

Research on the Relationship between Pragmatic Logic and Computer Language Behavior

Xing Zhang^{1, a}

¹ Liaoning Police Academy of Basic Faculty, No.260, Yingping Road, Ganjingzi District, Dalian, Liaoning, China, 116036

^aemail,

Keywords: Pragmatic Logic, Computer Language Behavior

Abstract. Pragmatic Logic is logic analysis tool and logic analysis system of Speech Act Theory. The research on pragmatic logic is focusing on logic features of statements and inference relations that consists of a group of special performative verbs—pragmatic performative verbs. This definition includes three aspects: 1. language form. In natural language, there is a group of verbs that have some certain forces. Such as, judge, require, promise, congratulate and declare, etc. are having separately the pragmatic force of judging, ordering, promising, expressing and declaring. The purpose of uttering sentences with those words in is to do some certain things. For example, the sentence “I promise that this job can be done on time.” is making a promise. It is called pragmatic force, words that occupying pragmatic force are called pragmatic performative verbs, and sentences that made up of these verbs are called pragmatic performative sentences. Pragmatic logic is researching on the logic features of pragmatic acts. 2. Inference relations. Pragmatic logic is based on the above language materials and researching on logic features of all sorts of complex pragmatic behavior statements and the inference relations between them. For example, “If he promises to finish the job on time, then I promise to finish the job on time” is an inclusion statement that composed of two pragmatic behavior statements, they are “He promise to finish the job on time.” and “I promise to finish the job on time.” Its structure is $F(P) \rightarrow F(Q)$. We have to confirm a few starting points that can be used for inference and establish some inference rules in those generally effective statements, which will contribute to the further inference of some other generally effective sentences. 3. System features. We also have to do a research on the meaning and explanation of system formulas and on features of this inference system, such as reliability, consistency and completeness. Reliability is that all the provable system formulas possess a certain property; completeness is that all the formulas that possess a certain property are provable. Reliability, consistency and completeness associate the grammar concept “provable” with the semantic concept “possessing a certain property” and establish the relationship between grammar and semanteme. For this reason, reliability, consistency and completeness are called the key features of a system. Reliability theorem, consistency theorem and completeness theorem are called the metatheorems of a system. The metatheorem that reflects the features of pragmatic logic system contributes to mark the difference between itself and metatheorems of other systems, so it becomes the main researching content of pragmatic logic.

Introduction

Pragmatic logic is widely used. Austin and Searle used to analyze human speech acts with speech act theory and illocutionary logic, they drew many meaningful conclusions. Stegmuller comments that “in the past 2500 years, for all those who were studying language in any way, this is a folly thing for them, for they haven’t come up with a concept that whose essence can be described with a brief sentence long before J.L.Austin, which is ‘we could use language expression to perform all kinds of acts.’ [1]

One of the main application fields of illocutionary logic is computer language analysis. Computer language is a typical illocutionary act language, which is the language of “performing things”. Computer program is an ordered set that composed of many pragmatic

statements. From the perspective of pragmatic logic, computer language shows the most distinct and typical feature of pragmatic logic.

Pragmatic Force Is A Part Of Statement Meaning

The fundamental meaning of computer language is that it produces a pragmatic act that has a special pragmatic force through statements in a specific speech context (source language or target language). Every statements of computer language is used for “performing things”, the statements that not “performing things” are called “remark statements”, they are put in brackets and will not be performed. Computer Language is used for “performing things”, not for “describing things”, which is more obvious than daily language.

Pragmatic force is a part of statement meaning. This sentence includes two meanings. The first one is computer language is related to context. For example, the statement “DIR” is used to list all the documents under the current path of DOS system and is used to list only database files (.DBF) in FoxBASE/FoxPro system. For another example, statement “USE ABC” is used to listing all the database documents named ABC.DBF in FoxBASE/FoxPro system, but the executive results might be different. It might open different database files under different paths (if the files exist under the specific path), it might not be able to open the files (if the files exist under the specific path) and return an error message “File does not exist”. It has to be pointed out that in the man-machine interaction, “Speaker” under context factors refers to the user, and “listener” refers to the computer. As mentioned above, language context, path, user (speaker), computer (listener) are computer language factors.

So, computer language is related to context. The second implication is that the pragmatic force of computer language is realized through specific pragmatic verbs—commands and functions. Pragmatic verbs in computer verbs define the pragmatic force of specific statements. As Searle said, “the successful implement of that kind of acts will necessary achieve that purpose, and that kind of purpose need to be realized through that kind of acts.” [2]For example, in the language BASIC, the following statements: 10LETA=1000 20 PRINT A, they are possessing different language force because they have different pragmatic verbs—commands.statement10 is an assignment statement, it assigns a number (1000 in this statement) to a variable (A in this statement); statement20 is a print statement, it prints a specific number out (the value of A in this statement). In language BASIC, functions can not be used directly as pragmatic verbs, they can only form expressions and make up statements together with commands. For example, the following statements, 30PRINT LEN (“ABCDEFG”) 40 PRINT ASC (“ABCDEFG”) are both print statements. Due to the different functions that used in the expressions, their printing results are different. The printing result of statement30 is 6 (the length of character string “ABCDEF”), the result of statement40 is 65(the ASCII code of A, the first character of string “ABCDEFG”). From the perspective of pragmatic logic, these two statements include the same pragmatic verbs, which are the same commands PRINT. So they have the same pragmatic force, which is to print a certain value (the value of the expression here). But the different proposition content lead to different results of pragmatic acts. It can be concluded that the commands and functions are typical pragmatic act verbs of computer language. In computer language, the pragmatic forces of statements are realized through these pragmatic verbs. Without these pragmatic verbs, any statements of computer language won’t make sense.

Pragmatic Logic Features Of Computer Language Can Be Analyzed With Universal Grammar

Searle thinks a sufficient pragmatic logic is indispensable for a sufficient universal grammar. Pragmatic force and proposition are two parts of basic statements meaning. The ideal language of universal grammar should include logic constants and operators that can produce and express all the possible pragmatic forces of the statement. Any statements of any natural language

should be able to be converted into ideal language statements of universal grammar, and these statements should reflect the pragmatic potentials of natural language.

The following passage is going to analyze pragmatic logic features of computer language, including grammar analysis and semantic analysis. The analysis will be done from aspects of the basic statements of computer language: complex statements, successful conditions of statements, programs, function call, procedure call, and context.

First, basic statements of computer language are typical pragmatic act statements. A pragmatic act consists of two parts that are pragmatic force and proposition. The structure is F (P).

Attention, formula1 has two uses. It represents a pragmatic act and the statement structure of the pragmatic act. For the first usage, we called F (P) a pragmatic act, F represents the pragmatic force of the statement, and P represents the proposition content of a statement. For the second usage, we call F (P) a pragmatic act statement, here F represents pragmatic force verb of a statement, and P represents the proposition form of a statement.

In formula1, F represents pragmatic force verb of a statement, and P represents the proposition form of a statement. In computer language, pragmatic force verbs are commands and functions of this computer language, proposition patterns are proposition expressions of this computer language. Therefore, we can analyze the basic statement structure of computer language by using F (P). Commands and functions of computer language are the pragmatic force verbs of basic statements in this certain computer language, F can be used to represent this part; expressions that correspond with certain rules are proposition patterns of basic statements, P is used to represent this part. It can be seen from the above analysis that the basic statement structure of computer language is the statement structure of a typical pragmatic act. Furthermore, this kind of analysis has deepened the understanding of speech act theory and illocutionary act.

Second, complex statements are the complex structures of basic statements. In computer language, complex statements are composed of basic statements that are being applied with specific commands and functions. For example, in language BASIC, one conditional statement can be realized through using the following structure IF...ELSE...THEN:

```
10INPUT"Please input two integers:",x,y
20Ifx < Ythen
30min=x
40ELSE
50min=y
60END IF
70PRINT"the minimum value is:",min
80 END
```

In C language, similar results need to be realized with the following structure:

```
Viod main() {
  Int x,y;
  Print ("Please input two integers :");
  If<y min=x;
  Else min=y;
  Print ("The minimum value is:%d\n".min);
}
```

In the above example, complex statement of structure IF...ELSE...THEN includes several basic statements, these statements need to be composed by using commanding verbs like IF, ELSE, THEN, and sometimes need to use conjunctions like NOT, AND, OR. Some similar complex structures also include selective structure SELECT CASE...END SELECT and loop structure WHILE...WEND, etc. Generally speaking, in computer language, statements collection that composed of any two or more basic statement structures and have an independent meaning all can be called complex statements. Sequential structure, selective

structure, loop structure are three basic complex statements of computer language. In addition, in computer language, function call and procedure call are often used to realize the functions of complex statements. Therefore, complex statement of computer language is a pragmatic logic concept that is meaningful, powerful and able to realize different kinds of objectives.

Third, the condition of a true statement is the corresponding semantic condition of pragmatic logic. We can use semantic model to precisely define the truth value, satisfiability, effectiveness, universal validity of a statement. According to semantic model, the conditions of a true statement can be defined by concluding statement structures in computer language.

In pragmatic logic, if the statement is true, it is able to implement the corresponding acts under the certain context; if the statement is false, it is able to implement the corresponding acts under this certain context. This kind of semantic analysis is of great importance in computer and behavior analysis.

Fourth, computer program is an ordered set of statements. A computer program is composed of some statements that made up of specific language and it has a certain order. From the perspective of pragmatic logic, computer programs are ordered sets of “performing” statements.

Fifth, function call and procedure call are vital methods of program structuring. It is convenient to realize context conversions without destroying the structure of a program by using function call and procedure call between different languages. Therefore, function call and procedure call are important methods that are commonly used in program structuring.

Sixth, computer language environment is combination of all the above essential factors. Computer language environment is not merely referring to one specific language factor; it usually refers to a combination of many language factors.

Seventh, computer linguistic competence is a combination of the pragmatic force of a language statement. Every statement has its defined pragmatic force, and computer linguistic competence is a combination of specific pragmatic effects of different statements sets under specific context in a specific language.

The above includes analysis (syntactic analysis) and semantic analysis. Statement analysis is the basis of semantic analysis. Only statements with correct grammar can be applied with semantic analysis. Manipulative approach is usually used in grammar analysis, it is intending to define whether a statement has a certain property or not in limit procedures. Such as, the grammar is right or not. Manipulative approach is not used in semantic analysis, which means it is not able to define whether a statement has a certain property or not in limit procedures. It is possible for a computer language compiler to detect grammar mistakes, but not possible to detect semantic mistakes.

Reference

- [1] W. Stegmuller, (German) Contemporary Prevailing Philosophy, Page 66, Beijing, the Commercial Press, 2000.
- [2] Shushan Cai, Speech Act and Pragmatic Logic, Beijing, China Social Sciences Publishing House, 1998.
- [3] Yutian Wang, Bingrong Wu. Inductive Logic and Artificial Intelligence. Shanghai, China Textile University Publishing House, 2009.
- [4] Minghua Ma, Theory of Computability, Beijing, Tsinghua University, 2004.
- [5] Jialong Zhang. The Development Mathematical Logic, Beijing, Social Sciences Academic Press, 2000.