

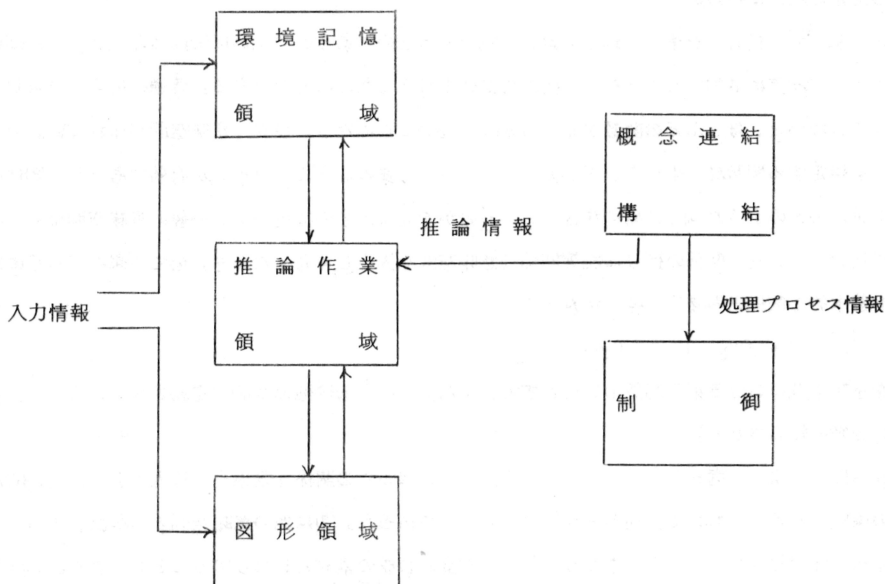
A 1. 初等数学における概念の連結構造とその応用

長沢 勲 (九州大学中央計数施設)

1. まえがき

近年人工知脳の研究は、種々のアイディアに基いて盛んに行われるようになってきた。これらの中でも(自然)言語と論理とを用いてシステムを構成する方法は、情報工学的応用の観点からも特に興味のある方法である。しかしながら、この方法を用いて、人間の持つ複雑で莫大な知識を体系的に整理する試みは、その重要性にもかかわらず、あまり行われていないように思える。ここではこの問題を一般に論ずるのは困難であるから、初等数学の分野を研究の対象にとり、この分野の諸概念を概念の連結構造という手段を用いて整理することを試みた。(概念の連結構造とは、我々が持つと考えられる諸概念を相互に規定されたものとして捉え、これを標準化した構文を用いて記述した意味体系である。)初等数学の概念の多くは、定義され、導入された複合概念であるから、思考対象の母集団をうまく規定する方法を採用することにより、かなりの程度構成的に議論を進めることができる。

システムをなるべく簡単にするため、キャラクタと線分からなる簡単な図形の領域をとり、全ての議論の基礎領域として設定する。また、これらの対象間の関係付を行ったり、対象の使用目的を決めたりする原始関数を導入しておく。(これらはシステムの基本機能であり、多くはプログラムによって実現できるものとする。また推論などに利用する場合には真偽決定関数としての意味を持たせる。)意味体系はこれらの原始関数の機能と具体的に与えられた対象を素材として構成する。思考対象の母集団を規定するものを、ここで意味空間とよぶことにすれば、概念の連結構造は、意味空間内部の結合とその近傍空間との結合関係に分けて議論するのが便利である。意味空間は、前述の原始関数またはその総称関数によって、切分られたり、構成されたりするので、意味空間を表現するため、これらに対して大別して4種類の制御属性を導入し、これらを必要に応じて適当に宣言しなおすことにより、近傍空間の生成を行う方法を用いた。



(第1図) 概念連結構造を利用した意味処理システムの構成

2. “見方.”と制御属性

システムの原始関数は本来ある機能であって、システムの中では種々な形で使用され、その役割は固定されていないと考えられる。そこで、意味空間を定義するため大別して次の4種の制御属性を導入する。これらは、原始関数またはその総称関数がある特定の意味空間の中で使用された場合の役割を表わしている。(特定の意味空間からみたとき、同等な機能を与える原始関数が複数あることがあるから、これらのグループを代表する総称関数を考え、原始関数と総称関数を合せて“見方”とよぶことにする。)

- (F0) 思考対象間の関係付を行ったり、対象の構成的定義を与える関数の属性
- (F1) 意味空間の座表軸(互に従属性が強いが)への射影を与える関数の属性
- (F2) 意味空間内部の対象間の関係を定める関数の属性
- (F3) 近傍空間の対象を与える関数の属性

3. 意味空間の構成

意味空間は、ある対象の注目している側面について母集団規定を行うものである。意味空間に対して次の情報を与える。

- (1) 一般に複数組の意図のモデル
- (2) モデルと成りうるための条件
- (3) 意味空間内に張った座表軸と軸への射影関数
- (4) 母集団内部の対象間の関係を決定する関数
- (5) (3),(4)で定義される関数関係と述語表現で与えられる関係の相互定義
- (6) 述語表現で与えられる関係で作られる推論規則
- (7) 近傍空間にある対象を取出す関数
- (8) (3),(4),(7)の関数の相互変換規則

意味空間はモデルに基いて“見方”を集めこれらを素材に構成するものであると考えれば(1)は当然であろう。(2)は意図のモデルが代入によって次第に増加することから、代入の条件を与えるためのものである。普通の場合、思考対象の範囲が充分限定されていないと、ほとんどの関数が無定義領域を持つことになり、役立つ意味空間が作れない。そこで特定の意図のモデルを規定する関数群の中から、不必要なものを落して条件を作るプロセスが有効であろう。(3)は座表軸の集合を与え、モデルのどのような側面に陽に注目しているのかを陽に示すのに用いる。一般に意味空間は多くのモデルを許容する方がよいから、同一関係の代りに相等関係(意味空間を固定して考えたとき、全ての属性が同じになり区別不能になる)を基本的な関係と考え次の様に定義する。

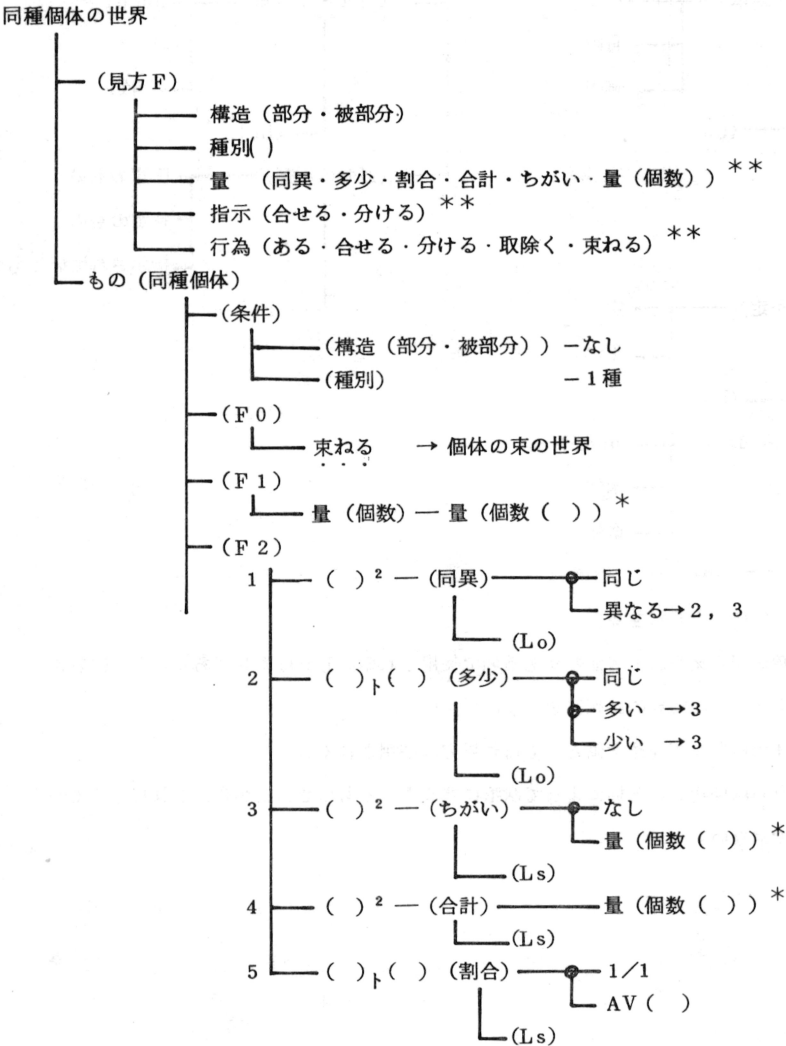
$$(*) \quad x=y \quad \stackrel{Df}{=} \quad (\forall f)_F \quad (f(x)=f(y))$$

但し、Fを意味空間における射影関数の集合とする。また $f(x)=f(y)$ は他の空間で定義されているとし、最終的には原始関数に帰着させる。

(4)は基本的には原始関数によると考える。このような与え方をすれば多くの関係を効率良く導入できる。(5),(6)は論理的推論の世界との橋渡しを考えたもので、述語論理の援用も必要であろう。(7)は他の意味空間との橋渡しであって、これによって他の意味空間の対象をくり込んできたり、新しい対象の合成の素材としたりすることができる。(8)は多くの意味空間を定義したとき派生した補助的なものである。

(図3)は同種個体の世界において使用されている“見方”のリストを示したものである。“見方F”と示されて

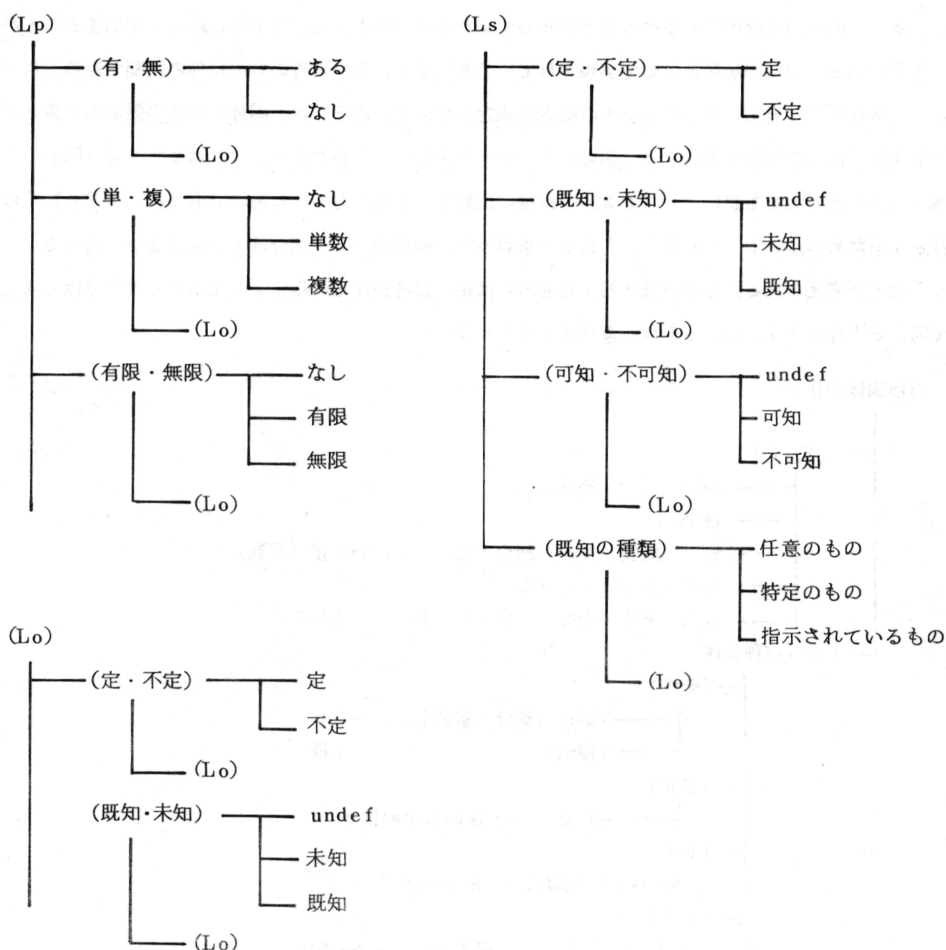
いる部分は常にこの系内で使用されるべき見方を集めて来たものである。従って必要に応じて増加することがある。このうち**印のあるものは原始関数として実現されていると仮定し、それ以外のは他の空間で定義される総称関数とする。“（条件）”は見方を用いてモデルの範囲を規定している。あるいは、同種の個体の領域の定義と考えても良い。“（F1）”はこの空間の場合“量（個数）”という座表だけしか持たない。これはまた“量（同異）”という見方から導入されたと考えても良い。“（F2）”は量の比較を行う見方を与えたものである。“（F0）”には個体のたばを構成するために用いる“束ねる”，“拾うつ束ねる”，個体のグループ分けを変化させる“合せる”，“分ける”，“取除く”などが考えられる。指示（分ける・合せる）は単に思考操作上の指示を与えるもので“分けて考える”，“合せて考える”などとすればもっとはっきりするであろう。



(図 3) 同種個体の世界を表わす意味空間の見方のリスト

** 基本関数で実現される見方

* システムの機能で表現する。例えば pointer の集合または一意なコード

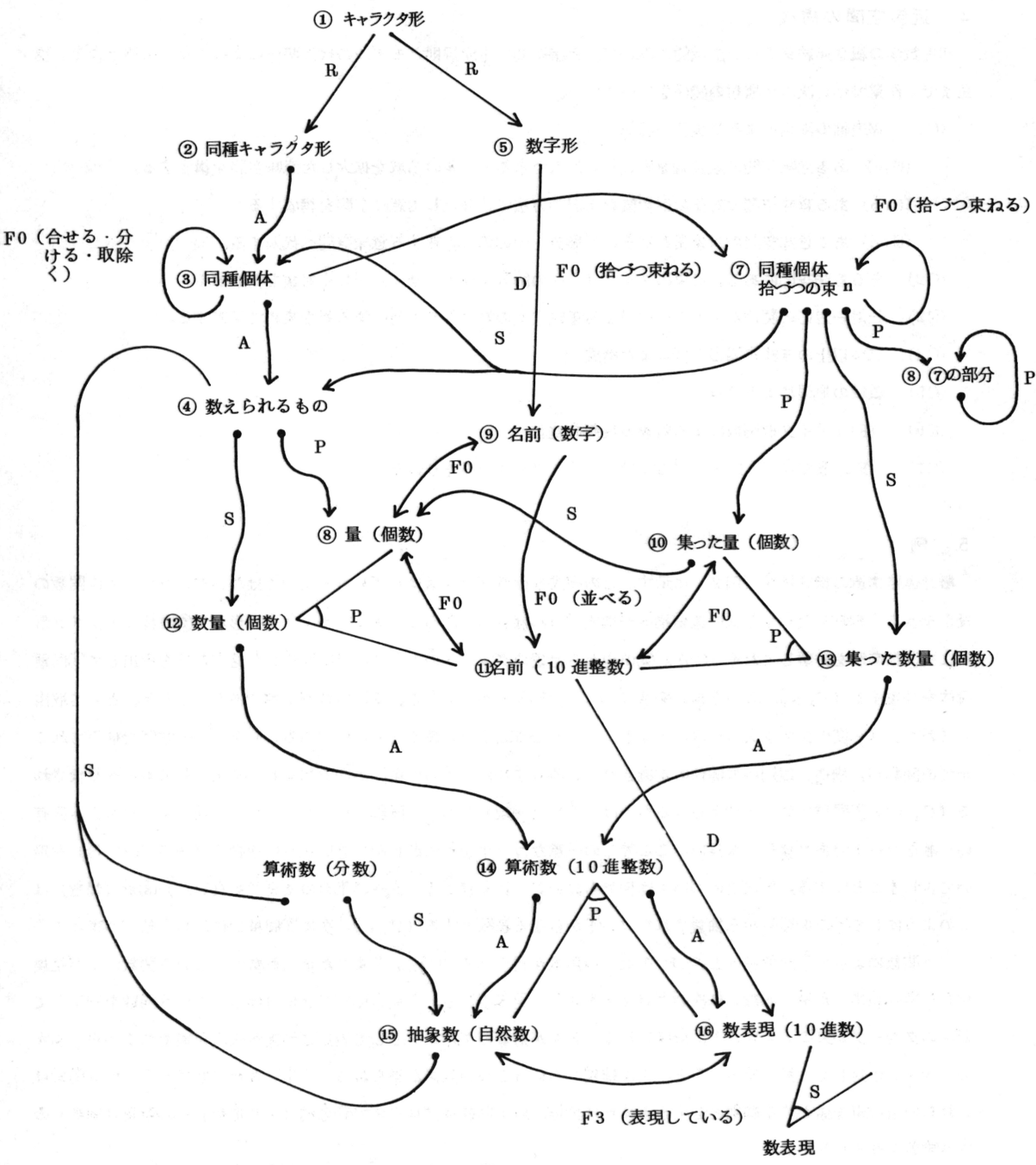


(図4) 論理的様相を表わす見方

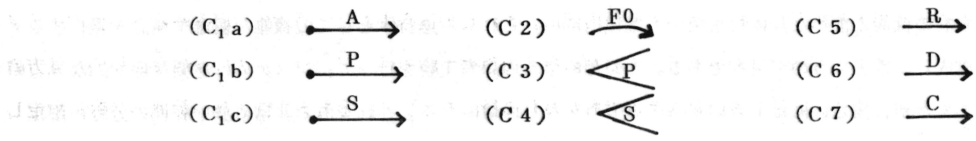
Lp は複数個が与えられる可能性がある場で使用される。Ls は単数であることが知られているときそれらの論理的諸属性を示すのに用いる。

Lo は個体ではなくて述語を値として持つ見方に使用される。

但し、これらは使用される場によって次第に構成されるものであるから、これらだけでつ



(図2) 概念連絡構造の構成例



4. 近傍空間の構成

初等数学の概念連結構造は、広域的にみれば、近傍空間の生成規則とモデルの代入関係によって良く記述できる。現在までの作業では、次の7種類の関係を用いている。

(C1) 座表軸の限定による抽象化の関係

(C_{1a}) ある意味空間の座表軸を限定し、代入できるモデルの領域を拡大した意味空間を構成する。

(C_{1b}) ある意味空間の対象の射影値を上記の方法で抽象化して意味空間を構成する。

(C_{1c}) ある意味空間の対象またはその射影値を他の既に存在する意味空間へ代入する。

(C2) ある意味空間にある、対象にタイプ (F0) の関数を作用させて、新しく対象を構成する。

(C3) 一対一対応の関係がある二つの異なる意味空間の対象の互に対応する対を集めて構成する。

(C4) 二つ以上の意味空間の合併による構成

(C5) 領域の制限による分離

(C6) (F0) タイプの関数による対象の役割の宣言

(C7) (F2) または (F3) タイプの関数の (F1) タイプへのくり込み

5. 例

概念連結構造の構成例を(図2)に示す。この構成を行うプロセスは必ずしも一意的ではないが、システムの関数の役割をうまく制御しながら概念の連結構造が構成できれば良いであろう。まず①キャラクタ形から②同種キャラクタ形および⑥数字形を分離しておく (C5)。②の中から対象のグループの大きさにだけ関係した見方だけを取り出して③同種個体を抽象する (C_{1a})。このとき、構造(部分・被部分)がないこと、種別の数が1種であることを条件として取りしておく。(一度に条件を大巾にゆるめても、モデルの構成が行い難くなるだけである。)次に③の空間で使用される全ての関数が、構造、種別に無関係に定義されているとすれば、さらに条件がゆるめられ④数えられるものへ抽象される (C_{1a})。③同種個体、④数えられるものは、“量(個数)”という関数を持つが、これは値域としてシステム固有の一意的コードであれば何でも良い。ここでこの一意的コードを意味がわかり易いように漢数字(一二三……拾)を用いて表わすことにする。拾以上については何でも良い。(例えばポインターの集合のままでも良い。)⑧量(個数)はこのようにして④の射影値から抽象されたコードからなる領域とする (C_{1c})。次に③同種個体から“拾づつ束ねる”という関数によって⑦同種個体と拾づつの束^{II}の領域が得られる (C2)。“集った量(個数)”という関数および抽象化した領域⑩集った量(個数)は次のプロセスを通して定義される。“束の種別ごとに分ける”という関数を利用して対象のグループが束の大きさごとに分けられる。各々が対象のグループとして設定され④へ代入可能であるから、各々のグループに対して“量(個数)”と“束の種類”という二つの射影が得られる。“集った量(個数)”という関数はこれらの値の組を値とする関数として定義すれば良い。以下同様なプロセスで⑪名前(10進数)、⑫数量(個数)の順へ定義が進められる。

6. あとがき

機械が持つ固有の機能と簡単な具体物を用いて機械内部に、これらの連合体としての複雑な概念や概念を操作するアルゴリズムを構成しようというのが目的である。このため多くの思考実験を行って、システムに必要な原始的な見方の分析をつづけてきたが、実さいには小さいシステムでもかなりの量になる。これをある非常に狭い範囲の分野に限定して定義システムとして実現してみたいと考えているが、まだ試みの段階であって、今後多くのアイディアを細かく検討して行かなければならないと考えている。

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場(=情報処理学会電子図書館)で公開されているにも拘らず、古い報告集には公開されていないものが少なからずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者(論文を執筆された故人の相続人)を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長(tsuji@math.s.chiba-u.ac.jp)までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>