

Research on Computer Language Behaviors from the Perspective of Logical Pragmatics

Zhangxing

Liaoning Police Academy of Basic Faculty, Dalian 116036 Liaoning
1721231788@qq.com

Abstract - To analyze the computer behavior by the point view of the pragmatic logic, the computer behavior refers to the computer operation of the program execution. Because the computer program is a collection of pragmatic statements, so the computer behavior is a typical pragmatic behavior.

Index Terms - pragmatic logic, computer

Logic plays an important role in computer, it promotes the development of computer science. Logic is researching on inference, and mathematics is closely related to calculation. Inference and calculation have a lot in common, calculation can be transferred into inference and inference can also be transferred into calculation under certain environment. For this reason, a large part of research results in logic can be applied to computer technology. Some researchers used to conclude the relationship between inference and calculation into the following formula: Calculation= Manipulated Inference. This forum shows the essence of relationship between them. In the process of general inference, the order of rules that being referred to is not mechanical. But if the order that used is a predetermined and mechanical one, the usage of inference rules is called calculation. Thus besides seeking calculation methods, studying on the accuracy and efficiency of calculation methods and designing machines for calculating implement, the computer researchers have to study on problems like programming and the syntactic and semantic features of program language.[1]

To do a research on computer behaviors from a pragmatic logic view, computer behavior refers to the computer actions that generate from the execution of programs. Computer program is a collection of pragmatic sentences, so computer behavior is a typical pragmatic behavior. It has the following features:

1. Pragmatic Force Determines the Patterns and Features of Computer Behaviors.

Pragmatic force is proved through special performative verbs----pragmatic behaviors. In computer language, pragmatic verbs are commands and functions. Different commands or functions lead to different computer running results. Thus pragmatic force determines computer behaviors. A standard result of a simple pragmatic behavior statement is F (P). F represents pragmatic force, while P represents contents of the proposition. In computer languages,

F is the force that a command or function occupies, P is the parameter or expression that the command or function refers to. It can be concluded from the logic features of commands and functions (pragmatic force) the patterns and features of computer behaviors.[2]

2. The Logic Features of Computer Behaviors can be Analyzed Through Universal Grammar.

Firstly, computer behaviors are implement results of computer language program. As mentioned above, programs are computer languages, the implement of programs are computer behaviors. Seeing from computer behaviors, programs are in certain order, so are computer behaviors, only if the last step is finished, it will come to the next one. From the perspective of language behaviors and pragmatic logic, every pragmatic behavior can proceed to the next step only after the implement of language effects. [3]The following is a part of a language program BASIC.

```
10CLS
20INPUT"please input an arbitrary number:"; X
30IFX> 0THEN
40PRINT"POSITIVE NUMBER"
50ELSE
60IFX< 0THENT
70PRINT"NEGATIVE NUMBER"
80ELSE
90PRINT"ZERO"
100END IF
110END IF
120END
```

This program judges the input number is a positive number, negative number or zero. The execution result is a series of computer behaviors. The implement result of statement10 is clearing screen; then goes to statement20, the computer waits for data input and store those data into variable X. Statement30 is a conditional statement, firstly, the computer has to do a judgment on the value of X>0, if the return value is true, it will come to statement40, and type the phrase "positive number" on the screen; on the contrary, statement40 and statement ELSE will be stepped over and turn to the following statements. It will go to the end of the program in this way. It can be concluded that the execution result of every correct

statement is a language effect, which is an act that can bring some certain effects. According to Searle, every language effect is a pragmatic behavior. It can be seen that computer behavior is a typical pragmatic behavior.[4]Second, the success condition of computer behavior is the success condition of pragmatic behavior. In the topic of computer language, we have confirmed the true and false of pragmatic behavior statements. The condition of a true statement is an indispensable semantic condition of a pragmatic logic. Semantic analysis can help get the idea whether one computer behavior can be successfully implemented; if possible, the execution result will also be known. The analysis on computer behavior is the analysis on those success conditions of corresponding statements, which can be achieved through the definition induction of statement structures.[5] Attention needs to be paid to the difference between this analysis and the former one: the analysis objects of computer are statements, and semantics is the “true” or “false” of a statement; the analysis object of computer behavior is pragmatic behavior, semantics is whether one behavior can be successfully or unsuccessfully carried out. It can be known from the assignment conditions of semantic models.

- (1) The sufficient and necessary condition of successfully implementing one simple behavior $F(P)$ is the content P of a relevant proposition is true under any relevant contexts; the sufficient and necessary condition of unsuccessfully implementing $F(P)$ is the content P of a corresponding proposition is false under at least one relevant context.
- (2) The sufficient and necessary condition of implementing a negative behavior $\neg A$ is behavior A can not be successfully implemented under any relevant contexts. The sufficient and necessary condition of unsuccessfully implementing $\neg A$ is behavior A can be implemented under at least one relevant context.
- (3) The sufficient and necessary condition of successfully implementing an implication behavior $A \rightarrow B$ is if behavior A is successfully implemented, then behavior B will be successfully implemented under any relevant contexts. The sufficient and necessary condition of unsuccessfully implementing an implication behavior $A \rightarrow B$ is behavior A is successfully implemented, while behavior B will not be successfully implemented under one relevant context.
- (4) The sufficient and necessary condition of successfully implementing a conjunction behavior $A \wedge B$ is both behavior A and behavior B are successfully implemented under its context. The sufficient and necessary condition of unsuccessfully implementing a conjunction behavior $A \wedge B$ is either behavior A or behavior B is unsuccessfully implemented under the context.
- (5) The sufficient and necessary condition of successfully implementing a disjunction behavior is either behavior A or behavior B is successfully implemented under its certain context. The sufficient and necessary condition of

unsuccessfully implementing a disjunction behavior $A \vee B$ is both behavior A and behavior B are unsuccessfully implemented under a certain context.

Computer behavior is the process and result of a computer program. The true and false of a statement will change the direction of a computer program, which is changing computer behaviors and then have an effect on the final result of a computer program. For this reason, in the process of computer programming, one should first know which computer language is being used. In this context, it is necessary to do a semantic analysis on statements and computer program to figure out whether computer behaviors are successful or not and find out the method of successful implement.

The following example is intending to figure out how the semantic analysis is able to predict computer behaviors by analyzing several computer language programs. This program is a part of a BASIC language program, which is a common conditional statement of BASIC.

```
CLS
INPUT"PLEASE INPUT TWO NUMBERS:", X,Y
IF X< Y THEN
Min=X
Max=Y
ELSE
Max=X
Min=Y
END IF
PRINT"THE MAXIMUM NUMBER IS"; Max
PRINT"THE MINIMUM NUMBER IS"; Min
END
```

This program is used to judge the magnitude of the two input numbers. In this program, what affects computer calculating results is the true or false of the conditional statement $X < Y$. If the conditional statement is true, those statements between THEN and ELSE will be implemented. The computer will store the value of X into variable Min and store the value of Y into variable Max. Then statements between ELSE and END IF will be stepped over. The value of X will be stored into variable Max and the value of Y will be stored into variable Min. then the statements after PRINT will be implemented, which are print the maximum number X and print the minimum number Y . these two models of operation lead to totally different results. Thus in this program, what affects computer behaving model and result is the true or false of the conditional statement $X < Y$.

According to the above analysis, it can be concluded that:

First, computer language is a kind of pragmatic language, which is a language of “performing things”

Second, computer behavior is a kind of pragmatic behavior, which is caused by language and will bring some certain effects. It is the result of language program operation.

Third, computer behavior is determined by computer language, different languages, and different contexts will result in different behaving effects.

Fourth, the pattern and effect of computer behavior can be known beforehand through analysis of corresponding language program and be controlled through language program management.

We can also analyze the application of pragmatic logic in computer behaviors from other respects, such as algorithm, functions of operating system, software engineering method, numerical calculation oriented application, measurement and control oriented application, computer assistant designing oriented application, management oriented application, etc.

Reference

- [1] Shun-Ji Huang. Science and Technology Revolution Influence Theory. Beijing: Renmin University of China Press.
- [2] Austin J Friet.(American)Debate and Argument
- [3] Shirong Zhang. The logic of Scientific Development. Beijing: People's Publishing House, 2008.
- [4] Jialong Zhang. History of mathematics logic development. Beijing: Social Sciences Academic Press, 2003.
- [5] Shiyeshoulang (Japanese). Fussy Engineering—Thoughts of New Century. Shenyang: Liaoning University Press, 2001.