(7)と(8)が、活性化、

MARY1 (P1 ↔ P1)
CONN12 (CON1 ↔ (10))

を仲介にして結合された。

10. この後、理解したことの評価として Enabling inference が起る。語りかけられた文章の内容を一応認識したあと、それが旧知識に照して承服できるものであるか判断する。この時点で(11)と(12)が励起痕跡#1も張っているが、これらを出発点にし、ENABLE-リンクを経由して resultative inference を展開する。その各 pitch が起ることに、拡大した活性化域内部で知識間の矛盾が存在しないかどうか検査する。これを ENABLE リンクが連続しなくなるまで実行する。

① ENABLE-リンクのみによる pitch resultative inference の実行。結果は、(7)による



が励起されている。この ENABLE リンクは

否定の調整がなされているから、(13)の条件 [B] は否定的に活性化されている。

(Negative Activation)

② [B] は不特定の ACTION act1 を含み、その否定を結論しているから、このことから評価手続きが触発される。すなわち SN の活性化域において、MARY1 を ACTOR とする否定されない ACTION が存在するかどうか確かめる。時間情報への制約も考慮する。存在すれば、矛盾の存在を表明し、Enabling Inference を起す。存在しなければ、①へ戻り、繰り返す。②の評価手続きは、他にもあるが、ここでは言及するのを省略する。

より詳細な手続きを示せば次の通りである。

① (MARY1) ↔ ACTION
↑ TS
T2

によって、SNの活性化域を照合して、励起する。この例では(13)が該当する。

② (9)の時間情報が(14)を満たすかどうか調べる。T2 は BEFORE T1 であるから、満たしている。したがって否定されない ACTION が存在した。

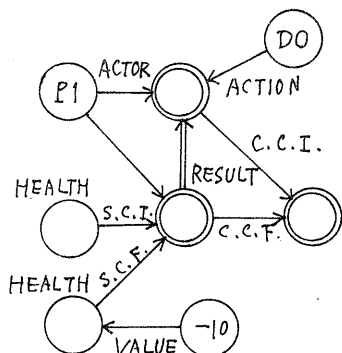
以上、本節の事例によって ASN モデルのいう SN の活性化動態とはいかなるものか例示できたと考える。

3-2. ASN 制御言語によるプログラム例

言語表現は SN の中にそれに該当する内部表現をもっている。前節の例の中に出てきた Kill oneself も(7)の[A]のような内部表現をもっている。これを SN 表現に変換して示せば、図1のようになる。

入力文に触発されてこの部位が励起され、特定化される過程を例にとって ASN 制御言語⁹⁾によるプログラム例を示す。

```
MATCH N.EQT.<DC:DO> 1
TRANSFER STATE(1).EQT.<DC:+ACTION> 2
STORE N<DC:CON.,ID:CON01> 3
SET LOGICAL MODE AND,N 4
TRANSFER STATE(2).EQT.<DC:-RESULT> 5
CLEAR EXCITATION NA 6
MATCH N.EQT.<DC:HEALTH> 7
TRANSFER STATE(2).EQT.<DC:+S.C.I.> 8
CLEAR EXCITATION NA 9
```



S.C.I. = State Change Initial
S.C.F. = State Change Final
C.C.I. = Causal Chain Initial
C.C.F. = Causal Chain Final

図1. CDプログラム(?)の[A]のSN表現

プログラム続き.

```

MATCH N.EQT.<FZ:-10> 10
TRANSFER STATE.EQT.<DC:+VALUE> 11
STORE N<SS:i> 12
CLEAR EXCITATION 13
MATCH N.EQT.<SS:i,DC:HEALTH> 14
TRANSFER STATE(2).EQT.<DC:+S.C.F.> 15
CLEAR EXCITATION NA 16
MATCH N.EQT.<#1> 17
PUT N<#1> 18
TRANSFER STATE(1).EQT.<DC:+C.C.I.> 19
CLEAR EXCITATION NA 20
MATCH N.EQT.<#2> 21
PUT N<#2> 22
STORE N<DC:CON.,ID:CON02> 23
TRANSFER STATE(1).EQT.<DC:+C.C.F.> 24
CLEAR EXCITATION NA 25
MATCH N.EQT.<#1> 26
PUT N<#1> 27
STORE N<DC:CON.,ID:CON12> 28
CLEAR EXCITATION 29
MATCH N.EQT.<ID:CON01> 30
TRANSFER STATE.EQT.<DC:-ACTOR> 31
STORE N<SS:MARY1> 32
CLEAR EXCITATION NA 33
CLEAR LOGICAL MODE 34
SET SHARING EXCITATION.EQT. 35
<DC:-STRUCTURAL LINK>

```

MATCH (1) N.EQT.<ID:CON12> 36
CLEAR EXCITATION N 37

このプログラムによつて、図1のSNが、P1がE1(MARY1)になり、3つのCONがほぼ同定子CON01, CON02, CON12を与えられて安定化される。さらに、全体の節点が節点起痕跡井上を残している。

[4] ASNモデルの連想メッセージによる実現

4-1. ASNモデルの概要⁸⁾

図2にASNモデルの図式表示を与える。意味ネットワーク記憶には活性化されたSNが蓄えられる。ASNの節点や弧はSIMD並列に制御される。この制御を行うのが賦活部である。SNの活性化状態は触発因子によって起動される。賦活部がその触発因子を認知する。触発因子には、入力文からくるものと、SNの活性化状態からくるものがある。前者は文分析部から賦活部へ、後者はスタック2を通じて意味ネットワークの照合操作によって通知される。SN記憶部の制御はSIMD並列の照合操作が基本となるので、図では、照合子の経路を明確に示している。

4-2. 大規模分布論理記憶に関する考察⁶⁾

連合形態の知識の蓄積とその有効利用のための新しいアーキテクチャを考察しようとすれば、

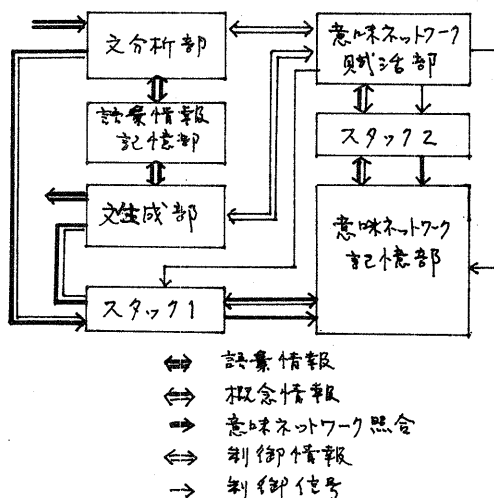


図2. ASNモデルの図式表示

きわめて広範囲の諸分野を総合的に考察しなされるべきである。人工知能、ベクトルベースおよびコンピュータアーキテクチャにある。すなわちデータモデルや推論機構の諸モデルを評価するには、それと実現するアーキテクチャの検討も含まれるべきなのである。

知識を有効に利用するには必要な知識とすばやくとりだせることが肝要である。探索過程における予読き数が急激に増えることを防がなければならない。(Combinatorial Explosion)

このために並列プロセッサや連想プロセッサの採用が考えられる。それらの紹介論文はすでに立派なものがある。^{10) 11)}

並列プロセッサや連想プロセッサのもっとも直感的な分類基準は、セルすなわち単位処理装置の規模とそれらの相互結合形式をみることである。セルの規模が大きいほど、それらの間のプロトコルが複雑になり、相互結合形式は複雑なデータ構造を配置しやすく、かつたどりやすい構成ほどよい。これら二点を考慮して、連想プロセッサ、とくに分布論理記憶(以下、DLMと略記する)を基本形として採用した。

データの複雑さと量とが大きくなって、それに対処できるメモリシステムへの要望は日甚しに強くなる。その要望に応えるために加えられてきた改良技術の基本思想は、メモリシステムにおける論理の分散であった。キャッシュメモリや仮想記憶は、データの論理構造を物理構造に配置する予読きをプログラムから解放してシステムの仕事にした。DLMの提案は探索過程をキャラクターやアイテムのレベルまで並列操作にするため、メモリに論理操作の開始を分散した。さらに神経のモデルであるしきり素子回路網もこの一貫した視点の下に置くことができた。もっと細かい記憶単位のレベルまで論理が分散されたシステムとみることもできる。このような視点の設定(視座)が正しいければ、これを通してみられる一貫した技術発展の系譜の中で、すでに実現している仮想記憶等の技術から連続して実現可能なものはDLMである。

連想プロセッサの基本的性質

- 1) Content Addressability
- 2) Single Instruction Stream Multiple Data Stream (SIMD)

1)は内容によって目的のデータにアクセスできる性質である。2)は並列プロセッサのある分類基準にもとづく性格付けであり、一つの命令で並列に多くの処理装置が、それぞれに割り当てられたデータを処理できる性質を表わす。SIMD構機には連想プロセッサのほかにArray Processorがある。これは基底によるアクセス法(location addressable)を採用している。

こゝでは連想プロセッサとしてDistributed Logic Associative Processorを考へる。これは当初、キャラクター単位の処理を目的とするものであった。(Character-Oriented) (しかしこれから発展させたシステムはかならずしも処理情報の単位をキャラクターとしないので、1セルに格納されるデータを、たいていセルデータとよぶことにする。セルの実行する命令の内容は次の通りである。

- ① 状態の変更。
- ② 状態やデータの隣接セルへの転送。
- ③ 入力バスからデータを受け入れ。
- ④ 出力バスへのデータの送り出し。

こゝで状態は、2状態に限る。セル状態素子で表わされる、activeかquiescentかというか一つの状態である。

連想プロセッサを採用する理由は、データの探索過程に生ずるCombinatorial Explosionの解消にある。まずContent Addressabilityがその一端を担う。さらにアドレスレック機能を強化し、複雑なデータ構造をたどりやすくするために、セル間の相互結合形式が重要な働きをする。こゝでは、各データモデルのもつ基本的なデータ構造を物理構造に配置する問題がかわるので、いかなる相互結合形式がよいかはデータモデルとの関連の下に考えなければならぬ。たとえば、リレーショナルデータモデルでは、基本的に関係集合の構造が求められるだけだから、線形配列でよい。より複雑な結合構造を直接物理的システムに写像しようとするのは、セルの2、3次元配列と多様な相互結合形式が要求されるべきでない。

DLM基本原理の生かし方

DLMの基本原理を生かして開発された連想プロセッサに二つの大きな支流が識別される。それぞれはDLMの二つの基本的性質のうち、一方をより強調して発展させたものである。

- 1) Circulating Associative Processor

第1群はContent Addressabilityを強く意識している。当初提案されたDLMは完全並列マツチング方式であったが、それから派生した各システムでは実現可能性を高めるために、その並列性を低減する傾向をたどった。とくに大量の循環メモリの使用によってblock-oriented bit-parallel word-serialで照合と実行するシステムが多い。この種の機械の数理モデルは、線形格納方式をもたず循環テープ状の蓄積媒体をもつキューリングマシンの並置システムである。

並列探索性は並置されたプロセッサの数だけ実現される。応用面からみれば、蓄積すべき知識の大量性に対応する連想プロセッサシステムであり、とくにデータの線形格納を基本とするリレーショナルデータベースモデルを採用したデータベースの管理に適切である。この思想に基づいた開発例は多く、わが国でもよく紹介された。¹²⁾こゝでは1例として、B. ParhamiのRAPID, Rotating Associative Processor for Information Disseminationの論文を紹介しておく。¹³⁾名称の中にRotatingという表現をとりいれているのが、よく体を表わしている。

2) Structure-Oriented Associative Processor

第2群はセル内相互結合形式の柔軟性に注目した。セル内結合路の制御もSIMD方式の下にある。こゝではデータ論理構造をたどる過程をセル内状態転送に写像しようとする。(hardware link) 応用面からみれば、蓄積データ構造の複雑性に対応するものといえる。この種の機械の数理モデルは状態出力オートマタのネットワークモデル¹⁴⁾である。(但しJ.R. Fikselのモデルは多状態出力オートマタのネットワークである。2状態出力オートマタが、この種の機械の正しい数理モデルである。)

セル内の励起伝播(Active-state transfer)によってデータの論理構造をたどる。励起伝播はネットワークに存在するすべての命令について並列に実行されるから、この意味での並列探索性によってCombinatorial Explosionにおおむね対応する。相互結合路にデータを伝播する方法もある。この場合は、番地アクセス方式を採用し、アドレス情報を伝播することにより、複数の命令に対して並列に各セルの固有アドレスと伝播アドレスとの照合と実行しながら論理構造をたどる方式になっている。

第2群の代表例を挙げれば、

① Tree Channel Processor¹⁵⁾

② Association-Storing Processor¹⁶⁾

③ Network-Organized Association Handler⁴⁾

などがある。それらのセル相互結合形式について概略を述べる。

①のTCPはトリ構造とセル配列の中に組織的に写像する。したがってトリ構造で表現されるヒエラルキのあるデータにおいて、部分集合の抽出、上位や下位のアイテムへの信号伝播をいとも実現し、その伝播経路の最小化をはかっている。

②のASPは、phrase-orientedとitem-orientedの2通り提案されている。前者はセルの1次元配列、後者は2次元配列を採用し、いずれも、節点レベル付グラフを格納する。たとえば、2次元配列の場合、アイテムの逐次探索を格納するために、一番低いレベルでは1つのセルに1つの組織構造が格納される。3組織構造の各頂は、それらの内容が格納されていることを示すアドレス情報である。このように3組織構造を基本単位とする限り、ある3組織構造を一般高いレベルの3組織構造の1つの頂となしうるので、階層状に構造化されたデータを格納できる。このセル2次元配列では、上方および右方への伝播路があり、これを通じてアドレス情報を伝播しながら、データ構造を並列にたどる。ASPは3組織構造、すなわち2次元関係を基本単位とするデータ表現の専用プロセッサである。

③のNOAHは、同じくセルの2次元配列を採用するが、従来の常識を脱して、個々のセルからみた隣接セルの数を一帯にふやした。相互結合路に流れるのは励起状態情報のみである。状態を論理構造と物理構造に直接写像できるので、節点レベル付有向グラフや有向ハイパーグラフなどが格納でき、かつ並列にたどることができる。(parallel tracing)

これから第2群の基本思想は、構造の複雑さに対応する新しいデータプロセッサの提唱として注目すべきである。格納されたデータの各アイテムの直接的な励起、非励起の集合体バタンの動態が、忘却の想起や消滅を実現しているという新しいアーキテクチャは、データ依存型推論機構のハードウェア実現のために、きわめて明確な方向付けをもつ構想として注目される。

データプロセッサの実現性を高めるために、

上記の第1段階の考え方が第2段階の考え方の複合システムの開発かゝりました。

4-3. 典型的な励起動態

2つの再帰的拡張節点¹³⁾があり、その一つから他方へ励起状態が移送される。この基本的な励起動態は形態論的に2通りに分類できることをすでに示した。¹²⁾ これらの基本動態によって構成される典型的な励起動態について考察する。

1) 進行 Carrying on

これは上記の基本動態をいくつも連続していく場合である。最初は基底節点すなわち原子概念から始まってもよいし、概念化構造からでもよい。しかし途中は道程の構造化要素を固定しながら進行する。

2) 連結 Connecting up

これは M. R. Quilliam¹³⁾, K. Colby,¹⁷⁾ J. R. Fiske¹⁸⁾ 等の一連の研究に受けてみられてきた話題である。すなわち2つの原子概念の間の関係を探求して、その間の関係をいくつかの文章で表現する。語彙の三題断ならぬ二題断の過程である。より大きな概念化構造同志を連結する意味で Knitting という表現もあてはめられる。

二題断の過程を分析すれば、次の2つの段階が識別される。

第1段階：基底節点から成る一筋の結合路をみ出す。

第2段階：上記の結合路を案内として、更めて途中の概念化構造を固定しながら、Carrying on を実行する。

フランクレーを伴った創造的技法として、この二題断の過程を積極的に使うことを提唱している人もいる。¹⁷⁾ 上記の第2段階は Carrying on とかわらない。第1段階のみを区別して、交差 Crossing とよぶことにする。交差アルゴリズムの多状態出力記憶オマトタネットワークモデルによる表現が、J. R. Fiske¹⁸⁾ によって与えられている。¹⁴⁾ 交差の手段として、励起域の垂作為な展開である拡散 Scattering と、選択的展開 expansion とが考えられる。

4-4. NOAH Double Cellars Construction

による ASN システムの実現

CD 理論¹⁾では記憶機構そのもののモデルと

して、

Conscious Processor	CP
Intermediate Memory	IM
Long Term Memory	LTM

などの Mental Location を区別している。

これを ASN 理論で解釈しようとするのは、これは SN の励起され、活性化された領域において識別され、CP は活性化域でしかも励起されている領域、IM は活性化域だが、いま励起されていない領域、そして LTM は活性化されていない領域、したがって分論、励起もされていない領域である。

このように記号化機構を ASN システムでは、

NOAH Double Cellars Construction NOAH-DCC

固定ハードディスク記憶装置 MD

によって実現する。図3を参照せよ。

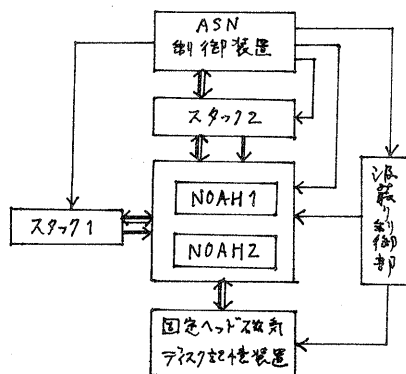


図3. NOAH-ASN システム

NOAH-DCC において SN の活性化動態が制御されるので、この部分が CD 理論の CP や IM の働きをする。MD では励起状態を示すことができないから、CP の働きはできない。MD は IM と LTM の働きをする。

さて次に、概念情報処理を行うのにいかなる記憶操作が必要で、そのために NOAH-DCC や MD をどのように制御すればよいか述べる。

(1) 概念データの転送 mental transfer

(1-1) MD → NOAH, 想起 recall

MD から NOAH へデータ転送するときには NOAH 1 か 2 かいずれか一方あるいは両方を選択できる。両方へ同じ内容と同時に送ることはできても、同時に異なる内容を送ることはできない。

(1-2) NOAH → MD, 記録 memorize

NOAH 1 か 2 かいずれか一方から MD へ、その内容を送る。

(1-3) STACK → NOAH 短期記憶 short-term memory

STACK から NOAH へデータ転送するとき、送り先を NOAH 1 か 2 かのいずれか一方あるいは両方を選択できる。STACK 内は LISP 言語 S 式で表現されている。たとえば、一つの概念化構造を NOAH へ格納するとき、ルーチンプログラムにしたがって、節点や弧の原子概念が STORE, LINK 命令によって蓄えられていく。(MACROSTORE)

(1-4) NOAH → STACK 発話 utterance

送り元は NOAH 1 と 2、いずれからでもよい。

(MACROREAD)

ASN 制御言語によるルーチンプログラムを MACRO 命令とよぶ。ASN 制御言語の照合形式^(*)の内容はスタックの内容でも指定できる。

その場合は、照合形式の概念素項の位置に \$1 あるいは \$2 と記入するものとする。

(2) 照合 matching

STACK の内容によって、NOAH に格納されている SN へ照合する。構造化要素単位の照合ルーチンプログラムを作成する。(MACROMATCH)

(3) 併合 merging

NOAH 1 と 2 とにそれぞれ格納され、共に構造化され、共通する同定子をもち、異なるとすべき相方をたとえば NOAH 2 において併合する。STACK 1 に NOAH 1 からデータ転送し、STACK 2 には NOAH 2 に対する照合子を格納し、これに NOAH 2 上の SN を同定しながら、STACK 1 から NOAH 2 へデータ転送する。

(MACROMERGE)

(4) 広域交差 great sphere crossing

2つの概念を連結する路を探し手続させて、2つの位置が一つの NOAH では含みきれない位離れている場合の交差アルゴリズム^(*)の実現法である。拡張といってもある程度選択的な展開によって、NOAH ↔ MD のデータ転送と照合を繰り返しながら路をたどる。この場合、2つの概念のそれぞれからの路に対して NOAH 1 と 2 とを割り当てる。NOAH → MD 転送のとき筋起痕跡を残し、MD 上で両方の痕跡が遭遇するようにする。勿論、それを見出すのは NOAH である。(MACROCROSS)

4-5、汲み取り制御命令^(*)

NOAH においても、MD においても、セル節点面は境界のない球面を形成している。NOAH せ

るレスラムの方だけ、格納 SN データの境界線が存在するかそれは浮動している。

NOAH と MD との間でデータやりとりするときの状況は、MD 節点球面上に NOAH 節点球面が接しなごころがり、両者がやりとりする SN データは、NOAH 球がある地点で停止し、そこで NOAH 球面を切り開いて MD 球面上に展開して重なる部分が互れに相当する。したがって NOAH と MD との間でのデータ転送に必要な制御命令は次の通りである。

(1) ROLL u, v

これは NOAH 球の位置きめ命令である。

位置きめは常に現状から相対的に行なわれる。

u, v は +, - の値をとり、u は右方へ、v は上方へ、いくつかのデータポイント(1セル分)移動するかを指定する。

(2) MEMOLIZE t

これは、現在地点の対応関係で、NOAH の内容を MD へ書き込むことを指示する。t は NOAH 1 か 2 かの二者択一の選択を示す。NOAH の内容は変化しないか、MD のその位置で以前に格納されていた情報は書き換えられる。

(3) RECALL t

これは、現在地点の対応関係で、MD の内容を NOAH へ汲み取ることを指示する。追加された方向に新しく汲み取った節点面があるのと等量、逆方向に NOAH 球面から消えていった節点面がある。

4-6 多重整合処理

多重整合処理(Multiple Match Resolver, MMR)の機能は

(1) 励起セルの数をかぞへる。

(2) 励起セルの位置を同定する。とくに残る一次元的な位置きめにしたがって、一審若い審若の励起セルの位置を同定できる。

この MMR の機能を利用して 2つの制御命令が得られる。

(1) BRANCH(A).y.S

S は正整数等、y は大小関係を表す。この命令は励起セルの数を MMR から参照し、それと S とを比較し、条件を満たせば基地 A の命令へ分岐し、満たされれば次の命令を実行する分岐命令である。

(2) PREFERENCE x

MMRの機能(4))を利用して、励起セル群から、1セル分だけ優先読出しを行う。

4-7 自動割り出し機能

- (1) セルへの励データ格納に際し、指定の励起セルの励格納場所(AS)が使い尽されているとき、自動的に処置される。すなわち空明隣接セルとさかし、現在の励起セルと励起共有設定しなうち、目的の励データ格納も実行する。
- (2) NOAH上の励起動態のさ中に、読み取り節点間の境界に励起域が及んだとき、その方向にROLL, RECALLが自動的に起り、励起域の展開に降着が起るのを防ぐ。

以上 本稿の記述と文献(4), (5)を参照すれば、ASNシステムの全貌が明らかにする。

[5] おわりに

- (1) ASN理論は、生成規則や書き換え規則によるプログラミングの実現法に批判的である。生成理論より連合理論へのかたよりを示す。
- (2) ASN制御言語は、従来のグラフ処理言語が型合わせと書き換えを基本的機能とする單一形式の命令系であるのに比べ、ネットワークの活性化動態という新しい概念に基づいたグラフ処理言語である。
- (3) ASN理論は形態論志向の意味推論表現体系である。
- (4) ASNモデルのためのオートマトン理論の背景、ASNモデルはSIMDテンタ・ヘースマンを想定している。J. R. Fikselのオートマ・ネットワークモデルが忘却性の多状態出力オートマトンのネットワークであるのにくらべ、ASNモデルは記憶をもつ二状態出力オートマトンで表現される。オートマトン理論の課題としたい。
- (5) 現行コンピュータシステムの問題点。

意味推論を現行システムで実現するとき、蓄積データの膨大化が問題である。原子概念、あるいは概念化構造物の水準の連合形態知識を格納するとき、見出しに対する概念データの中心化によって同じデータが多重に格納されることになる。手続記憶型の実現法のために、同じ知識が宣言的と手続的との二重の形態で同時に格納されることも生ずる。また breadth before depth の推論過程をシリアルシミュレーション

することによる探索時間の増大もシステムを非現実的にする。ASNシステムはこれらの問題を解消する。ASN制御言語は文献(4)のNOAHのマイクロプログラムレベルの言語から、矢張り1レベルプログラムで実現可能である。

(6) ASNシステムのNOAHとMDによる実現は、心理学的現象である制限された注意領域のワーメモリになっている。

(7) 本稿のはじめに述べた実現後の関係図式、現行コンピュータシステム⇔LISP型言語⇔手続記憶型推論(書き換え規則)と対比し、新しい使用図式連想プロセス(テンタ・ヘースマン)⇔ASN制御言語⇔データ依存型推論(活性化動態)と対比する。

参考文献

- 1) R. C. Schank, Conceptual Information Processing, North-Holland, 1975
- 2) R. Schank and R. Abelson, Scripts, Plans, Goals and Understanding, L.E.A., 1977
- 3) C. J. Rieger III, Conceptual Memory, Stanford University, STAN-CS-74-419, 1974
- 4) 嶋田, テンタ・ヘースマングラフを基盤とした二重連想プロセスの設計, EC76-24, 1976
- 5) —, NOAHシステムによるセマンティックネットワークファイル管理, EC76-36, 1976
- 6) —, 連想プロセスとセマンティック, 「80年代のソフトウェア」より, 日本エレクトロニクス, 1978
- 7) —, 意味ネットワークの活性化することについて, AL79-68, 1979
- 8) —, 活性化された意味ネットワークの理論, AL80, (投稿中) 1980
- 9) —, テンタ依存型推論, AL80, (投稿中) 1980
- 10) K. J. Thurber, L. D. Wald, acm Com. Sur. 2(4) '75
- 11) S. S. Yan, H. S. Fung, acm Com. Sur. 2(1) '77
- 12) 関野陽, 根拠発表, 情報処理12(10) '76
- 13) B. Parhami, Design Technique for Associative Memories and Processor, Ph.D. UCLA, '73
- 14) J. R. Fiksel, A Network-of-Automata Model of Question-Answering in Semantic Memory, Ph.D. Stanford University, 1973
- 15) G. J. Lipovski, The Architecture of a Large Distributed Logic Associative Memory, AD692195, 1969
- 16) D. A. Savitt, H. H. Love, R. E. Troop, Association Storing Processor, AD818529, 30, 1967
- 17) シン・カグリ, フラングーの文法, 読書書房(録) '78
- 18) M. R. Quillian, Behav. Sci. 12 1967
- 19) K. Colby, L. Tesler and H. Enea, Math. Biosci. Vol 2