

TR-88-20

AUTOMATIC EVALUATION ON RADICAL-  
BASED CHINESE INPUT METHODS

中研院資訊所圖書室



3 0330 03 000093 4

0093

# Automatic Evaluation on Radical-Based Chinese Input Methods

K. C. Cheng, S. Y. Hsu, M. S. Hwu and K. Y. Cheng

Department of Computer Science and Information Engineering  
National Taiwan University, Taipei, Taiwan, R. O. C.

and

Institute of Information Science, Academia Sinica  
Taipei, Taiwan, R. O. C.

## ABSTRACT

A theoretical basis for the evaluation of radical-based Chinese input methods is discussed. A radical-based Chinese input method chooses certain radicals as primitives so that an operator may easily follow the rules to decompose and encode the Chinese character into a string of external codes which identify the character. A decomposition graph which describes the morphological decomposition process of a Chinese character may be generated by a transformer. The evaluation procedures are based on the parsing of an operator's visual routines which are expressed in the decomposition graphs of Chinese characters. In this paper, we show that a decomposition graph can be converted into a derivation diagram from which a set of production rules of a context-free web grammar can be derived. We also show that the encoding of a Chinese character and the computation of evaluation parameters of each Chinese character can be performed by evaluating the attributes of an attributed context-free web grammar.

## I. INTRODUCTION

A Radical-Based Chinese Input (RBCI) method chooses certain radicals as primitives so that an operator may easily follow the rules to decompose and encode the Chinese character into a string of external codes (keys depressed) which identify the character. The efficiency of a RBCI method depends on how an operator parsing his/her visual routines [17] in the decomposition of Chinese characters and the operator's skill which will vary from one operator to another. Suppose we don't care about the skill, instead, we assume there is only one operator who just follows the steps produced by parsing his/her visual routines. The evaluation under such an assumption is method-oriented, because the factors concerning operator's behavior are excluded.

In this paper, we are interested in the derivation of the production rules of a context-free web grammar from parsing an operator's visual routines. For each production in a context-free web grammar, the semantic rules are specified which define all the synthesized attributes (such as the string of external codes and its string length) of the nonterminal on the left-hand side of the production and all the inherited attributes (such as the limit of the string length) of the terminal and nonterminal on the right-hand side of the production [3,14]. The encoding process of a RBCI method is through the applications of the semantic rules of the productions mentioned above. Also, the evaluation parameters of each Chinese character is computed by evaluating the synthesized attributes (such the decomposition speed of this character) of an attributed grammar. Our purpose here is to present a theoretical basis for a machine which by itself can do the evaluation analysis automatically from the input of dot-matrix Chinese character patterns. With this, a designer can define his/her RBCI method to this machine and obtain the evaluation result for further improvement of his/her method.

In the following, we shall divide our discussion into sections. Section II

introduces the RBCI methods and gives them a formal definition. Section III discuss the graph representation (decomposition graph) of the decomposition of a Chinese character under a particular method. Section IV and section V show the encoding process of this method and the computation of evaluation parameters through the attributes evaluation of an attributed context-free web grammar. Section VI shows that the evaluator is a machine which accept the Chinese character patterns as its input and obtain the evaluation result as its output.

## II. RADICAL-BASED CHINESE INPUT METHODS

A RBCI method specifically means the one that uses a set of radicals as the external codes in the morphological decomposition of Chinese characters [13]. Among them, Tsang-Chi, Three-corner, and Simplest methods are the most often mentioned [212,21]. The process to generate a string of external code symbols of a Chinese character in a particular method is guided by the decomposition rules and the encoding rules of this method. The decomposition rules, which are normally an order of spatial relations in a square block and its exceptions, specify how to decompose each Chinese character into a string of ordered radicals. The encoding rules, which are normally a proper selection of these ordered radicals, map each Chinese character into a string of external code symbols (input string) which in turn identify this character. For example, in Three-Corner method, the decomposition rules are based on a Z scanning order, i.e., from upper-left to upper-right then to lower-left then to lower-right. So the character 矮 can be decomposed to the string 人禾大女. The principle of encoding rules in this method is : the first three radicals for each character in the same order with the decomposition rules will be selected. So the string of radicals 人禾大 is selected to identify the character 矮. Similarly, in Tsang-Chi method, the normal order of the decomposition rules is based on the writing behavior of Chinese characters, i.e., from up-to-down then left-to-right then outside-to-inside. So 腦 can be decomposed into the string 月 ‘ ‘ ‘

☒ . The string 月 ‹ ‹ ☒ is selected by applying the encoding rules [21] to identify the character 腦 .

In some RBCI methods, there are many radicals associated with the same set of keys depressed. These radicals belong to the same radical family with same meaning or similar morphology. Each radical family consists of one primary radical and several parasitical radicals of the primary radical. The primary radical is used as the external code of those radicals in the same family. For example, in Tsang-Chi method 手 and 扌 have the same meaning, 女 and 㚹, 田 and 𠂇 have the similar morphology respectively. So the input string of the character 腦 is 月 女 女 田. The relation between keys depressed and the radicals in a family is a mapping from primary radical to a set of keys. For example there are 24 radical families in Tsang-Chi method and each primary radical maps to a key on keyboard. Key 'B' is used to represent the radical 月, key 'V' is used to represent the radical 女 and key 'W' is used to represent the radical 田. So the string of external codes of the character 腦 is B V V W. In Three-Corner method, there are 99 radical families and each primary radical maps to a two-digit number. For example, 人 maps to 83, 禾 maps to 29 and 大 maps to 42. So the string of external codes of character 矮 is 83 29 42. Under the consideration mentioned above, a RBCI method can be formally defined as follows :

[Definition 2.1] A RBCI method is a 3-tuple

$$\text{RBCIM} = (R, P, M_k).$$

$$R = \{ r_{ij} \}, \quad i = 1, 2, \dots, k, \quad j = 0, 1, \dots, n_i,$$

$$P = \{ P_d, P_e \}, \text{ and}$$

$$M_k(r_{ij}) = \{ \text{keys depressed} \},$$

where  $R$  is a set of radicals ( $r_{ij}$ )

which can be partitioned into  $k$  radical families,

$r_{i0}$  is the primary radical of the  $i$ -th family,

and  $r_{i1}, r_{i2}, \dots, r_{ini}$  are parasitical radicals of  $r_{i0}$ ,

$P$  is a set of rules which consist of a decomposition rule set  $P_d$  and an encoding rule set  $P_e$ , and

$M_k$  is a mapping from a primary radical to a set of keys depressed.

In using a RBCI method, an operator has to memorize a table of the external code symbols as well as to be trained to have the capability in creating some visual routines to extract the string of external codes to identify a character. Thus the method-oriented evaluation approach regards the action of Chinese character input of an operator as visual routines and the evaluation procedures are based on the parsing of these routines. In the following sections, we shall use Tsang-Chi method as an illustrated example to discuss the decomposition and encoding process of Chinese characters as well as the evaluation procedures from theoretical point of view.

### III. DECOMPOSITION GRAPH AND CONTEXT-FREE WEB GRAMMAR

In actual operation, any Chinese character input to an operator's visual mind can be regarded as a sensory icon being processed by a perception mechanism to form a decomposition graph as a recognition of the visual routines concerning a particular method. Although different methods produce different decomposition graphs for each Chinese character, but all of the graphs can be defined uniquely as  $G_d = (V, E)$ , where  $V$  is the set of nodes which represents the code symbols and  $E$  is the set of edges which represents the part-of predicate and the spatial relations between code symbols. Fig. 4.1 shows the decomposition graph of the character 腦 in Tsang-Chi method, where the

part-of predicate edge is shown in an arrowed dash-line and the spatial relation edge is shown in an arrowed solid-line. The visited terminal nodes in a counterclockwise traverse of the graph are the primitive code symbols corresponding to the radicals 月, 丿, ㇇, ㇆, ㇇, ㇇, 囧.

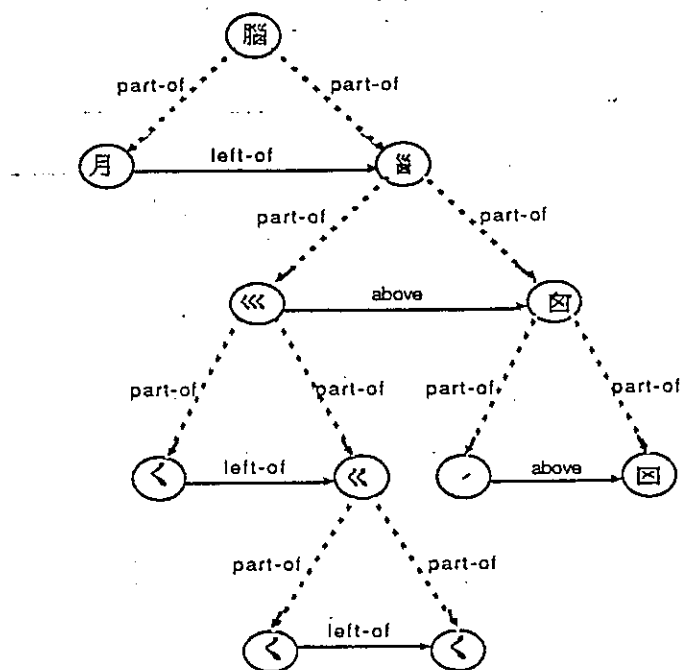


Fig. 3.1 The decomposition graph of the character 腦.

In order to understand the formation of a Chinese character, we need to distinguish the part-of predicate from the spatial relations, and so we introduce a new graph called the derivation diagram as follows. Basically, a derivation diagram is obtained from a decomposition graph by preserving the inherited relationship of the spatial relations from an ancestor node to all of its descendant nodes. However, the readers are encouraged to consult [7] for a detailed procedure. Fig. 3.2 shows the derivation diagram of the character 腦.

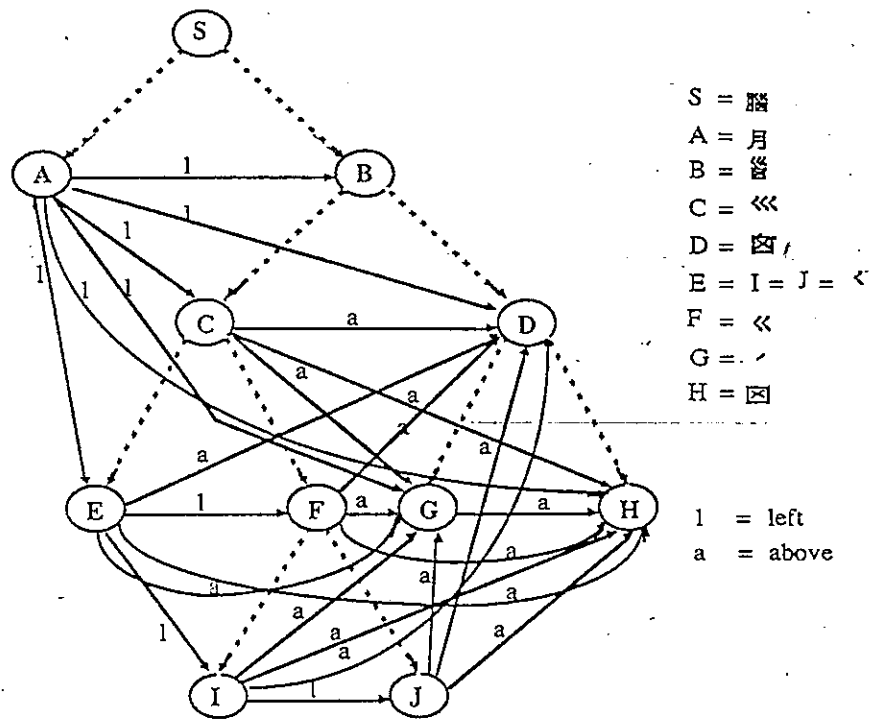


Fig. 3.2 The derivation diagram of the character 腦 .

From formal language theoretical point of view, a derivation diagram is a web for a context-free web grammar [6,9]. The definition of a web grammar is as follows

$$G_W = (V_N, V_T, P, S)$$

where  $V_N$  is a set of nonterminals,  $V_T$  is a set of terminals,  $S$  is a set of initial webs, and  $P$  is a set of web production rules. A web production rule is defined as

$$\alpha \rightarrow \beta, \gamma$$

where  $\alpha$  and  $\beta$  are webs and  $\gamma$  is an embedding of  $\beta$ . In order to substitute a subweb  $\alpha$  of a web  $\omega$  by another subweb  $\beta$ , we have to specify how to embed  $\beta$  in  $\omega$  in place of  $\alpha$ . For example, the web grammar that generates the web 腦 is



$V_N = \{ \text{腦}, \text{齒}, \text{囟}, \text{《《}, \text{《} \}$ ,  $V_T = \{ \text{月}, \text{‘}, \text{’}, \text{囟} \}$ ,  $S = \{ \text{腦} \}$ ,  
 and  $P =$  production rules shown in Fig.3.3.

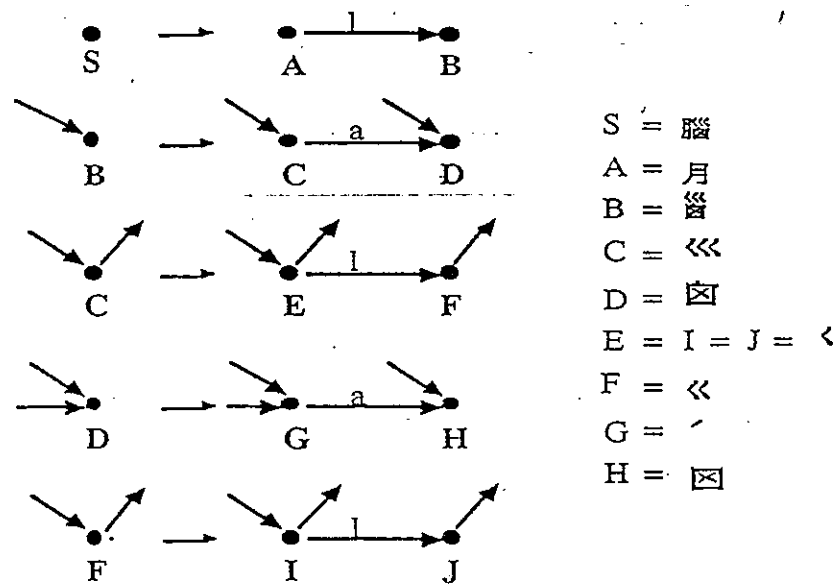


Fig. 3.3. The web production rules that generate the character 腦 .

To derive web production rules from a derivation diagram, we shall consider two graphs, one is called the skeleton and the other is called the section. A skeleton is a subdiagram obtained from a given derivation diagram by removing all of its spatial relation edges and leaving only part-of predicate edges and their connecting nodes. For example, Fig. 3.4 shows a skeleton of the diagram of Fig. 3.2. Notice that the skeleton is a tree structure whose root or subroot corresponds to a web  $\alpha$  and its decedents to another web  $\beta$  and thus a production  $\alpha \rightarrow \beta$  where  $\alpha \in V_N$  and  $\beta \in (V_N \cup V_T)$  is obtained. The grammar thus obtained is context-free because of the nature of tree structure.

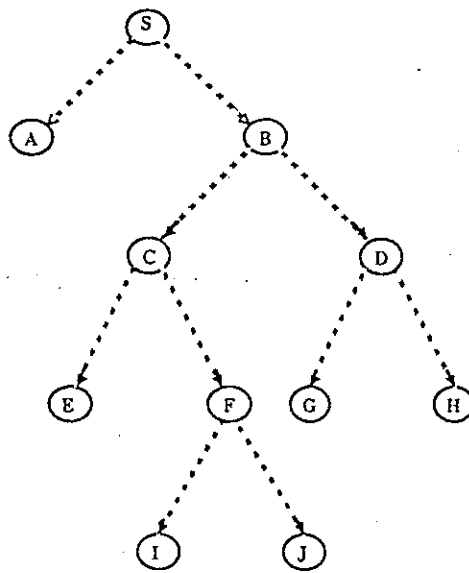


Fig. 3.4 The skeleton of fig. 3.2.

A section is a subdiagram that corresponds to a sentential form of the language generated by the web grammar [6]. We are only interested in the terminal section that shows the external code symbols in order. For example, Fig. 3.5 shows the terminal section of Fig. 3.2.

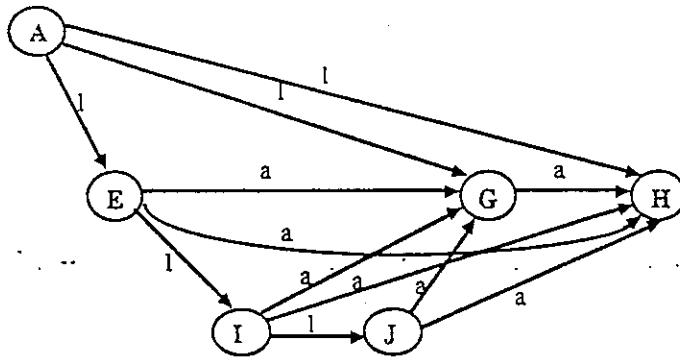


Fig. 3.5 The terminal section of fig. 3.2.

To obtain the web grammar from a given derivation diagram, a skeleton is first obtained to derive the production rules with  $\alpha \rightarrow \beta$  where  $\alpha \in V_N$  and  $\beta \in (V_N \cup V_T)$  and then a section is obtained to describe the embedding relationship of a production rule [7,19]. Notice that the number of production

rules with  $\alpha \rightarrow \beta$  where  $\alpha \in V_N$  and  $\beta \in (V_N \cup V_T)$  equals to the number of internal nodes in the skeleton. The higher the skeleton, the lower the decomposition speed. Also notice that the nodes in terminal section are a string of external code symbols in  $V_T$ .

#### IV. ENCODING PROCESS AND ATTRIBUTE EVALUATION

A good RBCI method should provide an operator some easy-to-use decomposition rules for decomposing Chinese characters fastly and appropriate encoding rules for selecting shorter external codes to decrease the average number of keys depressed. In order to get a shorter input string of a Chinese character, the external code symbols of some terminal nodes in a decomposition graph will be selected and the other will not. Since the string of external codes of each node in a decomposition graph is obtained by the concatenation of its son nodes' external codes, the input string of a Chinese character can be synthesized by the external codes of each terminal node. Here, we use the string of external codes (Exc) and its length (Len) as two synthesized attributes to obtain the input string and its length from its son nodes respectively. In order to decrease the number of keys depressed in the Chinese character input, we shall limit the number of external codes in each node of decomposition graphs. We use the limitation of number of codes (Lmt) in each node as an inherited attribute to inherit the Limit of number of external codes from its father node.

In the following, we shall define the semantic rules of an attributed context-free web grammar according to the encoding rules of Tsang-Chi method [21]. Where,  $S \rightarrow A B$  is the production which disarding the embedding of the production  $S \rightarrow A B, \gamma$ .  $S$  is the starting nonterminal which can be decomposed into  $A$  and  $B$ ,  $A, B \in (V_N \cup V_T)$ .  $A \rightarrow B C$  means that  $A$  can be decomposed into  $B$  and  $C$ , where  $A \in V_N, B, C \in (V_N \cup V_T)$ .  $\oplus$  is the

concatenate operator and  $\lambda$  is the null string. The encoding rules can be defined as :

a. IF  $S \rightarrow AB$

if A and B are two disconnected component then

$$\text{Exc}(S) = \text{Exc}(A) \oplus \text{Exc}(B)$$

$$\text{Len}(S) = \text{Len}(A) + \text{Len}(B)$$

$$\text{Lmt}(A) = 2$$

$$\text{Lmt}(B) = 3$$

else

$$\text{Exc}(S) = \text{Exc}(A) \oplus \text{Exc}(B)$$

$$\text{Len}(S) = \text{Len}(A) + \text{Len}(B)$$

$$\text{Lmt}(A) = 3$$

$$\text{Lmt}(B) = 4 - \text{Len}(A)$$

b. IF  $A \rightarrow BC$

if  $\text{Lmt}(A) \geq 2$  then

$$\text{Exc}(A) = \text{Exc}(B) \oplus \text{Exc}(C)$$

$$\text{Len}(A) = \text{Len}(B) + \text{Len}(C)$$

$$\text{Lmt}(B) = \text{Lmt}(A) - 1$$

$$\text{Lmt}(C) = \text{Lmt}(A) - \text{Len}(B)$$

if  $\text{Lmt}(A) = 1$  and the relation between B and C is surrounded then

$$\text{Exc}(A) = \text{Exc}(B) \oplus \text{Exc}(C)$$

$$\text{Len}(A) = \text{Len}(B) + \text{Len}(C)$$

$$\text{Lmt}(B) = \text{Lmt}(A)$$

$$\text{Lmt}(C) = \text{Lmt}(A) - \text{Len}(B)$$

if  $\text{Lmt}(A) = 1$  and the relation between B and C is not surrounded then

$$\text{Exc}(A) = \text{Exc}(B) \oplus \text{Exc}(C)$$

$$\text{Len}(A) = \text{Len}(B) + \text{Len}(C)$$

$$\text{Lmt}(C) = \text{Lmt}(A)$$

$$\text{Lmt}(B) = \text{Lmt}(A) - \text{Len}(C)$$

c. IF  $X \in V_T$

if  $\text{Lmt}(X) > 0$  then

$$\text{Exc}(X) = \text{Mk}(X)$$

$$\text{Len}(X) = 1$$

else

$$\text{Exc}(X) = \lambda$$

$$\text{Len}(X) = 0$$

Fig. 4.1 shows the attributed context-free web grammar that generates the character 腦 with semantic rules associated with each production. Fig. 4.2 shows the skeleton of the derivation diagram of Fig. 4.1 with the semantic information at each node. After the application of the semantic rules associated with each production, the evaluated attributes at the root node constitute the "meaning" of the derivation diagram of the character 腦, i.e., the input string 月女女田 and its length 4.

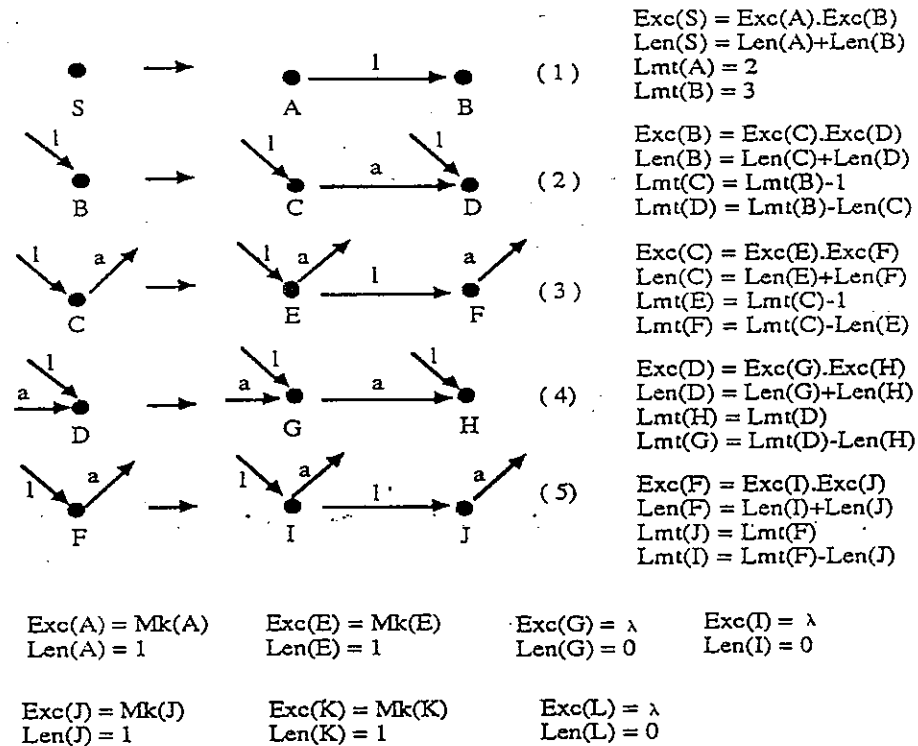


Fig. 4.1 The attributed context-free web grammar that generates the character 腦.

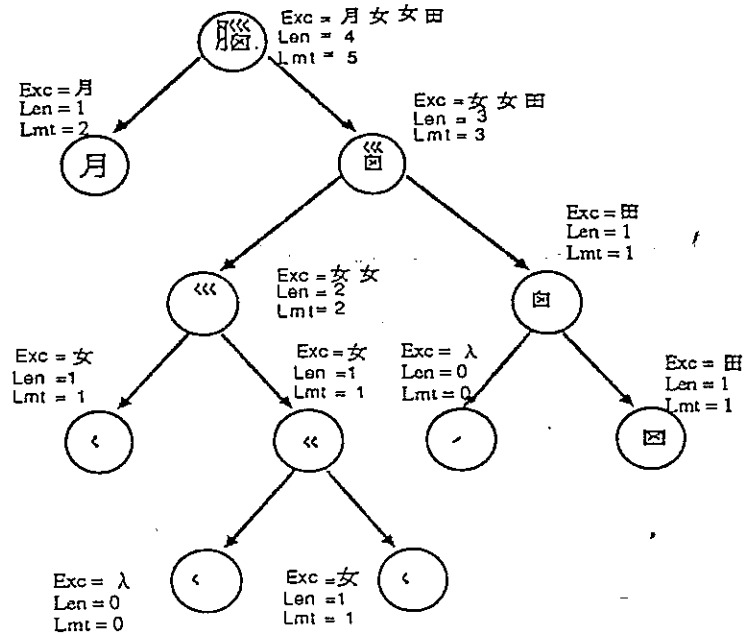


Fig. 4.2 The skeleton of the derivation diagram of Fig. 4.1 with the semantic information.

## V. COMPUTATION OF EVALUATION PARAMETERS

The efficiency of RBCI methods can be measured by several evaluation parameters such as the number of radicals, the number of decomposition rules and the number of encoding rules, the collision rate  $R_c(M)$ , the average keys depressed  $A_k(M)$ , the decomposition speed  $S_d(M)$ , the input speed  $S_i(M)$ , ..., etc. of the input method  $M$  [2,18]. The first three parameters can be obtained directly when an input method is defined.  $R_c(M)$ ,  $A_k(M)$ ,  $S_d(M)$  and  $S_i(M)$  will be computed step by step during the evaluation process is progressing. These parameters concern the occurrence frequency of each character, the frequency of the radicals and the frequency of the decomposition rules used. In the following, we shall discuss the computation of these four parameters.

Let  $C = \{ c_i \}, i = 1, \dots, N$  be the collection of Chinese characters. Then, the collision rate of  $M$  is computed as follows

$$Rc(M) = \frac{1}{n} \sum_{i=1}^N C(c_i)$$

where  $C(c_i)$  is the number of other characters whose external codes are exactly the same as that of  $c_i$ . The average number of keys depressed per character can be computed as follows

$$A_k(c_i) = \text{Len}(c_i) \cdot |M_k(r)| \cdot f_c(c_i)$$

where  $|M_k(r)|$  is the number of keys depressed which is mapped by a radical  $r$  defined in  $M$ . So the average number of keys depressed of  $M$  is

$$A_k(M) = \frac{1}{N} \sum_{i=1}^N A_k(c_i)$$

As stated previously, decomposing each Chinese character into a sequence of external code symbols is through applying some visual routines. Here a routine specifically means one application of a rule. Thus, the decomposition speed depends on the number of rules applied. The more rules are applied, the slower the decomposition speed is reached. Also, the frequency of each rule in use is a factor that affects the decomposition speed. The more often the applied rules are used, the easier the decomposition is reached. The measurement of decomposition speed may be expressed in terms of fuzzy logic [4,11].

Let  $U$  be the universal set of  $R$  or  $U = 2^R$ , then the Chinese character set  $C$  is a proper subset of  $U$ . Let  $\mu(c_i)/c_i$  denote that each  $c_i$  in  $C$  is assigned with a function  $\mu(c_i)$ , called the membership function of  $c_i$ . We use this membership function to indicate its decomposition speed. Then the set of Chinese characters under consideration is a fuzzy subset of  $U$

$$C = \sum_{i=1}^N \mu(c_i)/c_i$$

where  $\mu$  is a mapping

$$\mu : U \rightarrow [0,1].$$

In order to calculate the values of  $\mu$  at each node, we define the frequency of a radical  $r$  in use as follows

$$f_r(r) = \frac{\sum_{i=1}^N \sum_{j=1}^{nc_i} f_c(c_i) \cdot x_j}{\sum_{i=1}^N f_c(c_i) \cdot nc_i}$$

where  $f_c(c_i)$  is the occurrence frequency of Chinese character  $c_i$ ,  $nc_i$  is the number of external codes of  $c_i$  and  $x_j$  is an indicator whose value is 1 if  $r \in \text{Exc}(c_i)$  otherwise is 0. The value of membership function at the terminal node of the decomposition graph is the frequency of its corresponding radical. The decomposition speed at each nonterminal node depends on the frequency of the decomposition rule used in this node and its son nodes' decomposition speed. Such a membership function is defined as follows.

$$\mu(s_i) = \begin{cases} f_r(s_i), & s_i \in R \\ \left[ \prod_{j=1}^m \mu(s_j) \right] \cdot f_p(p_{dk}), & \begin{array}{l} s_j \in \text{child}(s_i), j=1, \dots, m \text{ and} \\ p_{dk} \text{ is the decomposition rule used in} \\ \text{the decomposition of } s_i \text{ into } s_1, \dots, s_m. \end{array} \end{cases}$$



where  $m$  is the number of son nodes of  $s_i$  and  $f_p(p_{dk})$  is the frequency of the  $k$ -th decomposition rule  $p_{dk}$  used.

From the above discussion, the decomposition speed of a Chinese character  $c_i$  can be obtained as

$$S_d(c_i) = \mu(c_i).$$

So the decomposition speed of  $M$  is

$$S_d(M) = \frac{1}{N} \sum_{i=1}^N S_d(c_i)$$

For each Chinese character  $c_i$ , the faster the decomposition speed, the faster the input speed. However the longer the external codes, the slower the input speed. The input speed of  $c_i$  can be computed as

$$S_i(c_i) = \frac{S_d(c_i)}{nc_i}$$

Then the input speed of  $M$  is

$$S_i(M) = \frac{1}{N} \sum_{j=1}^N S_i(c_j)$$

As we showed in the computation of the decomposition speed per character, the decomposition speed of each node is obtained from the decomposition speed of its sons and the frequency of the rule applied. Thus the computation process of decomposition speed is synthesized from terminal nodes to root node. So semantic rules are added to the context-free web grammar that generates a

Chinese character by associating a synthesized attribute  $\mu(s)$  to each node  $s$  in the derivation diagram of this grammar. The evaluation parameters  $S_d(c_i)$ ,  $S_i(c_i)$  and  $A_k(c_i)$  can be computed as shown in Fig.5.1.

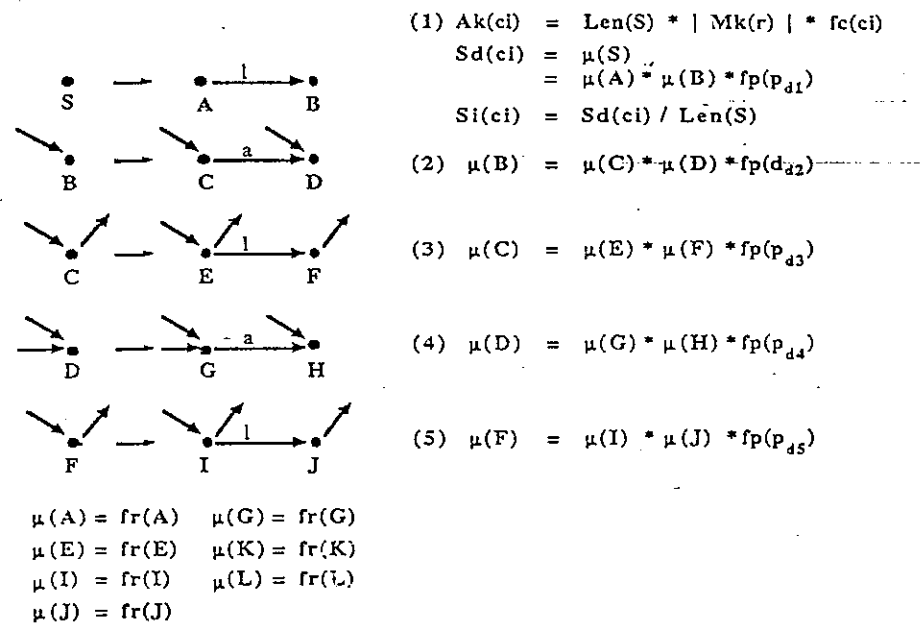


Fig. 5.1 The attributed context-free web grammar that generate the character 腦 with the evaluation parameters as its attributes.

## VI. AUTOMATIC EVALUATOR

A RBCI method evaluator is a machine that accepts the Chinese character patterns as its input and produces the evaluation result as its output. The evaluation result is obtain by performing the evaluatin procedures on the user-defind method automatically. The evaluator consists of two parts : a transformer and a simulator. A transformer is a preprocessor which transforms the dot-matrix patterns into the decomosition graphs of Chinese characters in the user-defined method. A transformer should extract radicals from dot-matrix patterns based on the predefined radical set. Also, it should constructs decomposition graphs from the recognized radicals [9,15,16,20] and their spatial relations based on the decomposition rules defined in a particular method

[1,8,20]. We are interested in the simulator here. According to the discussion in the paper, an attributed context-free web grammar can be derived by parsing the decomposition graph of a Chinese character and the evaluation result of a RBCI method can be obtained by evaluating the attributes of the grammar. We use a simulator as a machine for the evaluation of a RBCI method. A simulator is a push down automata [5] which accepts decomposition graphs generated by a transformer as its input and obtains the evaluation result as its output. Thus a simulator RBCIS can be defined as a six tuple

$$\text{RBCIS} = ( Q, S, \Delta, \delta, q_0, F )$$

where:

1.  $Q$  is a finite set of states.
2.  $S = G(V, E)$  is a finite set of decomposition graphs, the node set  $V$  is a finite set of code symbols and the edge set  $E$  is a set of edges with a dash-line arrow representing the part-of predicate and the solid-line arrow representing the spatial relations between code symbols.
3.  $\Delta$  is a finite set of output values.
4.  $\delta$  is a mapping from  $Q \times (S \cup \{\lambda\})$  to a finite subset of  $Q \times \Delta^*$ , and  $\lambda$  denotes a null element.
5.  $q_0 \in Q$  is the initial state.
6.  $F \subseteq Q$  is the set of final states.

It is worthwhile to note that the values of  $A_k(M)$ ,  $S_d(M)$  and  $S_i(M)$  are updated at each intermediate state. In other words, to update the value of  $A_k(M)$  at the state  $q_x$ , the values  $x-1$  and the sum of  $A_k(c_i)$  from  $q_1$  to  $q_{x-1}$  were accumulated at the state  $q_{x-1}$ .

## VII. CONCLUSION

This paper present a theoretical basis for the evaluation of radical-based Chinses input methods. An automatic evaluator of RBCI methods may consist of two parts : a transformer and a simulator. The transformer transforms the dot-matrix Chinese characte patterns into decomposition graphs of Chinese characters in the user defined method. The simulator accepts decomposition graphs as its input and produces the evaluation result as its output. A derivation diagram can be obtained from a decomposition graph. A parser converts such derivation diagrams into a set of web productions of a context-free web grammar which are derived from two types of subgraphs of the derivation diagram : the skeleton and the section. By the application of the semantic rules of each production in the grammar, the encoding result of a RBCI method can be obtained and the evaluation parameters can be computed.

#### REFERENCES

- [1] C. F. Herot, "Spatial Management of Data," *ACM Trans. on Database Systems*, Vol. 5, No. 4, pp. 493-514, 1980.
- [2] C. K. Chen and R. W. Gong, "Evaluation of Chinese Input Methods," *Computer Processing of Chinese & Oriental Languages*, No. 4, pp.236-247, 1984.
- [3] D. E. Knuth, "Semantics of Context-Free Languages," *Mathematical Systems Theory*, Vol. 2, No. 2, Springer-Verlag Work Inc., 1968.
- [4] H. J. Zimmermann, *Fuzzy Set Theory and its Applications*, Kluwer-Nijhoff Publishing.
- [5] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- [6] J. L. Pfaltz and A. Rosenfeld, "Web grammars," Proc. First Int. Joint Conf. Artif. Intell., Washington, D.C., pp.609-619, May 1969.
- [7] J. M. Brayer and K. S. Fu, "Some multidimensional grammar inference methods," *Pattern Recognition and Artificial Intelligence*, (C. C. Chen, ed.) Academic Press, New York, 1976.
- [8] J. P. Hwang, "Extracting Spatial Relations from Images Represented in Quadree," M. S. Thesis, Graduate Institute of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., June 1988.
- [9] K.S.Fu, *Syntactic pattern recognition and applications*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
- [10] K. S. Fu and T. L. Booth, "Grammatical Inference: Introduction and Survey ," *IEEE*

- Trans. on System, Man and Cybernetics*, Vol. SMC-5, No. 1, pp. 95-111, 1975.
- [11] L.A.Zadeh, "Fuzzy sets," *Inf. and Control*, pp.338, 1965.
- [12] L. R. Hu, Y. W. Chang and K. T. Haung, "The Three Corner Coding Method," *Proceedings of 1979 International Conference of Chinese Language Computer Society*, pp.45-61, 1979.
- [13] M. W. Du, C. C. Hsieh, S. H. Cji and S. C. Chu, "Character Composition and Methods to Represent Radicals," *Proceedings of 1st International Symposium on Computer and Chinese, I/O Systems*, pp63-78, 1973.
- [14] P. M. Lewis II, D. J. Rosenkrantz, and R. E. Stearns, *Compiler Design Theory*, Addison-Wesley, Reading, Mass., 1976.
- [15] P. N. Chen , Y. S. Chen and W. H. Hsu "Stoke Relation Coding - A new Approach to Recognition of Multi-Fonts Printed Chinese Characters," *Proceedings of National Computer Symposium*, R. O. C., pp.432-436, 1987.
- [16] P. P. Wang, "The Topological Analysis, Classification, and Encoding of Chinese Characters for Digital Computers Interfacing," *Proceedings of 1st International Symposium on Computers and Chinese I/O System*, pp.417-439, 1973.
- [17] Steven Pinter, *Visual Cognition*, The MIT Press, pp. 97-161,1984.
- [18] S. Y. Lo, "A Scientific Model for Comparing Various Methods of Inputting Chinese Characters into Computer," *Computer Processing of Chinese & Oriental Language*, Vol.2, No.1, pp.36-57, 1985.
- [19] T. G. Evans, "Grammatical Inference in Pattern Analysis," *Software Engineering* Vol. 2, J. T. Tou (ed), Academic Press, 1971.
- [20] "仿宋體、明體、黑體、楷書四種印刷字體M中文字形識別系統的設計與製作," *Proceedings of Workskop on Computer Vision, Graphics, and Image Processing* , R.O.C., 1987.
- [21] 倉頡第三代中文輸入法, 全華科技, June 1988.