The Journal of National Chiao Tung University
Vol I, April 1976
pp. 41-56

印製中文表格之新語言

A New Language for Tabulation-CHITAL

蔡中川 Jong-Chuang Tsay

Department of Computer Science

(Received January 6, 1976)

ABSTRACT——This paper defines a new kind of language. The language is called CHITAL — CHInese table TAbulation Language. The main purpose of the CHITAL is used for describing a table including Chinese characters. The purpose of this language is so general that a fairly complicate table and even a simple graph can be described. In addition to defining the language, this paper describes the implementation of an interpreter for it and gives some illustrated results.

I. Introduction

Three main problems exist in the design of Chinese information processing system. They are (1) input of Chinese characters, (2) data processing of Chinese characters, and (3) output of Chinese characters. There are two alternative approaches to the third problem. The first method uses a graphic-display device to display Chinese characters. This device can be a storage display CRT (such as Tektronix 4010) or a non-storage display CRT (such as DEC Type 340). Using a graphic-display device, a hard copy can be obtained either by taking picture directly from the screen or by using a 'push to print' hard copy machine. Alternatively, a plotting device can be used to plot the Chinese characters. A suitable device to accomplish this job is the Versatec Matrix electrostatic plotter. The output quality of the electrostatic plotter is better than that of the graphic-display device. In this paper, a Versatec 1600A Printer/Plotter will be chosen as output device for Chinese characters. The Plotter has a nib density of 1600 dots with 160 dots per inch.

Usually, the final product of a Chinese information processing system is in the form of a table. The Chinese characters are to be arranged inside the table. Between these characters we may need horizontal or vertical lines of various width. There are two approaches to generate a table. The first approach uses a preprinted form of a table. In the preprinted form, frame of a table(constant parts of a table) is printed on the paper before it is installed in the plotter. The non-constant parts of the table can then be generated by the computer and filled into the table. The second approach uses computer to generate whole parts of a table. There are several drawbacks in the first approach. It is not easy to align the non-constant parts with constant parts of a table; different tables require different preprinted forms; the constant parts of a table can not be modified once it has been printed. Due to these drawbacks, the second approach will be chosen. The second approach avoides all the above drawbacks. It is flexible for the user to design various forms of table without the help of the printing company. Since it is not easy to describe a table by using a low-level language (such as assembly language) and an existing high-level language(such as FORTRAN), a high-level language is designed exclusively for describing a table. This language is called the CHITAL as mentioned in the Abstract. The design criteria for the CHITAL are (1)The language is easy to learn, (2) It is flexible enough to describe different forms of table, (3) A group of basic-format specifications can be repeated by enclosing the group in parentheses and preceding the left parenthesis with an unsigned integer denoting the number of repetitions (repetition count), (4) Repetition can be nested up to 50 levels.

During the development of Chinese information processing system at the National Chiao Tung University(NCTU), the author realized that it is not easy to use a conventional language (e.g., assembly language or FORTRAN language) to describe (or draw) a table including Chinese characters. In order to provide a solution to this problem, a tabulation language is designed. This tabulation language is similar, in purpose, to the FORMAT statement of the FORTRAN language. However, their corresponding output devices are different. The FORMAT statement is used to specify the output format on the printer, while the tabulation language is used to specify the output format of a table on a plotter. The interpreter of the tabulation language is implemented on the PDP-11/40 digital computer (currently in NCTU, it has two DEC tape drivers, 32K words memory, two RK05 disk drivers, one Versatec 1600A Printer/Plotter, and one Tektronix graphic display). In the following discussions, CHITAL will represent this new language and its corresponding interpreter will be denoted by ICHITAL.

In the next section, the syntax and semantics of the CHITAL will be described. Section III, which follows, will give a brief description on the implementation of ICHITAL. Several results will be shown in Section IV.

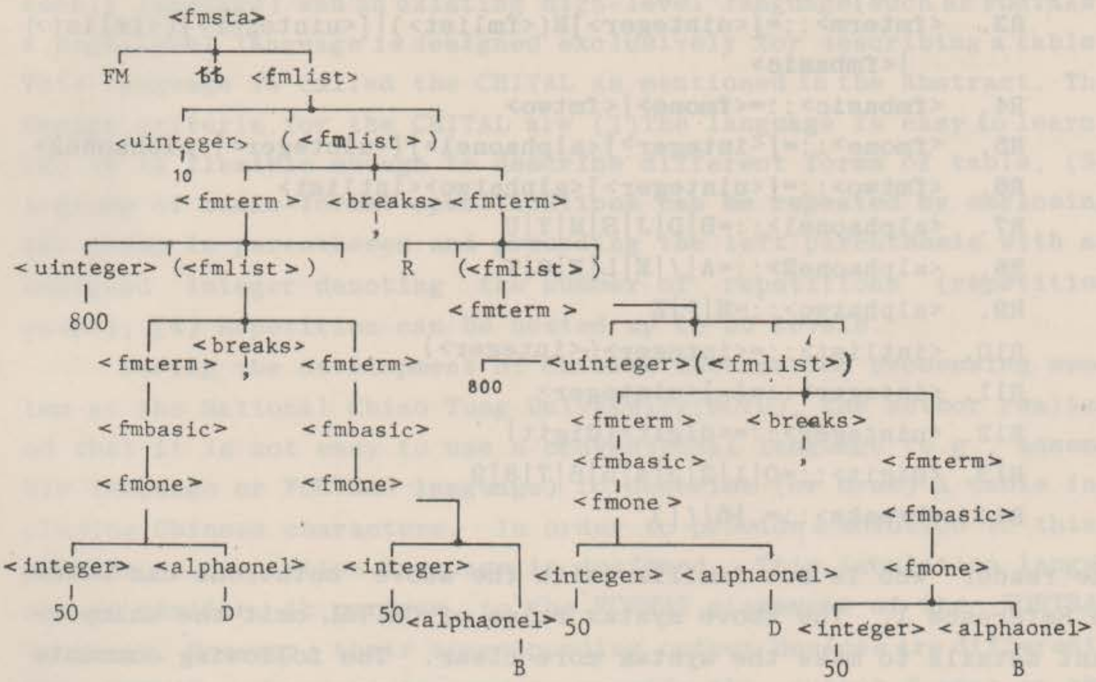
II. Syntax and Semantics of CHITAL

It is well known that the syntax of a language can be precisely defined by Backus-Naur Form (BNF). To define the CHITAL, a modified-BNF will be used [1]. This notation is as powerful as BNF. However, it is more readable than pure BNF. The syntax of CHITAL as defined by the modified-BNF is given as follows with <fmsta> as distinguished symbol.

```
R1.  <fmsta>::=FMb̸{b̸}<fmlist>
R2.  <fmlist>::=<fmlist>{<breaks><fmterm>}|<fmterm>
R3.  <fmterm>::=[<uinteger>]R(<fmlist>)|[<uinteger>](<fmlist>)
        |<fmbasic>
R4.  <fmbasic>::=<fmone>|<fmtwo>
R5.  <fmone>::=[<integer>]<alphaone1>|[<uinteger>]<alphaone2>
R6.  <fmtwo>::=[<uinteger>]<alphatwo><intlist>
R7.  <alphaone1>::=B|D|J|S|M|Y|U
R8.  <alphaone2>::=A|/|E|L|X|G|T
R9.  <alphatwo>::=H|P|Z
R10. <intlist>::=<integer>{•<integer>}
R11. <integer>::=[-]<uinteger>
R12. <uinteger>::=<digit>{digit}
R13. <digit>::=0|1|2|3|4|5|6|7|8|9
R14. <breaks>::=,|b̸|(|)
```

The reader who is not familiar with the above notations can refer to Reference 1. The above syntax rules of CHITAL omit the unimportant details to make the syntax more clear. The following comments on these details. There are length or value limitation on the sentences and subsentences of the language. For example, the value of an unsigned-integer should be less than $2^{15}-1$, the length of a sentence should be less than 720 characters. Another comment on the syntax is the number of integers in the syntactic entity <intlist> for an H-format should equal to the unsigned-integer preceding H. For a Z-format, the number of integers in the <intlist> should be equal to four. For a P-format it is equal to one.

Now, let us take a look at the CHITAL's syntax. Starting from the distinguished symbol <fmsta>, R1 says that the format-statement <fmsta> is defined as FM followed by at least one blank, then by format-list <fmlist>. R2 is a definition of the <fmlist>, it says that the <fmlist> is a string of format-terms <fmterm> separated by break-symbol <breaks>. Each <fmterm> as defined by R3 can be a basic-format <fmbasic> or a <fmlist> enclosed in parentheses preceding by an optional R and an unsigned-integer <uinteger>. R4 says that the <fmbasic> has two forms. The first form is iF where i is an integer and F is a terminal symbol derived from <alphaone1> or <alphaone2>. The second form is iFi.i.i.i --- where F is a terminal symbol derived from <alphatwo>. Rules R5-R13 define the above syntax. An example of a sentence of the CHITAL and its corresponding syntax tree is shown in Fig. 1.



FMbb10(800(50D,50B),R(800(50D,50B)))

Fig. 1. Syntax Tree of a Sentence

In order to understand the semantics of the CHITAL, functions of each type of the basic-format <fmbasic> will be described. Let us first define some important variables.

LINEB1,LINEB3,LINEB5-Each variable is a line buffer of 1600 bits.

X1—A pointer points to a bit of the line buffer. Its value belongs to the range [0, 1599].

VPBUF—A Versatec plotter buffer of 3.2k words (32 x 1600 bits). It is used to hold the 32 x 32 dot matrices of Chinese characters.

AMODE—A value of one indicates that it is currently in the character-plotting mode, while a value of zero indicates that it is in the line-plotting mode.

SBIT—A pointer points to a column of VPBUF. Its value belongs to the range [0, 1599].

JJJ—A pointer points to a specific location of the VPBUF or LINEBl. The specific locations is one of the following locations 0, j, 2j, 3j, ---, where j is equal to the size of the Chinese character in horizontal direction plus the horizontal space between Chinese characters. JJJ has a value belonging to the range [0, 1599].

INIT—A subroutine which initializes the variables Xl, SBIT, and JJJ. The initialized value of the variables Xl, SBIT, and JJJ are 0, LEADS, and 0 respectively. Where LEADS is equal to the horizontal space between Chinese characters. For a 32 x 32 Chinese character its value is 4.

i— An integer.

n— An unsigned integer.

VPLOT—A plotting subroutine which transfers the content of a line buffer or VPBUF to Versatec plotter. If AMODE=0 the line buffer is chosen, otherwise VPBUF is chosen. In the latter case, LINEBl is exclusive-ORed with each horizontal line of the VPBUF before output. VPBUF is cleared after it is output. Before leaving the subroutine, INIT is called.

Now, the main functions of each type of the basic-format and iR-format can be described as in the following.

iB—Let T=X2+i, where X2=Xl if AMODE=0, else X2=SBIT. If T>=1600 then moves zeros to LINEBl [X2, 1600), else moves zeros to LINEBl [X2,T). In the former case, VPLOT is called to output LINEBl or VPBUF, then 1600 is subtracted from T, then repeats from the above testing step. In the latter case, updates Xl =T if AMODE=0, else updates SBIT=T. Underflow of buffer (T<0) is not permitted.

iD—Same function as iB except ones is moved to LINEBl instead of zeros.

iJ—Updates JJJ as JJJ+ij, where j is defined in the definition of JJJ. If the updated value is less than zero, JJJ will be cleared. If it is greater than a prescribed limit (the limit is equal to 1599 if AMODE = 0, else equal to 1551), a modulo operation will be performed. After the above operations, JJJ is used to update X1 (if AMODE=0) or SBIT (if AMODE = 1). 0J has a special function of calling VPLOT if both AMODE=1 and there is data in VPBUF. AMODE is cleared after the operation.

iR—The current output-flag is complemented so that the following output of ones and zeros will be switched. The integer i is disregarded.

iS—If both AMODE=1 and there is data in VPBUF then outputs it. Clears AMODE. Let T = X1 + i. If T < 1600 then moves ones to LINEB3 [X1, T), sets X1=T. If T > = 1600 then moves ones to LINEB3 [X1, 1600); exclusive-ORs LINEB1 to LINEB3; calls VPLOT; clears LINEB3. 0S has a special function of calling VPLOT to output the exclusive-OR of LINEB1 and LINEB3.

iM—Same function as iB except LINEB1 is not modified.

iY—Calls VPLOT to perform the output of the exclusive-OR of LINEB1 and LINEB5. Disregards i.

iU—Performs advance of paper to the top of next page. Disregards i.

nA—Retrieves n Chinese character codes from A-file in sequence and its corresponding Chinese characters to VPBUF. Whenever overflows, calls VPLOT to output VPBUF and continues retrieving. In the above process, SBIT is updated.

n/—Calls VPLOT to perform the output of VPBUF n times (notice that after the first output of VPBUF, it is cleared). Vertical spacing between Chinese characters is automatically performed.

nE—It has the same function as n/ when AMODE=1. When AMODE=0, calls VPLOT to output LINEB1 and then skips paper n-1 lines.

nL—If there is any data in LINEB1 or VPBUF, calls VPLOT to output it according to the current AMODE. After this, calls VPLOT to draw n horizontal lines.

nX—Same function as nA except blank characters are moved to VPBUF.

nG--Same function as nA except the Chinese character codes are retrieved from G-file.

nT—Calls VPLOT to output LINEB1 n times.

$nHn^1.n^2.$ --- n^n— Same function as nA except the Chinese character codes are obtained from the codes n^i.

nPn^1--Sets the nth parameter with a value n^1. Some useful P-commands are as follows.

n=1 or 2, sets the size of the dot matrix of Chinese character.

n=5, controls a 90 degree rotation of Chinese characters.

n=6, controls the amplifications of Chinese characters.

n=7, control the reduction of Chinese characters.

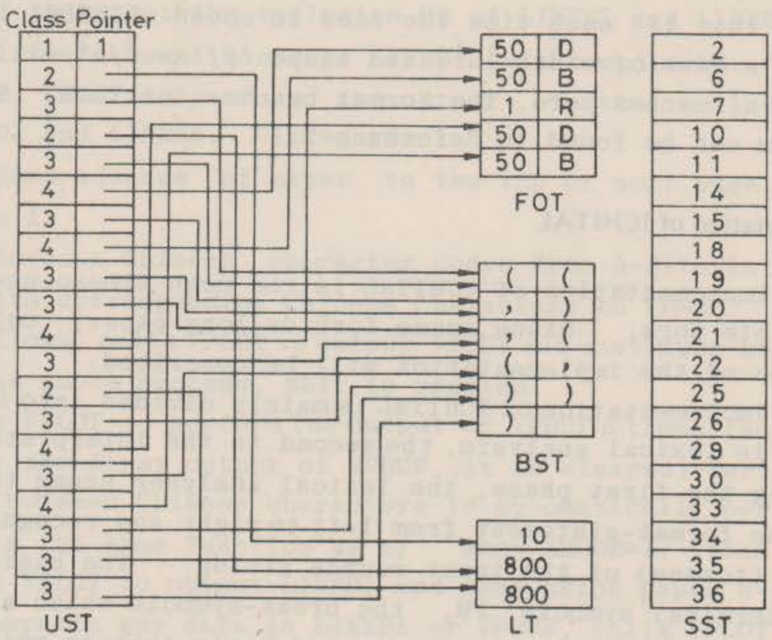
n=15, controls character font.

$nZu^1.u^2.u^3.n^1$--This is used to draw a slope line. Each time the format is scanned by the interpreter, n^1 is decremented by one. Whenever n^1 reaches zero, $|u^1|$ of ones are moved to LINEB5 [X1, X1+u^1). When the format is scanned again, n will be decremented by one and the ones in the LINEB5 will be moved to other position of LINEB5. The above algorithm will accomplish that if each time the ones is moved LINEB5 is output, then a line of width $|u^1|$ and slope $\Delta y/\Delta x=u^3/u^2$ will be drawn. When n reaches zero, the format becomes inactive, This algorithm can be found in Reference 2.

III. Implementation of ICHITAL

The implementation of ICHITAL is the most time-consuming part of the whole work. Since space forbids long paper, only a brief description of the implementation will be described.

The implementation of ICHITAL is mainly divided into two phases. The first is lexical analysis, the second is the interpretation phase. During the first phase, the lexical analyzer scans the characters of the format-statement from left to right and recognizes basic elements (tokens) of the input source string. The basic elements are the terminal symbols FM, the break-symbols which are derived from the syntactic entity <breaks>, and basic-formats which are derived from <fmbasic>. These basic elements are recognized and translated into an internal form suitable to be interpreted in the second phase. The following tables are used to represent the internal form of the translated source statement: Uniform Symbol Table (UST), Break Symbol Table (BST), Literal Table(LT), Format-One Table(FOT), and Souce String Table (SST). UST consists of a list of tokens as in the souce input statement. Whenever lexical analyzer scans a token (or symbol), it is entered into UST in sequence. Each entry of UST is of fixed size and consists of a syntactic class and a pointer pointing to other table entry of the associated token [3]. The main function of UST is to represent the souce input string in a uniform manner of the pointers of tokens. BST is used to store break symbols. Whenever a token of the type <breaks> is scanned, it is entered into BST. LT is used to store all the constants of <integer> type. FOT stores all the tokens of the terminal symbols which are derived from the syntactic type <fmone>. The SST plays a different role. It keeps pointers pointing to the original input string. During the interpretation phase, if the interpreter finds an error, the SST will be looked up to get the substring (of the original input string) where error has been detected. Futhermore, error message will be typed to inform the user where the error has been detected. Fig. 2 shows an example of the tables built by the lexical analyzer.



The Input String is

FMbb10(800(50D,50B),R(800(50D,50B)))

Fig. 2. Tables Constructed by the Lexical Analyzer

Once all the tables have been constructed, the second phase can be entered. The second phase is an interpretation phase. The interpreter scans the UST and looks up its related tables and then interprets or executes it. Whenever basic elements of the type

<fmbasic> (except R-format) has been read, it is executed according to the semantics of CHITAL as described in Section II. If the interpreter has read the input string n(or (or nR(or R(, three values will be pushed down into three stacks respectively. The first value is the repetition count as specified by n in n(; for the cases (or nR(or R(it is equal to 1. The second value is a pointer (rescanning pointer) which points to the UST where a pointer points to the break symbol (in LT. The third value is a output-flag whose function has been described in the semantics of iR format. When a matching right parenthesis has been found, the repetition count in the first stack will be decremented by one, and the rescanning pointer in the second stack will be retrieved to determine where to start interpretation again. Whenever the repetition count is decremented to zero the top elements of the three stacks are popped up. By this method part of the input string can be interpreted repeatedly and the repetition can be nested.

Whenever ICHITAL detects a syntactic or semantic error in the input string, an error message of the form

ERXX YYYYYYYYYYY

will be typed and the job is aborted. Where XX is the error code and Y's are part of the original input string. The ICHITAL has detected an error at a place to the left of sixth Y. Most error messages are designed for the interpretation phase. However, a few error messages are also provided for the first phase. Appendix 1 is a listing of the error messages.

IV. Results

To see the amazing power of CHITAL, several examples are shown here. All the results shown are directly taken from the Versatec plotter's output. From the semantics of CHITAL as explained in Section II, it is not difficult to see the relation between each sentence of the CHITAL and the plotter's output. Only some comments will be given here.

Fig. 3 is an output of the following sentence.

```
FM  10(800(50D,50B),R(800(50D,50B)))
```

Fig. 4 can be described by the following sentence.

```
FM  10X,21A,E,16X,5(A,X),0J,5(9J,828D,E),1600B,10E,6X,*
31A,E,5(4X,33A,E),4X,28A,2E,0J,*
10(40(18D,J),E),10E,1600B,14X,13H8494.8516.8514.*
8497.8510.8515.8508.8497.8536.8505.8511.8510.8495,E
```

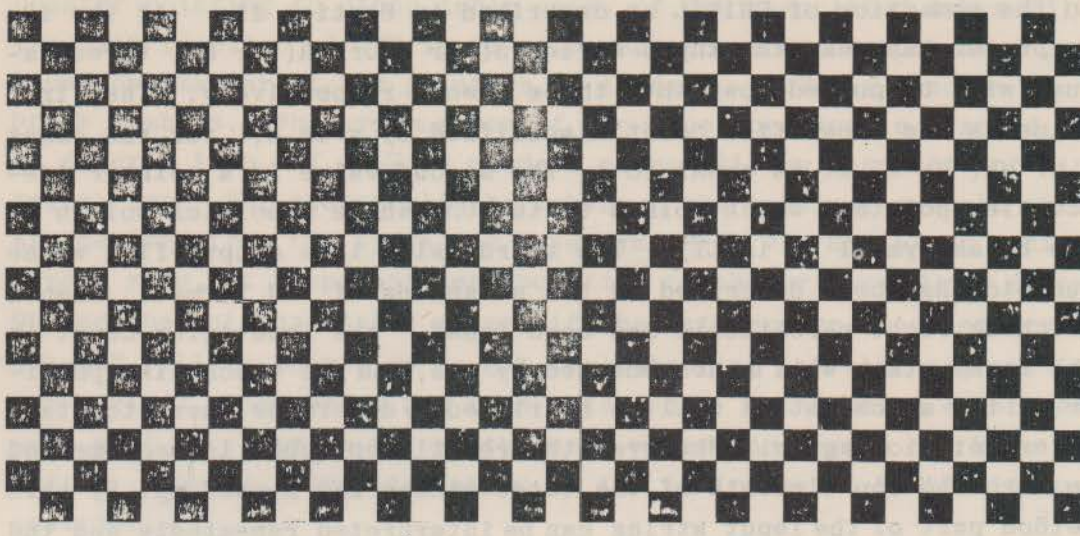


Fig. 3.

As can be seen from the sentence, the continuation of input string is done by appending the character * at the end of each lines. The lower part of the output starting from the word "Tourists" is not described by the above input string, but by other means.

WHY NOT STAY AT HOME?

在家千日好

觀光客多半是一夥表情陰鬱的人，我在葬禮時見到遠比聖馬可廣場上遊客開朗的臉色。實際上，旅行家眞正喜愛旅行的百不得一。如問他們何以要花錢旅行，與其說是出諸好奇，找樂子，或想見見珍奇的景物，倒不如說是出於一種附庸風雅的心理。人們旅行和蒐集藝術品，出於同一動機；因爲這些都是一流人士的玩藝。就社交而言，到過大地表面的若干地點自很不壞，而去過某某地方的人也較蟄居一地的人强。再者，旅行使人在還鄉後有話可談。可作談的話題既不多，人人自願抓住一次增廣見聞的機會。

(TRANSLATION)

Tourists are, in the main, a very gloomy-looking tribe. I have seen much brighter faces at a funeral than in the Piazza of St. Mark's. The fact is that very few travelers really like traveling. If they go to the trouble and expense of traveling, it is not so much from curiosity, for fun, or because they like to see things beautiful and strange, as out of a kind of snobbery. People travel for the same reason as they collect works of art; because the best people do it. To have been to certain spots on the earth's surface is socially correct; and having been there, one is superior to those who have not. Moreover, traveling gives one something to talk about when one gets home. The subjects of conversation are not so numerous that one can neglect an opportunity of adding to one's store.

Fig. 4.

Fig. 5 can be described by the following sentences.

```
FM  8L,8D,44J,8B,8D,29X,5H182.1056.1606.189.89,9X,H291,29X,*
5H2002.93.58.286.8488,3L,4J,3D,20J,3D,4J,3D,8J,3D,4J,3D,E,*
4H219.131.1033.146,3A,17X,4H1482.348.291.1813,8X,H219,2X,*
H244,E,3L,H218,2X,H1766,20X,4H164.2127.435.520,6X,-3B,3D,*
2H227.959,-48,4B,2H148.2202,3D,108B,E,3L,6H921.2202.53.959.*
128.33,4X,H53,4X,2H959.273,6X,H959,4H171.2202.480.421,396B,*
E,3L,4H2202.1057.286.852,16B,H60,16B,X,H2002,16B,X,H93,*
16B,X,H58,16B,X,H77,16B,X,H70,16B,X,H42,16B,X,H236,16B,X,*
24J,10B,H2202,10B,H1057,10B,H193,E,3L,4H171.2202.182.266,-4B,*
4B,H1718,*
5X,H269,5X,H3328,5X,H8572,180B,H1222,5X,H293,5X,2H708.176,E,*
8L,E

FM R(2X,36B,17H266.598.265.706.219.1606.462.168.57.480.1355.*
214.921.2202.258.106.204,E)

FM  R(79X,9H128.33.62.53.33.26.123.62.58),U
```

The double-size characters are described by the second sentence. Before the sentence is interpreted, the control parameter for the size of the characters are doubled. The half-scale Chinese characters are described by the third sentence. While the table between these two lines of output are described by the first sentence. The same result can be obtained by only one sentence when we using P format to control the size of the output characters. This can be seen from the following two examples.

台灣省汽車燃料使用費申請補繳通知書

新監燃字第 號
填發日期：

車主姓名	蔡中川	牌照號碼		車 別	
住 址		氣缸容量		每季應繳	
補繳年季	六十 年 季共 季	實繳費款			
繳納期限	自 填 發 日 起 五 天 內	繳納處			
實繳新台幣	萬 仟 佰 拾 元 角正				

Fig. 5.

Fig. 6 can be described by the following sentence.

```
FM 17X,11G,E,6P2,R(5X,12G,E),6P1,3G,2X,4G,2X,3G,3X,4G,2X,G,*
2X,2G,*
39J,3G,0J,3J,72S,6J,72S,33J,71S,0S,6E,8D,6J,2D,5J,2D,13J,2D,*
15J,*
2D,3J,2D,2J,8B,7D,*
8L,11X,2(2(G,5X),G,X),E,0J,11J,1008S,0S*
,G,4X,2G,3X,G,3X,-2B,2D,2(G,18B),G,-2B,2D,6X,2(G,3X),G,-2B,*
2D,6X,5G,E,0J,14J.144S,10J,324S,0S,*
```

```
0A,11J,3G,2X,-2B,2D,7J,6G,9J,-2B,2D,3J,-2B,2D,3J,9G,E,14J,4G,*
10J,9G,*
E,2L,G,-2B,2D,2G,*
X,5G,X,G,E,0J,2(6J,1600S,0A,6J,3G,X,G,E,0J),6J,1600S,G,5X,3G,*
X,G,E,*
0J,6J,1600S,0A,*
G,6J,G,3X,G,E,0J,J,1600S,A,2G,X,3G,3X,G,E,0J,J,1600S,0A,*
J,2G,X,3G,3X,G,E,*
4(0J,6J,1600S,G,5X,G,3X,G,E)*
,0J,6J,1600S,0A,6J,G,3X,G,E,0J,J,1600S,0A,2G,6J,-4B,4B,4X,G,*
E,8L
```

台灣省政府交通處公路局

代徵汽車燃料使用費月報表

單位： 區監理所 監理站 中華民國 年 月份 編號：____

年	度	季別	應徵數 查定數	增減數 本月	增減數 累計	本年度累計數	實徵數 本月份 徵收數	退還數	實徵數	本年度累計數	本年度未徵數	備註
汽	徵收 年度	第一季 秋										
		第二季 多										
		第三季 春										
車		第四季 夏										
×		小計										
汽	預徵 年度	全年										
	補徵 年度	全年										
油		全年										
×		全年										
部		全年										
		全年										
		全年										
份	合計											

Fig. 6

Fig. 7 is a chess-board which can be described by the input string.

```
FM 5(9(4J,8S),0S),9(4J,2D),-36J,*
2(2(4J,1152S,0S),144(16J,288Z2.1.1.0,8J,*
288Z2.-1.1.0,Y)),5(8J,8S,24J,8S,0S),2(4J,1152S,0S),141T,*
2(4J,1152S,0S),141T,5(4J,8S,4(8J,8S),0S),2(4J,1152S,0S),141T,*
10(*
4J,1152S,0S),1600B,6P2,15P2,5P1,*
0A,2J,-4B,2D,2J,7(A,X),-4B,2D,E,6P1,15P1,5P0,0J,1600B,9(4J,2D),*
-36J,*
10(4J,1152S,0S),141T,2(4J,1152S,0S),*
5(4J,8S,4(8J,8S),0S),141T,2(4J,1152S,0S),*
141T,2(4J,1152S,0S),5(8J,8S,24J,8S,0S),2(144(16J,*
```

```
288Z2.1.1.0,8J,288Z2.-1.1.0,Y),2(4J,1152S,0S)),*
5(9(4J,8S),0S),U
```

Two points should be mentioned here. Firstly, slope lines are shown on the output which are controlled by the Z formats. Secondly, 6P2, 15P2, and 5P1 are used to amplify the Chinese character, set the character font, and perform a 90 degree rotation respectively.

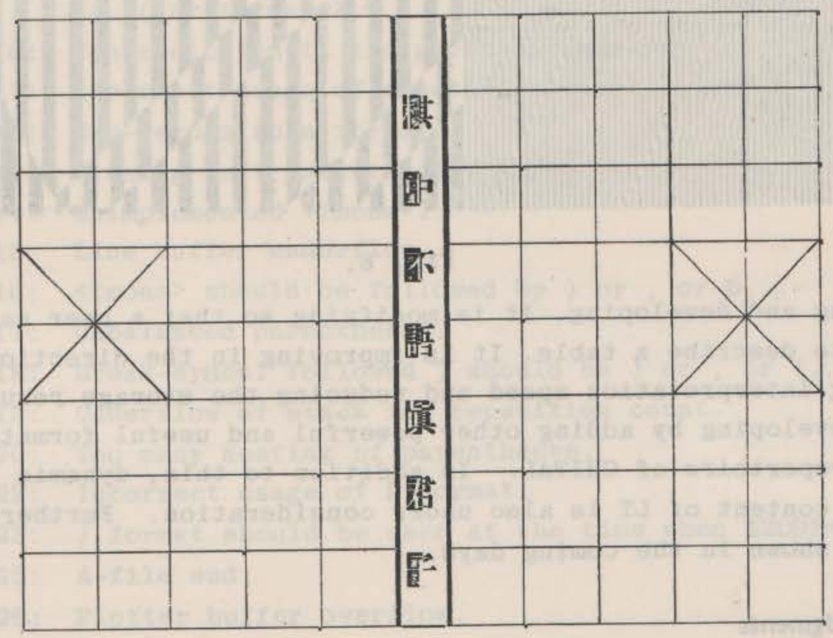


Fig. 7.

Fig. 8 shows an interested result which is described by

```
FM  160(5D,5B),900(5J,900Z100.1.1.0,450Z150.0.1.200,10J,*
800Z100.-1.1.0,*
700Z100.1.1.0,10J,500Z150.-10.1.250,600Z100.-1.1.0,-15J,*
450Z200.1.2.450,10J,17(20D,20B),20J,450Z200.-1.2.450,Y),*
1600B,360X,U
```

V. Conclusion

CHITAL is a new language which is designed to meet the general requirements of plotting tables including Chinese characters. As can be seen from the examples shown in Section VI, it is a general-purpose language capable of describing various kinds of table. CHITAL and ICHITAL are by no means complete. It is still under modifying,

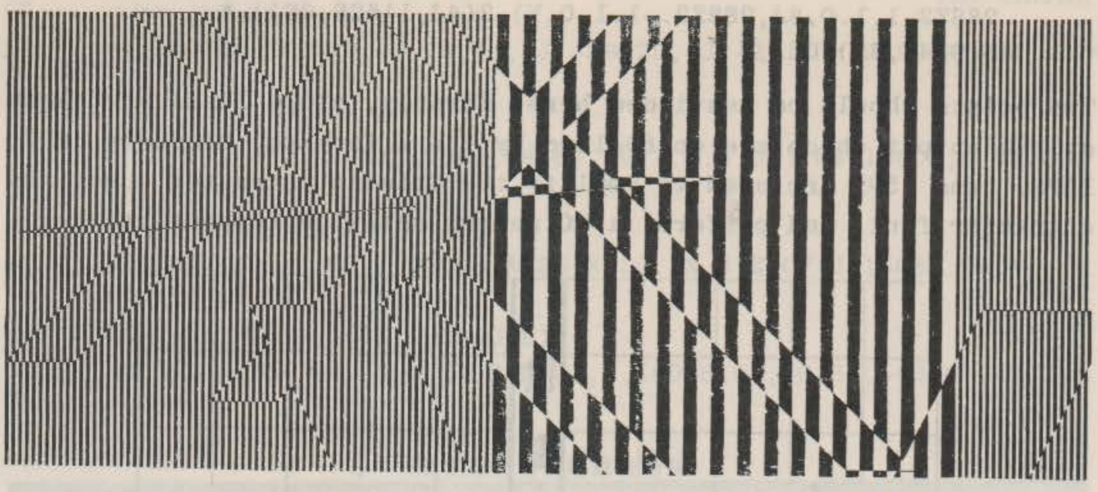


Fig. 8.

improving and developing. It is modifying so that a user can use it easily to describe a table. It is improving in the direction of increasing interpretation speed and reducing the storage requirement. It is developing by adding other powerful and useful formats to the format repertoire of CHITAL. In addition to this, dynamic modifying the content of LT is also under consideration. Further results will be shown in the coming days.

Acknowledgments

First of all, the author wishes to express his appreciation to Professor T. Y. Lee of Tsing Hua University for lending us the data file of the dot-matrices of Chinese characters. For the computer support of the research. The author wishes to express his gratitude to TEDPC and Professor C.T. Chang of NCTU. I am indebted to instructor S.M.Tsai of NCTU for his help to write a subroutine (transferring dot-matrix of Chinese character to plotter buffer). Finally, I would like to mention the operators of the computer center of NCTU, Mr. E. D. Jou of Tsing Hua University, NCTU graduate students, Mr. Y. J. Chen, H. S. Liang, colleagues and many unknown heros.

Appendix 1

ER01: Length of input string is too long.
ER02: 'FM' missing.
ER03: 'FM' should be followed by at least one blank.

ER04: Unimplemented format.

ER05: <fmone> followed by character other than <breaks>.

ER06: Illegal format of <fmtwo>.

ER07: <integer> missing before period.

ER08: Number of <integer> follows H,P,Z is not correct.

ER09: Consecutive <breaks> other than), or ,(or)) or ((or)(or <breaks> ƀ.

ER10: Contradiction in the given parameters.

ER11: Incorrect usage of repetition count.

ER12: Non-recogniable format.

ER13: Missing left parenthesis.

ER14: Unimplemented <fmone>.

ER15: Line buffer underflow.

ER16: <fmone> should be followed by) or , or ƀ.

ER17: Unbalanced parentheses.

ER18: Break-symbol followed) should be) or , or (.

ER19: Underflow of stack for repetition count.

ER20: Too many nesting of parentheses.

ER22: Incorrect usage of R format.

ER23: / format should be used at the time when AMODE=1.

ER25: A-file end.

ER26: Plotter buffer overflow.

ER27: Non-exist code.

ER28: Table Overflow.

ER29: Incorrect usage of P format.

ER30: Unimplemented character font.

ER31: G-file end.

References

1. David Gries, Compiler Construction for Digital Computers, John Wiley and Sons, Inc. New York, 1971.
2. John B. Peatman, The Design of Digital Systems, McGraw-Hill Book Company, 1972.
3. John J. Donovan, Systems Programming, McGraw-Hill Book Company, 1972.