

## 自動プログラム相談システムCONSについて

中川 雅俊

(名古屋大学・大型計算機センター)

## §1. はじめに

一般に、計算機センターには、プログラム相談という業務があり、計算機システムの効率良、使用、デバッグの相談、プログラミンケや数値計算の指導等において大きな役割を果たしている。特に年度毎にユーザの入替りが激しい大学の共同利用計算機センターにおいては、プログラミンケの経験者である指導員の存在は不可欠のものとなっており、要求度も高い。筆者は、過去の年間既にして千件近くの相談を受けました。その中で、もっとも多いのがデバッグ相談である。実際名古屋センターでは、翻訳時に20%、実行時に50%が、何らかのエラーを起して、満足に結果が得られていない。ところが、その内容を振り返ってみると、システムが用意しているエラーメッセージの不整備はもとより、(1)エラーの種類は千差万別あるが、頻繁に発生するエラーは大体決まっていま、同じようなエラー現象原因の相談が多い。(2)経験者にとっては、容易にエラー原因がわかる相談が多い。(3)相談者が指導員の前で説明していくうちに、少しの暗示で解決することがある。(4)相談したくても性格的にできない人がいる。という事に気付いた。

そこで、ある程度、バターン化によって、プログラム相談を相談者と計算機との応答システムによって実現できないだろうかとの考えから「プログラム相談の機械化システム」である。以下に、その概要について述べる。まずオ2章で、過去数年間のプログラム相談票(相談者か相談コーナーなどで相談を受けた際、その時の状況を書き込むもの)をデータとして、相談の分析を行い、オ3章では、プログラム相談そのものをシステムティックに解析し、プログラム相談の機械化が可能な否かを検討した。オ4章では、データを基に、あるエラー原因に対してマエラー番号が一つ出力される場合と、二つ以上出力される場合とに分けて、統計的処理の立場から検討した結果を示す。オ5章では、現在、名古屋センターでユーザ向けに公開されている自動プログラム相談システムCONSの説明を行い、最後に、今後の課題について述べる。

## §2. 相談の分析

## 2.1 相談内容

質問 ----- 24%

ジョブ処理結果について ----- 76%

## 2.2 難易度(相談に費した時間)

5分以内	6~10分	11~30分	31~60分	61分以上
44.85%	25.42%	25.10%	4.41%	0.2%

## 2.3 エラー番号

## 2.3.1 F230-60で用意されているエラー番号の数

ジョブ管理	約 70	タスク管理	約 150	LIBE	約 220	LIED	約 100
データ管理	約 440	ALGOL	約 220	FORTRAN	約 310		
COBOL	約 660	FASP	約 100				

## 2.3.2 ある期間中(約1年)にプログラム相談にきたジョブの発生エラー番号の数

149種類(うちFORTRAN 105)

### 2.3.3 発生エラー - 番号ベスト 20

ZZZ 1000	135	FT 722 Z	33	FT 731 Z	17	FT 142 X	12
FT 999 Z	60	FT 999WD	32	FT 988 Z	16	FT 151 Y	12
FT 999WO	49	SP 224 Z	30	LD 218 X	15	FT 240 X	11
FT 956 W	39	FT 957 W	23	FT 124 X	13	FT 744 X	10
FT 742 X	36	FT 166 X	21	FT 746 Z	12	FT 752 Z	9

残り 3 件以下 (うち 1 件のもの 65)

### 2.3.4 エラー原因

配列オーバー	11.5%	INPUT DATAミス	6.4%	CONTROLカードミス	11.5%	初期値設定ミス	5.0%
宣言文ミス	10.1%	FORMATミス	10.1%	パンチミス	10.1%	その他(引数の対応ミス等)	20.4%
ライブラリ使用ミス	6.4%	論理ミス	8.5%				

## § 3. 相談の解析

プログラム相談は通常、計算機利用者がプログラムを作成する際、又は、した際に発生した問題をプログラミングのベテランである指導員に個別に面談し、疑問点を説明しながら、その解決方法を教示してもらうという方式をとっている。これを機械化するために、プログラム相談なるものを解析してみた。図 1 は筆者自身が相談にあたる時の事を考へて、図に表わしたものである。相重なるところもあるが、これら全てが満たされて問題が解決されるのではなからうか。即ち、種々な情報が満たされ、ミックスされ、それを人間が取捨選択し、解を求めゆく。この情報の量が大きいことが、プログラム指導員の必須条件となるであろう。先ず最初に相談者が何を求めたいかを知り、次に解を求めるためのエントリーを見つけ、それからシステム・ツールに状況を分析し自分の持つ情報を駆使して、可能性の高いものから原因を探し相談者と会話しながら、問題を指摘しつつ説明していき、というようなステップを踏むのではなからうか。こう考えると、プログラム相談とは経験、知識等の量の情報の中から、いかに問題解決に最適なものを選び相談者に表示するか、ということになる。亦、3章の検討により、計算機システムに、必要な情報を蓄えておき、それを場合場合によって確率の高い順に相談者に表示し答へるという方式をとれば、プログラム相談の機械化は可能であるといえる。そこで次に最適な情報の選択について、統計的処理の立場から述べてみる。

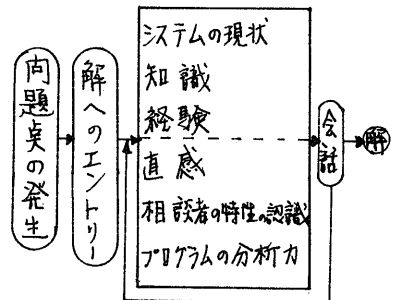


図 1. プログラム相談の形態

## § 4. 統計的処理からの検討

### 4.1 エラー番号が一つのエラー原因に対して 1 回出る場合

4.1.1 あるエラー番号をタイプインすると過去の経験データを基に作成された表をサーチし、頻度の高いものから順に表示する。

### 4.1.2 Bayes の定理の利用

上記のように  $X_{ij}$  に過去の経験サンプルの発生度数のみを用いるのではなく経験年数、研究分野等の情報を用い Bayes の定理により事前確率を求め、より効率良い診断を得る方法を考えられる。

	エラー原因項目 (j)	
エラー番号 (i)	$X_{11} \dots$	$X_{ij} = \max_j X_{ij}$
	$\vdots$	
	$X_{ij}$	

4.2 エラー番号が二つ以上出力される場合

この場合、エラー番号がそれぞれ互いに与える影響を考え、真の原因を考えなければならぬ。

4.2.1 判別分析を用いる方法

エラー番号を観測値として、あるエラー番号が出たとき、どの母集団（この場合はエラー原因）に含まれるかを判別する。つまり各種の標本からなる観測値  $X'_\lambda = (X_{1\lambda}, X_{2\lambda}, \dots, X_{6\lambda})$   $\lambda = 1, 2, \dots, n$  が得られているとき、ある観測値を各変量正規母集団  $\Pi_\lambda$  (母集団分布  $N(\mu^{(\lambda)}, \Sigma)$ ) に分類できると考えるわけである。そこで、この考えに基づき、ある期間中のプログラム相談票の中から、エラー番号が二つ以上あるデータをピックアップし観測値として判別関数を求め検討してみたので報告する。ただし観測値は50件、エラー番号は25種類、エラー原因は11種類あるが、表1においてデータの全てを記すことは紙面上無理なので簡略して書いた。

4.2.2 求められた判別関数

例えば、グループ3及び5の判別関数は、それぞれ

$$\hat{\Delta}^{(3)}(X) = 4.28 X_1 + 2.05 X_2 + 4.0 X_3 + 6.52 X_4 + 1.88 X_5 + 9.34 X_6 + 8.09 X_7 + \dots + 10.67 X_{12} + 7.29 X_{13} + \dots + 0.65 X_{24} + 0.20 X_{25} - 4.29$$

$$\hat{\Delta}^{(5)}(X) = 10.88 X_1 + 10.17 X_2 - 1.48 X_3 + 10.85 X_4 + 0.13 X_5 + 6.12 X_6 + 9.21 X_7 - \dots + 0.64 X_{24} + 0.24 X_{25} - 4.95$$

となった。従って、あるデータが発生した時、順次  $\hat{\Delta}^{(i)}(X)$  ( $i=1 \sim 11$ ) に代入し判別していき、最終的に大きい値をとるものをエラー原因とすればよい。ここで、上式を例として調べ、この式が判別関数として妥当か否か検討してみよう。グループはエラー原因が宣言文ミスである。係数の大きいエラーメッセージ  $X_6, X_{12}$  の意味は、 $X_6$  は「かぶり」、 $X_{12}$  は「宣言文関係の文で規

エラー原因	エラー番号		FT124X	FT141X	FT142X	FT155X	FT166X	FT169X	FT999X		エラー原因の意味
	サンプル1	サンプル2	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub> ... X <sub>24</sub>	X <sub>25</sub>	
Group 1	サンプル1	サンプル2						1			FORMAT文ミス
	サンプル3	サンプル4			1						
	サンプル5	サンプル6					1				
	サンプル7	サンプル8								1	
	サンプル9	サンプル10							1		
Group 2	サンプル1	サンプル2								1	配列オーバー
	サンプル3	サンプル4	1		1						
	サンプル5	サンプル6			1		1				
Group 3	サンプル1	サンプル2									宣言文ミス
	サンプル3	サンプル4			1		1				
Group 5	サンプル1	サンプル2									パンチミス
	サンプル3	サンプル4	1				1				
	サンプル5	サンプル6									
	サンプル7	サンプル8									

表1. 使用データ

定どおりの記述がしてない)であり、まさしく宣言文の誤りによることか大きい。又、一方、係数の小さい  $X_{24}, X_{25}$  は、 $X_{24}$  が「DIVIDE CHECK」、 $X_{25}$  が「INVALID INSTRUCTION」であり、これは余り宣言文ミスとは関係がない。又グループ5に関して調べてみると、係数の大きい  $X_1, X_2$  では、 $X_1$  は「許さない文がある」、 $X_2$  は「ある文字が既に他の目的に使用されている」であり、エラー原因はパンチミスによることか多い。又係数の小さい  $X_{24}, X_{25}$  は実行時のエラーメッセージであり、パンチミスは殆んど翻訳時に見つかることか多いので余り関係がない。従ってグループ3, 5共に判別関数は妥当であるといえる。他のグループに対しては吟味してみたが、サンプル数の少ないグループは別として、一応の妥当性が確認できた。

### 4.2.3 あるサンプルが真のグループに属する確率の推定値

いくつかの例を作り、計算した結果

FT166X ( $X_5$ ) と FT245X ( $X_{13}$ ) の場合、グループ 10 に属し、その推定確率	81.44%
" ( $X_9$ ) " ( $X_{12}$ ) " 8 "	86.24%
" ( $X_8$ ) " ( $X_{12}, X_{17}$ ) " 7 "	99.64%
" ( $X_1$ ) " ( $X_{16}$ ) " 5 "	89.90%
" ( $X_{17}$ ) " ( $X_{20}$ ) " 6 "	22.25%

となった。

この結果より、エラーメッセージが二つ以上発生した時判別関数を用いることが有用であるといえる。

### 4.3 システム化への応用

以上が統計データより検討した結果である。しかし、Bayes の定理、判別関数の応用は有用であることは確認できたものではあるが、これらの方法の確立に充分といえるデータ量が不足しているため、この方法により実際のシステムを作るには無理と思われる。そこで、現時点では、単純に発生頻度を用いる方法が最良と考へ、CONS を作成した。

## 5. CONS について

### 5.1 概要

CONS (Consultation of Nagoya System) は TSS 端末 (現在は 50 ボータイアライター) を媒体として F230-60 より発生した問題に関するプログラム相談をおこなうもので、例えば、発生したエラーのエラー番号をタイプラインすると、ある質問が CONS より寄せられ、それに対して Yes) または No) をタイプラインする。この繰り返しにより、あたかも指導員と対話しているような感じで、端末と会話しただけから問題を解決していくシステムである。もちろん人間と機械との間の大きなギャップはあるが、極力相談者にとって自由度の高い、親身さをもったシステムになるよう工夫してある。現在 CONS にはエラーデバック用ルーテンとファイル使用状況供用ファイルを使っている利用者用で、現在どのようなファイルを使っているか (どの大きさで使っているか) を知る事ができる) ルーテンがある。ファイルルーテンの方は単にファイルラベルをひとまとめでしたマトロクとマッチングして情報を出力するのみで、この橋では触れない。次に CONS の特徴を述べてみる。

1. Yes/No の答えによりアルゴリズムに沿って進むデバックルーテンの他にキーワードのタイプラインによる辞書的なサーチも可能にした。これは、相談者のレベルが千差万別の為、ある相談者にとっては当然知っていることで、他の人にとっては全然知らない場合がある。その為のオプションとして使うことができる。
2. 全体に通し番号をつけ、他のエラー番号に関して調べている時でも、その番号をタイプラインすれば随意に希望する箇所へ戻れるようにした。これは相談者にとって自分のペースで自分の気の向くやり方で進めるよう自由度を与えるためである。特にある所まで進み、前の方で勘違いを起こしていたことに気付き戻ると便利である。
3. 詳しい説明及び例題表示が必要な人、否。人のためにサブコマンドを設けた。初心者にとっては、必要な情報であっても、ある程度知っている人にとっては、全く不要なものであることが多い。その場合、不要なデータをタイプアウトさせることは相談者にとって煩わしく疲労度が大きい。そこで各々の相談者にとってなるべく必要な情報のみを与えるために考へた。
4. このシステム自身の経験データ (質問-応答) の進み方、不足しているデータを蓄

積させ、改良のための資料とする。

5. 必要なデータがアルゴリズム表を速くサーチし得るよう、極力無駄、無いファイル構造にした。データが多くなればなる程サーチ時間の短縮が大きな課題となる。

6. 更新容易なデータ構造にした。

### 5.2 データの作成

このようなシステムにおいては、集積したデータの良し悪しによってシステムの良否が決まるといってよい。いかにして多様に渡るデータを正しく求めるか。又必要なデータが経験的なものに依存する訳であるから、経験法則を導出するに足る大量のデータをいかにして集めるかが、先ずオノの課題である。幸いなことに名大大型計算機センターには過去5年間のプログラム相談に携わった人々によって記入されたプログラム相談票約千枚余り全て保管してあり、その票を基にして、このシステムのデータを作成することができた。もちろん記入もれ、論理的におかしいデータもあり、又表現不足のため理解出来兼ねるものもあったが、全ま目を通し、特に記事欄に記入されているものは全て書きあげ一応利用できる形にまとめあげた。

### 5.3 GENERAL FLOW

CONSの全体の大まかな流れは 図2 に General flow として示す。

### 5.4 ファイル及びデータ構造

CONSはメインプログラムのみ主記憶上に置き、必要なデータは、その都度データファイル(磁気ディスクパック使用)よりアクセスして来る。以降にファイル構造及びデータ構造を示す。

#### 1. ET (Error Table File)

エラーテーブルとは、エラー番号をサーチするための情報が入っているテーブルであり、INDEX部、エラー番号部、LINK部、COUNT部からなる。先ず、タイプラインされたエラー番号を言語部分と番号部分に分ける。

(F230-60 の出力するエラーメッセージにはエラー番号がついており、例えば FORTRAN のプログラムミス等起こしたエラーには FTnnn なるエラー番号が出力される。) 次に FT, LB(LINK BE という処理プログラムの場合エラー番号は LBnnn とする) などにより言語別に INDEX 部からエラー番号部のエントリーを求め、そこから順にタイプラインされた番号と等しい番号があるか否かをサーチして行く。但し、FORTRAN (FT) に関しては使用頻度が高いと考えられるので FT のエントリーに番号の値そのものを加えたものが、そのエラーに対する項目となるようにしてある。もっとも FT001~999 に全まエラーが割り当てられている訳ではないから欠番に関しては LINK 部に all blank を入れてあ

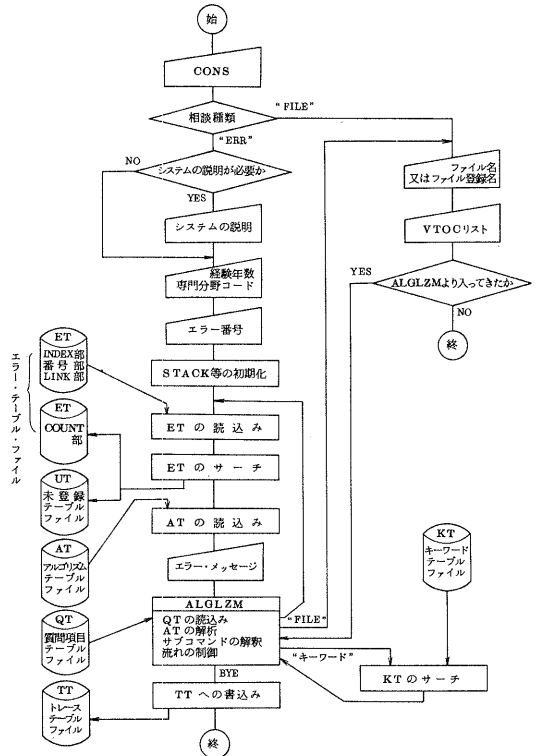


図2. CONS General Flow

る。従って空領域が、いくつかで無駄を生じるが、その代りアクセスの速さを得ることができる。LINK部とはアルゴリズムテーブルへのLINKであり、そのエラー番号に該当するアルゴリズムテーブルが格納されているファイルの元素名が入っている。

COUNT部は、それぞれのエラー番号が何回参照されたかを記録しておくものである。

### 2. UT (Undefined Table File)

ETに登録されていないエラー番号が入力された場合タイアインミスが無いかどうかを確認の上で、そのエラー番号をこのテーブルに登録しておく。COUNTの扱うデータはF230-60の出力するエラーメッセージ全てではなく、5.2で述べたように過去に発生しているエラー番号(プログラム相殺を受けたエラー)のみを対象として作っている。従って、このUTに記録されたエラー番号に因しては、適宜ETに登録し、データの拡充を計っていく方法をとっている。これは、過去5年間の経験データの分析による利用者が発生させるエラーの種類が、ある程度限られており、全データ(F230-60の全エラー全て)を対象とすることは、データファイルの容量、サーチに要する時間の増加を、まねくだけであるという考之によるものである。又、UTには、そのエラー番号が何度タイアインされたかをカウントし、そのエラーの必要度を後から解析できるようにしてある。

### 3. AT (Algorithm Table File)

ATファイルは分割型順編成ファイルで、1つの元素に約10個のエラー番号に対するアルゴリズムテーブルが格納されている。従ってETのLINK部にはこの元素名が入っている。10個の内どのエラーかをサーチするには、エラー番号でマッチングする。実際にファイルに書き込まれているデータの中味の1例を見やすくリストアップしたものを表2に示す。アルゴリズムテーブル(AT)とはエラーの原因を追求していくエラー解析部であり、各々のエラー番号に対し1個ずつある。ATの構造例を図4に示してあるが、このATにおいて、YADR、

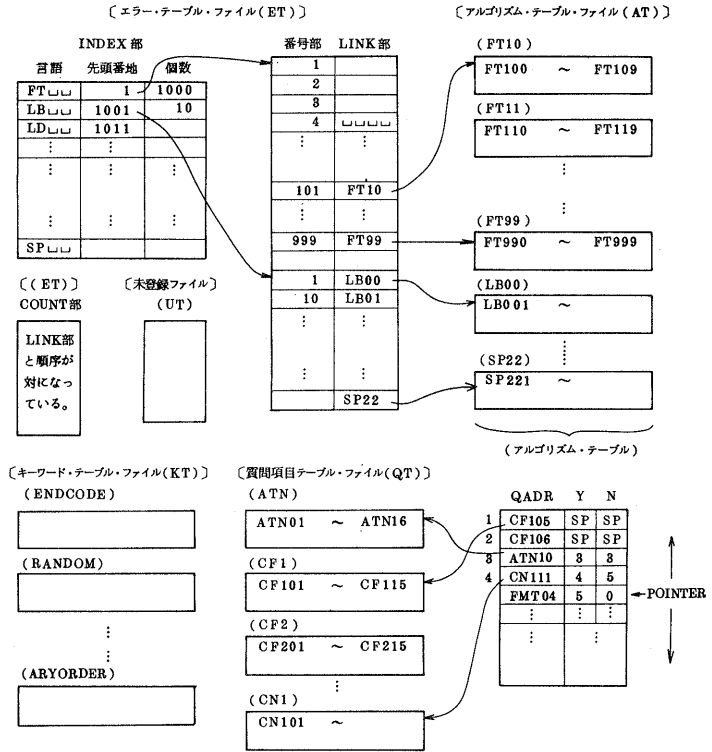


図3. データ・ファイル構造

このATにおいて、YADR、

### 3. AT (Algorithm Table File)

ATファイルは分割型順編成ファイルで、1つの元素に約10個のエラー番号に対するアルゴリズムテーブルが格納されている。従ってETのLINK部にはこの元素名が入っている。10個の内どのエラーかをサーチするには、エラー番号でマッチングする。実際にファイルに書き込まれているデータの中味の1例を見やすくリストアップしたものを表2に示す。アルゴリズムテーブル(AT)とはエラーの原因を追求していくエラー解析部であり、各々のエラー番号に対し1個ずつある。ATの構造例を図4に示してあるが、このATにおいて、YADR、

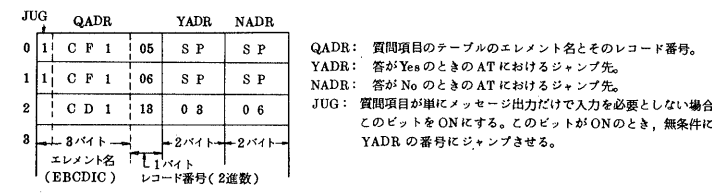


図4. アルゴリズム・テーブル

NAQRには、一般にATの中に付けられた番号が与えられているが、他にSP (STOPコード、このコードが来た時、このアルゴリズムは終ったとみられる。プログラムの方で終結処理をする)、NP(NON OPERATOR NONコード、何もしずに次の番号へ進む。これはATを修正し易くするために作った)がある。なお、ATの0番目--解決した項目(シユウセイオスル) / 1番目--未解決の項目(アロソウヘ)

は、どのATでも固定とし、それぞれATが、どのような状態で終了したかを知る事が、できるようになっている。

4. QT (Question Table File) 端末に打ち出すべく質問メッセージが格納されており、ATのQADR部により、このQTがCALLされる。QTファイルは分割型順編成ファイルで、一つのエレメントに対し最大18の質問項目が格納されている。それぞれのエレメントに含まれる項目は、関連性の高い項目でまとめている。1グループに18個以上の項目がある場合、1エレメントのブロックサイズを余り大きくしないために、数ブロックに分けてある。現在、大きな項目は次のとおりである。

- 1. コントロールカードミス
- 2. FOR MAT 文ミス
- 3. パンテミス
- 4. DATA ミス
- 5. リ副プログラム引数ミス
- 6. オ配列オーバミス
- 7. ス参照項目
- 8. ネット状況項目
- 9. 宣言文ミス
- 10. 初期値設定ミス
- 11. 変型違いミス
- 12. W. DATA 文ミス
- 13. L. LIE 文ミス
- 14. 誤である。ここで参照項目とは“何Rの何ページを見よ” というような項目で、エラー原因を述べた項目とは、少しニュアンスが異なる。又、状況項目とは、例えば“SUBROUTINEを使っているか” というように、解へ導く質問ではないが、アルゴリズム上で不必要な箇所へ進まなくともよいようにするための状況判定用のメッセージである。表3は、QT7

```

**FT983Z シ*ツクシ* ノ イ5- カ* オオキ*ル*??
00 1 CF206 SP SP
01 1 CF106 SP SP
02 0 ET205 03 01
03 0 ET206 04 00
04 0 ET207 05 SP
05 1 CF207 SP SP

-----
**FT984W SUBCHK フ*ン*チ* シ*シ*シ* ハイ*ツ*カ* タイ*ノ* ハ*ン*イ* コ*シ* -5DEBUG**FT984W ISN
(N) A(M) ARRAY BOUND(L) OVER AT XXXXXX -8N...I5- オオコ*シ*ル* ISN オシ*ス。-8
A...シ*ツ*ク* ノ* タ*イ*シ*ヨ*ト* フ*ツ*ク* ハイ*ツ*ク*。 -8M...オ*ハ*シ* ヴ*シ* ノ* ア*シ*。 -8L...シ*ツ*ク*シ*
ハイ*ツ*ク* ノ* オ*シ*。 -8XXXXXX...I5- ノ* ハ*ン*イ* ハ*ン*チ* -7I5- ノ* ケ*ン*イ*ン* カ* ヲ*カ* ノ* イ*ト* ハ* FT
999Z ノ* コ* オ* シ*ツ*ク* ス*ル* コ*ト* -9(*) ノ* ア*ト*ニ* FT999Z ト* タ*イ*フ*イ*ン*ス*??
00 1 CF105 SP SP
01 1 CF106 SP SP
02 0 CD110 00 01

-----
**FT985 I=D マ*ハ* I=R チ* D(マ*ハ* R) ノ* ア*シ*カ* 2**35 イ*シ*ヨ*ク*ニ* フ*ツ*ク*。 -9(コ*コ*ク* I:シ*
ハ*ン*ス*ク* R:シ*ツ*ク*シ*ク* マ*ハ* シ*ツ*ク*ヨ*ク*ス*ク* D:I*ハ*イ*シ*ト*シ*ツ*ク*シ*ク* マ*ハ* H*イ*シ*ト*シ*ツ*ク*ヨ*ク*ス*ク* -7I
5- ノ* イ*シ*カ* ヲ*カ* ノ* イ*ト*キ* ... REF.<<INTRANG>>??
00 1 CF105 SP SP
01 1 CF106 SP SP
02 0 IST01 08 03
03 0 SUB07 04 00
04 0 IST02 05 00
05 0 CD105 06 08
06 0 IST03 07 00
07 0 ET101 08 00
08 0 OVR01 09 01
09 1 CF102 10 10
10 0 CD110 00 01

```

表2. ATファイルの内容

```

1. FORMAT フ*ン*ハ* タ*ク*シ*カ* ?%
%
EX)%
K=100%
L=4000%
WRITE(6,1) K,L%
1 FORMAT(1H ,I3/I4)%
%
ト* ス*ル* ト* シ*ツ*ク*ヨ*ク* %
100%
000%
%
I* タ*ル*。 コ*ル* ハ* %
1 FORMAT(1H ,I3/I1H ,I4) ト* ス*ル*ハ* ヲ*イ*%
2. シ*ツ*ク*ヨ*ク*ヲ*シ*ト* FORMAT ノ* タ*イ*ク*カ* ト* ス*ル* イ*ル* ?%
%
EX)%
COMPLEX A,B%
WRITE(6,1) A,X,Y%
1 FORMAT(1H ,2HA=,F5.1/1H ,2F5.1)%
%
ト* ス*ル* ト* シ*ツ*ク*ヨ*ク* ハ* %
A=364.1%
20.3 98.2%
A= 50.6%
%
I* タ*ル*。 コ*ル* ハ* A ノ* IMAGINARY PART ニ* タ*イ*ス*ル* シ*ツ*ク*カ* ノ* イ* タ*ク*シ*ク* ア*ル*%
1 FORMAT(1H ,2HA=,2F5.1/1H ,2F5.1) ト* ス*ル*ハ* ヲ*イ*%
3. FORMAT フ*ン*ノ* 'I' ノ* タ*イ*ク*ハ* ト* ス*ル* イ*ル* ?%
4. オ*ハ* E13.5 ト* ス*ル*キ*ト*コ* オ* E13.5 ト* シ*ク* 'I' ト* 'I' オ* マ*カ*イ*フ* イ*ル* ?%
5. シ*ツ*ク*ヨ*ク*ノ* カ*シ* ハ* タ*ク*シ*カ* ?%
EX)%
SHABC.F1015 ト* ス*ル*ト* I5- ト* タ*ル*%

```

表3. QTファイルの内容

ファイルの中の1つで、FOR MAT文ミスに属する質問項目の一部をリストアップしたものである。

### 5. K7 (Keyword Table File)

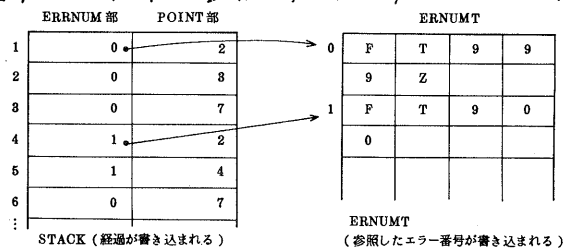
それぞれのキーワードに対するメッセージが格納されており、ファイル構成は、QTFファイルと同じである。質問項目の中に<<と>>で線くられたキーワードのみを扱う。

以上が、個々のファイルデータテーブルの説明である。それぞれのファイルの関連性は図3に示しておく。

### 5.5 アルゴリズム

CONSの中核部分をなす。タイアインされたサブコマンドの解釈、流れの制御部分である。図2のALGLUMZMの部分も言う。細かいステップの説明は省くが、この流れは、ATとSTACK, ERNUMTと数種のポインタによって制御されている。さて図4のATの構造をみるとYes又はNoによって進む方向が決まる。つまりこの構造はBinary Treeになっている。

CONSでは適宜、このTreeのあるnodeに来た後、前のnodeへ戻る事も考慮に入れている。STACK等は、このように戻る時にも使用している。又何れを調べる際、大きくグループ分けし、それぞれ対等のレベルの入口を作り、自分でその入口を選別、そのグループを調べ、もし必要が違くれば、他のグループを調べることも可能にしてある。これは特に対象となるものが多の場合、又は、より分かれている場合有効である。CONSでは、この並列のTreeの入口を定義するためにSTACKの201以降を使い、特別のエントリーとしてある。例えばFAT999Zなるエラー(命令を壊した場合に生じるエラー)の場合、エラー原因を大きく分けるとの添字のエラー、②データの受け渡しミス、③宣言文ミス、に分けられる。従って、まず、自分でチェックしたいと思うチェックルーチンへ入るためには、特別のエントリー(この場合201, 202, 203が用意されている)をタイアインすることになる。



STACKとERNUMT

### 5.6 サブコマンド

CONSは、サブコマンドをタイアインすることにより会話を進めていく。

- Y---質問に対する答えがYesのとき。
  - N---質問に対する答えがNo又は、どちらともいえないとき。
  - エラー番号---そのエラー番号についてのチェックを行うことを指示する。
  - 質問番号---その番号の質問に戻る。
  - EX---例題などの詳しい説明を必要とするとき(標準はEX)。
  - NOEX---説明が不要のとき。
  - キーワード---説明を必要とする言葉。質問の中で<<と>>で囲まれた形で現われる。
  - BYE---CONSを終りたい時。
  - FILE---ファイルの使用状況を知りたいとき。
- 以上のサブコマンドは任意の場面で使用できる。

### 5.7 実例

表4に簡単な実例を示す。



## § 6. 今後の課題

C O N S は 51 年 12 月より 東大 大型 計算機 センターで、一般利用者向けに公開された。プログラム相談の仕事の一端を担っている。プログラム相談の機械化とは、プログラム相談を全て機械でおこなうものではなく、より適切なプログラム相談をするための手助けとなるものをつくることであろう。このシステムの最初の設計目的は機械でできるような簡単な相談は、機械にやらせれば、人間のおこなう相談は、より充実した高度の相談に専念できる、というものであった。もちろん、このようなシステムは未だ出発点に立った状態であり、より研究を進め、高度な相談もできるシステムにしていくべきであろう。C O N S では、エラー原因を探る方法として確率論的なアプローチを用いたが、東大、阪大センターでは、又異なる方法を、とっており、今後の報告が興味深い。なお、C O N S は、単にプログラム相談のみでなく、より広い意味での計算機利用方法の解説、啓蒙といった教育的な利用の可能なシステムへの拡大を目指しており、現システム自体についても、その利用記録の解明、利用者の意見の集積といった方法による改良を進め

```

① ---- **KD
② ---- USER-NAME...WA? CNS.4000ZA0939
/* YOUR JOB-NO... 031208765TS 75.12.08 15:21:19
MACRO BUJ NYUURYOKU.
③ ---- **CNS
JOB KAISI.
ソウチン ショウ (ERR マシ FILE) オ タイグン セ。
④ ---- >ERR
エラー システム ノ エラー ガ エラーカ? Y マシ N オ タイグン セ。
⑤ ---- >N
プログラム エラー システム、センタリング コード オ タイグン セ。
リカグイ...100 フカグイ...101 ソウチン...102
イカグイ...103 ソウチン...104 ソウチン...105
>1101
エラー システム オ タイグン セ。
>ET956W
**FT956W F-OVERFLOW, A-OVERFLOW
:オキチ ショウ
⑥ ---- {
1. ショウ ノ エラー ニ エイジヨウ ナ マシ ガ ハイグン イイ。
2. プログラム ノ エラー ノ チョウ ノ ウケツキ ガ アカシ。
3. ファイル ノ エラー ミス。
4. ハイグン オール ニ ヨリ エイジヨウ ナ マシ オ コウシ。
⑦ ---- 1 エラー ノ エラー システム <<ERRADD>> ハ ショウ ノ プログラム ノ ナカ?
>Y
2 F-OVERFLOW カ?
>Y
3 オキチ マシ オ チョウ マシ チョウシ、オキチ マシ ノ ショウ ノ エラー ノ エラー (ABS(10**76) カラ ABS(10**76) <<REALRANG>>) オ エラーカ?
⑧ ---- >REALRANG
ショウ ノ エラー:
F230-60 チョウチン ショウ ノ エラー ハ 1.0E-76<ABS(X)<1.0E76, X=0.0
チ アル。グロブ X=Y*Z アル シキ チ Y=1.0E40,Z=8.0E40 ト スル
X ノ マシ ハ 8.0E80 ト マシ ショウ ノ エラー オ エイジヨウ ナ マシ
マシ ショウ。ソウチン FLUATING-OVERFLOW ト マシ エラー アル。
ショウ ノ エラー オ スル ハ マシ ノ エラー ニ ショウ シテ マシ。
>N
4 エラー ノ シキ ノ エラー ニ エイジヨウ ナ マシ ガ ハイグン イイカ?
>Y
5 ハイグン オール <<ARYOVER>> ノ エラー ハ アカシ?
>Y
6 REF. *FT999Z (ハイグン オール)
7 エラー ショウカ?
⑨ ---- >ET999Z
**FT999Z INVALID INSTRUCTION
コ エラー ノ エラー ショウ
1. ハイグン オール
* エイジヨウ ナ マシ ガ アカシ
* プログラム ノ エラー ノ チョウ ノ ウケツキ ガ アカシ
2. センタリング ミス
*** ショウチン エラー ***
201. エイジヨウ ナ マシ
202. チョウ ノ ウケツキ ショウチン
203. センタリング ショウチン
8 ショウチン エラー エラー ノ エラー オ タイグン セ
⑩ ---- >202
9 ***マシ-マシ ノ ウケツキ ショウチン エラー ***
10 エラー ショウ ト エラー ショウ ノ エラー ハ マシ イイカ?
EX)
DOUBLE PRECISION A
CALL SUB(A,B)
SUBROUTINE SUB(A,B)
コ エラー ショウ プログラム チョウチン エラー ショウチン
>N
11 ショウチン オ スル
** エラー コウチン エラー ニ カンシテ ショウチン ハ エラー マシ
⑪ ---- >2
12 F-OVERFLOW カ?
>N
13 オキチ マシ
>Y
21 エラー ショウカ?
>Y
22 ショウチン オ スル
** エラー コウチン エラー ニ カンシテ ショウチン ハ エラー マシ **
⑫ ---- >EXIT
マシ ショウチン エラー ショウチン。マシ エラー マシ ショウチン
/* CPU-TIME 00:00:29.835
JOB PWARI.
ZIKOKU 15:46 NI OFF-LINE NI NATTA. SIYECZIKAN 00:25:28

```

表 4. CNS の使用例。

(下線は、ユーザの入力を示す。⑩...は説明のため記入したものを示す)

ることを考えてある。一方、利用者に対して現実的な問題点としては、現在センターの現有計算機システムのTSSの応答時間の遅さ、日本語をカナ文字、英数字で表現した際の読み難さ、理解の困難さ、初心者計算機にまつ恐怖心、不慣れ(CONSは、前述のように本来初心者を対象としているのだが、初心者の中にはタイプライターの電源ボタンを押すことさえ躊躇する人が多く、実際にシステムの中まで入れない人がいる)など一朝一夕では解決し得ない問題も多くある。又現在のシステムは、過去のプログラム相談の経験に基づいて作られたものであるから、センター利用者が持っている問題を全て網羅しているとは決して云えず、それらの問題をどう把握するか、マデータをいかにして集積するかは大きな問題である。結局、計算機センターのこのようなシステムにおいては、利用者が多岐多様な問題点を解決するものである必要性がある。従って、当然、システム自身が、ダイナミックに成長していく学習システムの性質を持ち、CAIシステム、情報検索システム等を含んだ総合システムにしていく方向づけが必然となってくるであろう。そこで、先ず一歩として、華いなることに当センターには52年1月よりプログラム相談用として、キャラクターディスプレイが導入されるので、これを駆使し、より進んだシステムを作成するつもりである。

おわりに

この研究は料研費によるプログラム相談の機械化研究」に端を発しており、田中一教授(北大)、小野周教授(東大)を指め各関係者に深く感謝致します。TJおCONSは近藤一郎助教授(北大)のもと津田知子氏西本史雄氏(北大センター)らとの共同開発であります。

参考文献

- (1) 高橋他：臨床診断とコンピューター
- (2) 情報開発センター：CAIシステム
- (3) Donald E. Knuth：The Art of Computer Programming
- (4) 北大計算センター：デバ-7の手引
- (5) 大槻 泰野：“ボタン認識によるプログラムの診断” 情報処理 Vol. 10, No. 4, 1979
- (6) 北川敏男：多変量解析論
- (7) 中川：北大計算センター研究開発報告 No. 1, No. 2
- (8) 中川 津田西本：プログラム相談システム“CONS” 新計算機センターニュース Vol. 7, No. 1, 1974.2

附 録

7/16/1979		0/HT999Z																																																																							
(西暦の下2桁) 相談受付年月日		所 属 名 姓 工 字 部	相談機種 (A) 使用言語 (B)																																																																						
エラー番号 (REJECTの場合はREJECTコードを記入してください。)		エラー原因 (C) 相談結果 (D)																																																																							
<table border="1"> <tr><td>1</td><td>ジョブ始末結果について</td><td>01 FORTRAN(C)</td></tr> <tr><td>2</td><td>質問</td><td>02 FORTRAN-D</td></tr> <tr><td>3</td><td>(内容を記事欄に記入のこと)</td><td>03 FORTRAN-E</td></tr> <tr><td>3</td><td>システムIIについて</td><td>04 FASP</td></tr> <tr><td>4</td><td>その他</td><td>05 ALGOL</td></tr> <tr><td></td><td></td><td>06 COBOL</td></tr> <tr><td></td><td></td><td>07 PL/I</td></tr> <tr><td></td><td></td><td>08 LIME</td></tr> <tr><td></td><td></td><td>09 LIED</td></tr> <tr><td></td><td></td><td>10 アプリケーション</td></tr> <tr><td></td><td></td><td>11 ユティリティ</td></tr> <tr><td></td><td></td><td>12 その他</td></tr> </table>		1	ジョブ始末結果について	01 FORTRAN(C)	2	質問	02 FORTRAN-D	3	(内容を記事欄に記入のこと)	03 FORTRAN-E	3	システムIIについて	04 FASP	4	その他	05 ALGOL			06 COBOL			07 PL/I			08 LIME			09 LIED			10 アプリケーション			11 ユティリティ			12 その他	<table border="1"> <tr><td>01</td><td>コントロール・カード・ミス</td></tr> <tr><td>02</td><td>FORMAT文ミス</td></tr> <tr><td>03</td><td>配列オーバー</td></tr> <tr><td>04</td><td>宣言文ミス</td></tr> <tr><td>05</td><td>論理ミス</td></tr> <tr><td>06</td><td>パンチ・ミス</td></tr> <tr><td>07</td><td>Input dataミス</td></tr> <tr><td>08</td><td>ライブラリー使用法ミス</td></tr> <tr><td>09</td><td>引数の対応ミス</td></tr> <tr><td>10</td><td>初期値設定ミス</td></tr> <tr><td>11</td><td>未定義番号の使用</td></tr> <tr><td>12</td><td>ファイルの使用法ミス</td></tr> <tr><td>13</td><td>磁気テープの使用法ミス</td></tr> <tr><td>14</td><td>図形ジョブミス</td></tr> <tr><td>15</td><td>原因不明</td></tr> <tr><td>16</td><td>センター側のミス</td></tr> <tr><td>17</td><td>その他</td></tr> </table>		01	コントロール・カード・ミス	02	FORMAT文ミス	03	配列オーバー	04	宣言文ミス	05	論理ミス	06	パンチ・ミス	07	Input dataミス	08	ライブラリー使用法ミス	09	引数の対応ミス	10	初期値設定ミス	11	未定義番号の使用	12	ファイルの使用法ミス	13	磁気テープの使用法ミス	14	図形ジョブミス	15	原因不明	16	センター側のミス	17	その他
1	ジョブ始末結果について	01 FORTRAN(C)																																																																							
2	質問	02 FORTRAN-D																																																																							
3	(内容を記事欄に記入のこと)	03 FORTRAN-E																																																																							
3	システムIIについて	04 FASP																																																																							
4	その他	05 ALGOL																																																																							
		06 COBOL																																																																							
		07 PL/I																																																																							
		08 LIME																																																																							
		09 LIED																																																																							
		10 アプリケーション																																																																							
		11 ユティリティ																																																																							
		12 その他																																																																							
01	コントロール・カード・ミス																																																																								
02	FORMAT文ミス																																																																								
03	配列オーバー																																																																								
04	宣言文ミス																																																																								
05	論理ミス																																																																								
06	パンチ・ミス																																																																								
07	Input dataミス																																																																								
08	ライブラリー使用法ミス																																																																								
09	引数の対応ミス																																																																								
10	初期値設定ミス																																																																								
11	未定義番号の使用																																																																								
12	ファイルの使用法ミス																																																																								
13	磁気テープの使用法ミス																																																																								
14	図形ジョブミス																																																																								
15	原因不明																																																																								
16	センター側のミス																																																																								
17	その他																																																																								
<table border="1"> <tr><td>31</td><td>コントロール・カード</td></tr> <tr><td>32</td><td>FORTRAN全数について</td></tr> <tr><td>33</td><td>ライブラリーの使用方法</td></tr> <tr><td>34</td><td>ファイルの使用方法</td></tr> <tr><td>35</td><td>磁気テープの使用方法</td></tr> <tr><td>36</td><td>FORMAT文</td></tr> <tr><td>37</td><td>エラー見方・みつけ方</td></tr> <tr><td>38</td><td>図形出力について</td></tr> <tr><td>39</td><td>カードパンチ出力方法</td></tr> <tr><td>40</td><td>プログラムの書き方</td></tr> <tr><td>41</td><td>センターの利用方法</td></tr> <tr><td>42</td><td>ユティリティの使用法</td></tr> <tr><td>43</td><td>ジョブの見方</td></tr> <tr><td>44</td><td>その他</td></tr> </table>		31	コントロール・カード	32	FORTRAN全数について	33	ライブラリーの使用方法	34	ファイルの使用方法	35	磁気テープの使用方法	36	FORMAT文	37	エラー見方・みつけ方	38	図形出力について	39	カードパンチ出力方法	40	プログラムの書き方	41	センターの利用方法	42	ユティリティの使用法	43	ジョブの見方	44	その他																																												
31	コントロール・カード																																																																								
32	FORTRAN全数について																																																																								
33	ライブラリーの使用方法																																																																								
34	ファイルの使用方法																																																																								
35	磁気テープの使用方法																																																																								
36	FORMAT文																																																																								
37	エラー見方・みつけ方																																																																								
38	図形出力について																																																																								
39	カードパンチ出力方法																																																																								
40	プログラムの書き方																																																																								
41	センターの利用方法																																																																								
42	ユティリティの使用法																																																																								
43	ジョブの見方																																																																								
44	その他																																																																								

記事(現象・原因・処置等)についてなるべく詳しく書いてください。質問の場合はその内容を書いてください。  
(現象又は質問内容) (原因)

プログラム相談票様式

エラー原因項目テーブル例

大項目0	小項目1	小項目2	小項目3...9
A	コントロール・カード・ミス	¥RUNPCHがあるか	¥RUNがあるか
B	パンチ・ミス	Iと1と/のパンチ・ミス	Xと*のパンチ・ミス
C	FORMAT文ミス	カッコの数は正しいか	/の位置は正しいか
D	ライブラリー使用ミス	引数の意味を説明しているか	
E	宣言文ミス	EXTERNAL宣言しているか	DIMENSION宣言があるか
F	配列オーバー	DOの終端値は正しくセットされているか	終端値が正しく入っているか
G	READ/WRITE文ミス	読取を間違えているか (READ-5)	読取を間違えているか (WRITE-6,7)
H	INPUT DATAミス	データ枚数は正しいか	I4で「4」になる
I	初期値設定ミス	予定の値を入れているか	数値型 実数型の区別を失っているか
J	ファイルの使用法ミス	...	...
K	副プログラムの引数ミス	...	...
L	間違え	Iと1を間違えているか	RとIを間違えているか

エラー状況項目テーブル例

あ	変数を使っているか	イ	インプット・データがあるか
い	格精度を使っているか	け	文字型定数を用いて分枝しているか
う	サブルーチン・ファンクションを使っているか	こ	SSLを使用しているか
え	実行数にファンクション名、サブルーチン名があるか	せ	FORMATを配列に入れて使っているか
お	ファイルへの書式なし出力文を使っているか	し	構構指定は定数の変数か
か	右辺の式の要素に配列があるか	...	...
き	ファイルを使っているか	...	...