数理解析研究所講究録
第155巻 1972年 31-42

数理論理研究で行なった長時間計算

津田塾大　細井　勉
広島大　関本年彦

§1. 序

数理論理の研究に際して，我々2人がそれぞれ個別に行った3つの長時間計算について，簡単に報告する．いずれも，計算機を個人用同様に使用できる立場に居なければ実行できなかったであろうような長時間計算である．この種の長時間計算が，数理論理研究にとって必要であるかどうか，あるいは有効な手段または好ましい手段であるかどうか，などは，数理論理学の内部の問題であるので，ここではそのための議論や弁解はしないが，ただ，特権を利用して他の利用者にめいわくを及ぼして実行したものではないことだけは，付記しておきたい．これらの計算に関するデータや経験をもとにして，組合せ問題に要求される計算を定量的にとらえ，問題をつかもうというのが，本報告の目的である．

本研究集会の趣旨は，大型機のバック・グラウンド・ジョ

ブ(BGJ)に適した超長時間計算について知ろうとすることであろう．そのためには，BGJに対してどの程度の機能が提供されるかを知ることが必要であろう．実際，石田晴久氏の講演は，そのような内容であろうと想像される．しかし，よく考えてみれば，提供される機能を知るということは，本研究集会のような，準備段階の集会では本質的でないのではなかろうか．(現状がどうなっているかは，実際には，興味を感じていることであり，石田氏に対してケチをつけているつもりではない．)どの計算センターにおいても経験しているように，計算機の能力がどんなであろうと，その容量一杯に依頼計算が集まるのだから，BGJに提供される機能がどんなであろうと，それに合わせた依頼計算が十分に集まることは，目に見えているからである．

我々は，本研究集会の目的の一つを，

有効な超長時間計算を実行するためには，
BGJ用に，どの程度の機能が提供されることが望ましいか．

を探索すること，と理解し，少ないものではあるが，数理論理研究に際しての経験をもとにして，この問題に対する一つの解の例を提供することを試みたつもりである．

使用した計算機は，次の通り．

実験1，2

TOSBAC3300 （東大・理・数）

固定小数点加減算速度　　2.5 μS

実験3.

TOSBAC3400モデル41（広大・計算センター）

固定小数点加減算速度　　3.0 μS

§2. 数理論理計算の類型

数理論理研究の際の計算は，大体において，組合せ計算に属し，次の特色をもつ．

(i) 基本データは，ほとんど，1 byteに納まる．

(ii) 基本データの並べかえ操作が多い．

(iii) 比較判断が多い．演算は，せいぜい，固定小数点数の加減算だけであり，arrayの位置決めに乗算を用いることがあるかも知れない程度，除算はほとんど使用しない．

実際の計算は，次のような基本計算の組合せであることが多い．

(α) 与えられた正整数値nに対して，$\{0, 1, \cdots, n\}$の元を元とするn次正方matrix（モデルという）を，次々と生成する．

(b) 与えられた論理定数，論理変数，論理演算子，から構成される論理式（wffという）を，次々と生成する．

(c) 与えられたモデルにおいて，与えられたwffが正しいかどうかを判定する．

(d) 与えられたモデルが，与えられた代数的構造をもつかどうかを判定する．

(e) 公理系として与えられた論理体系において，与えられたwffが証明可能かどうかを判定する（証明のプログラミング）．

(f) 与えられた二つのモデルが同型かどうかを判定する．

計算（a）においては，大きなnの値を試みたいという欲望が生じるが，組合せの数から考えて，$n \leqq 10$ 程度が実用範囲であろう．したがって，matrixを入れるメモリは，実際上，ほとんど問題にならず，計算時間だけが問題である．$n \leqq 10$ でも，(f) の判定は意外に時間がかかるものである．(b) においては，論理変数の個数が，原理的には可算個であること，および，論理変数の種類を有限個に限定してもwffの個数が可算個であることから，どこで打切るべきか という問題がある．(c) は簡単である．(d) は，(c) ほど簡単ではないが，さほどに複雑ではない．(e) は，命題論理に関するかぎり，wffの分解法が個々のwffに関して有限なので，簡単である．

しかし，述語論理においては，分解法が有限ではないので，計算時間とメモリの点で，打切り問題が残る．

§3．実験1

1966年に，細井が行なったもの．

n点から成るrelatively pseudo-complemented latticeを（同型なものを除き）網羅的に求めることを問題とした．内容は，(a)の変形と(d)および(f)の組合せである．(f)を計算機が行なえば，outputは非常に少なくなったはずであるが，計算機時間を節約するため，(f)は，外側で人間が行なった．結果として，かなりのoutputがあった．inputはない．

latticeは順序集合であるから，matrixの元は，順序関係を示すための1（≧）と0（≧ではない）とに制限でき，matrixの総数は$2^{(n^2)}$個である．実際には，すぐ分る無駄な場合を除き，$2^{\frac{(n-2)(n-3)}{2}}$個にまで減じて，計算を行なった．

結果として，$1 \leqq n \leqq 9$に対して得られたlatticeの総数は，同型を除いて，62個だった．計算時間は，正確な記録をしなかったが，約500時間だった．この計算時間の大部分は，$n=9$の場合に使われている．

プログラムは，アセンブラ言語で書き，約1K語（1語24ビット）程度．したがって，結果論として言えることだが，62個のmatrixを記憶させて，同型の判断を行なわせたとしても，8K語程度（無理をすれば2K語）のメモリで処理できたようである．ただし，同型判断をさせると，計算時間は10～数十倍必要となろう．

ついでに述べれば，$n=10$の場合は，$n=9$の場合の約100倍の計算時間が必要であろう．このことから，$n=10$ぐらいが，この種の計算の限界ではないだろうか．

この計算をしているとき，待ち時間を利用して，プログラムの数学的な検討を行ない，何回かのスピードアップをした．$2^{(n^2)}$個のmatrixを$2^{\frac{(n-2)(n-3)}{2}}$個に減じたのも，その結果である．したがって，BGJでも，時々，プログラムに手を入れることを許してもらえるとありがたい．

§4．実験2

1968年に，細井が行なったもの．

二つの論理定数$a \supset b$と$b \supset a$，三つの論理演算子$\supset$，&，$\neg$から生成されるwffの集合Wを対象とする．a，bを論理変数とみなして，Wを，直観主義論理での同値性で類別した

クラスを知ることを目的とした．直観主義論理は，公理系によって定義されるので，証明のプログラミングが必要となる．

実際には，プログラミングが面倒だったので，(b)により，Wを生成することと，簡単なモデルを使って(c)の判断を行ない，あらい類別を行なうことだけを計算機に行なわせ，証明のプログラミングにまかせるべき所は，人間が手で行なった．結果としては，かなり短かいwffを調べることだけで，クラスが4個しかないことが判明したので，長時間計算にはならなかった．長時間計算になると予測して準備したのに，短時間で終ってしまった例である．

所要メモリ量は，実験1と同程度以下だった．

§5．実験3

1971年に，関本が行なったもの．

p，q，r，sを論理変数，Cをポーランド流の2変数論理演算子としたとき，

$$CCpqCCCqrpCsq$$

を満たし，

$$CCCpqrCCrpCsp$$

を満たさない，4次matrixを探すことが問題．

matrix の総数は 2^{32} 個，つまり約 43 億である．数学的な前処理により，この数を約 12.5 億に落としておき，計算機にかけた．結果として，約 0.5 億個の matrix を調べた所で，求める matrix が得られた．計算時間は 118 時間だった．

プログラムは FORTRAN で書き，オブジェクトプログラムの語数は約 3K語（1語 24 ビット）だった．この FORTRAN では，整数型データでも，1データあたり 2語使用していることでもあり，アセンブラ言語を使用すれば，プログラムはかなり短かくなるであろう．

求める matrix が存在しない場合には，約 3000 時間の計算となったはずである．

一般的なプログラムにしておいたので，二つの wff を指定するための入力を行なった．また，進行状況を知るために，5分毎に 1個の matrix を出力させた．いずれも，本質的ではないが，後者は，人間の精神衛生上，好ましいものであろう．

§ 6. 結論

少ない経験から大胆な結論をおすことは，あまり感心したことではないが，数理論理の計算を BGJ とする場合のほしい機能としては，次のようなことを結論として考える．

(α) 最低 8−16 K byte程度のメモリ. 逆に言えば, その程度で, かなりの仕事ができると思う.

(β) 計算の進行状況を知るためのメモリダンプを, 時々, 出したい.

(γ) 時には, プログラムのすりかえをしたい.

(δ) 入力は不要.

(ε) 出力用メモリ(8 K byte 位?)が与えられ, (β)よりはひんぱんに, その中味をはき出すことができれば, 非常に効果的。出力メモリがあれば, (α)は2 K byte 位でも何とかなるかも知れない.

補遺.

本報告の主題からはずれるが, 実験3を行った問題の背景, およびこの実験により得られた結果について簡単につけ加えておこう.

古典命題論理とは, Łukasiewicz−Tarski 流に定義すれば, 論理記号として 含意→, 否定¬, 論理和∨, 論理積& を用いた論理式のうち通常の2値モデルを満すものの集合である。この部分集合で, 論理記号として含意→だけを用いた論理式の集合を考える. これは Implicational Calculus of Propositions（以下ではICPと略す）と呼ばれている．以下では，ポーランド記法にしたがい，含意記号として→の代りにCを用いることにし，たとえば$p \to q$をCpqと書くことにする．

Tarskiは，3つの論理式

$CpCpq$

$CCCpqpp$

$CCpqCCqrCpr$

がICPをcharacterizeすることを証明した．すなわち，推論規則としてmodus ponens（もしp，Cpqならばqが推論される）および代入規則（ある論理式中の命題変数p，q，……等に別の論理式を代入して新しい論理式を推論する）だけを用いて上記3個の論理式から得られる論理式全体の集合は，集合としてのICPと一致する．

1925年にTarskiは「1個の論理式でICPをcharacterizeするものがあるだろうか？」という問題を提起した．この問いに対しては，肯定的に数個の論理式が発見されたが，Łukasiewiczは論理式の長さ（C，p，q，……等を1文字としたときの論理式の字数）のなるべく小さいものを見いだす研究を行い，1948年に13文字から成る前出の論理式

(1) $CCC_{pqr}\ CC_{rp}\ C_{sp}$

を発見し，かつ「長さが 13より小さい論理式の中には，ICPをcharacterizeするものはない」ことを証明した。しかし，Łukasiewiczは「長さ13の論理式でICPをcharacterizeするものが他にないかどうか？」すなわち「上記Łukasiewiczの公理(1)が最短公理として本質的にunique？」かどうかという問題は未解決のまま残した。

1968年Tursmanは，このuniqueness problemにつき，以下のような部分的解答を得た。すなわち「長さ13の論理式でICPをcharacterizeするものは，もしもŁukasiewiczの公理以外にあるとすれば，

(2) $CC_{pq}\ CCC_{qrp}\ C_{sq}$

だけであるが，この論理式(2)がICPをcharacterizeするかどうかは不明である。」

§5.の実験は，論理式(2)がICPをcharacterizeしないことを証明する目的で行ったものである。その結果，いくつかの所要のモデルを発見できた。これらのモデルを用いることによりŁukasiewiczの公理がICPをcharacterizeする本質的にただ1つの最短論理式であることを以下のように証明できる。

各モデルは2変数関数Cの関数値を与えるmatrixで表わ

すことができる。実際に発見したのは4次matrixであったが，これらをもとにつぎのような3次matrixも所要のモデルであることが分った。

C	1	2	3
1	1	3	2
2	1	3	1
3	1	1	2

このモデルは1をdesignated valueとする。一般に，あるモデルにおいて，論理式Aが成り立ち，Bが成り立たないときは，AからBを導くことが出来ない。しかるに上記モデルにおいて，(1)は $p=2$，$q=r=s=1$ とおくと

$$CCC211CC12C12$$
$$=CC11CC33=C12=3$$

となり，undesignated value 3をとるから成り立たないが，(2)は，p, q, r, s に1, 2, 3をどのように代入してもつねにdesignated value 1をとるから成り立つ。したがって(2)はŁukasiewiczの公理(1)を導くことが出来ないからICPの公理とはなり得ない。