

Semantics in Philosophy and Computer Science

Enrico Leonhardt
Computer Science Dept.
Technical University
Dresden, Germany

Enrico.Leonhardt@inf.tu-dresden.de

Dipl. Ing. Simone Röttger^{*}
Computer Science Dept.
Technical University
Dresden, Germany

Simone.Röttger@inf.tu-dresden.de

ABSTRACT

The term ‘formal semantics’ is quite important in computer science since it is used to define programming languages. The techniques behind this term are very powerful but hard to understand. To get a better understanding of the whole area it might be beneficial to have a wider impression of semantics in general. This paper tries to present why and how semantics is used in philosophy and what the correlations are to semantics in computer science.

Keywords

Semantics, Philosophy, Computer Science

1. INTRODUCTION

Semantics (Greek *semanticos*, giving sings, significant), in general, refers to the aspects of meaning that are expressed in either a natural or an artificial language.

Nowadays, semantics are important in many fields such as linguistics, philosophy, psychology, information theory, logic, and computer science. It is used in different ways whereby in linguistics and philosophy, the most established lines of meaning investigations are published. Many of them discuss the meaning of complex terms that are deviated from simple terms in consideration of syntax, and try to answer the question whether a phrase is true or not, which is known as semantic theory of truth.

To investigate a natural language, philosophy abstracts the content from natural phrases to build a formal language, and uses logical concepts.

The remainder of the paper is organized as follows. In Section 2, the problem of truth in philosophy is presented with some methods of resolutionis. Followed by a short overview of the different areas of semiotics in Section 3. Section 4 talks

^{*}Supervisor

about formal semantics in Computer Science and finally Section 5 gives the conclusion.

2. THEORY OF TRUTH

One goal in philosophy is to find a formal definition for the ‘true’ predicate according to the term of truth. By doing this for linguistic entities the difference between “sentence” and “statement” must be clarified as Stegmüller pointed out [2]. Sentences can contain so-called indicators such as ‘I’, ‘you’, ‘now’ etc. that create different statements with different meanings depending on speakers and situations. So, it is not possible to decide whether the sentence

“He is not here, today.”

is true or false.

In order to give a consistent definition of “truth” it is necessary to formalize languages or use formal languages without such indicators.

2.1 TARSKI Scheme

Only the theoretical meaning of the ‘true’ predicate is of importance here. As the logician and philosopher Alfred Tarski [3] pointed out, this is comparable with the intuitive association of ‘truth’ of a statement, which argues that something is so and so, and at the end it is really so and so (in the real world). Although this might be satisfying in terms of simpleness, it is a kind of fungous definition and not really clear and correct. In order to achieve a better ‘correctness’ Tarski came up with a scheme that is:

“X is true if and only if p.”

Where p can be replaced by any statement, and X refers to the name of it. One example where this scheme can be used is an intuitive verbalization such as:

“The statement ‘the sun is shining’ is true
if and only if the sun shines.”

Here, the use of the Tarski scheme, where p is ‘the sun is shining’ inclusive the quotation marks and X is some identifier for p, gives a partial definition of the ‘true’ predicate because the theoretical meaning of true is defined for this particular statement.

However, a partial definition is not a definition of the ‘true’ predicate within a colloquial language (natural language), which is in demand and has the requirements:

- adequate in respect of content
- formal correct

Adequate in respect of content means that “every statement” such as the example above with a ‘true’ predicate in it used with the Tarski scheme is logical determinable. Even though this is possible for some examples it does not work for “every statement” because of paradox, or antinomies respectively that is presented in the next section.

2.2 Antinomies of Truth

A paradox or antinomy is something where a conflict is generated in spite of faultless using logical and mathematical deductive methods of reasoning.

Mostly the terms paradox and antinomy are set equally. However, there is a difference between them, which should be clarified. An antinomy for instance is a logical paradox whereas there also paradox definitions and paradox act commandments exist.

2.2.1 Paradox definition

One example for a paradox definition is:

“ A suicide murderer kills all
that do not kill themselves.”

Since the question “Does he kills himself or not?” can be answered in both ways, there is a conflict. However, this situation only results from the assumption that such a person really exists. On the other hand, the inference says that such a person can not exists because the question above would generate a conflict.

For this reason, it is not an antinomy since there is no proof that such a situation exists. Respectively there is no logical problem.

2.2.2 Paradox act commandment

It is also possible to use the term paradox for act commandments that never ends. An example for such an endless act commandment is

“‘Give someone the commandment
‘follow the instructions on a sheet of paper’
where and on both sites is written
‘please turn around’.”

Even though this really exists, it does not generate a logical problem as well.

2.2.3 Logical paradox - Antinomy

In contrast to paradox definition, a logical paradox or antinomy arises once there is a proof for the statement S , whereas S contains two parts. The part of a statement that claims the opposite of another part S_i is called the negation $\neg S_i$. So, a proof for S exists if there are proofs for both sites because S is the ‘and’- catenation of its parts ($S_i.\neg S_i$).

From section 2.1 it is clear that a formal definition of the ‘true’ predicate must satisfy the requirement ‘adequate in respect of content’. This fails once a partial definition with the Tarski scheme is not logical determinable.

Now, if the ‘true’ predicate is used in such a statement antinomies can be created. One “popular” example is

“All Greeks are liars, said a Greek.”

Another more accurate version is from Lukasiewicz:

“The statement on page 8 is not true.”

As Stegmüller pointed out on page 26 the use of the Tarski scheme and the faultless use of logical deductive methods of reasoning ends in a conflict for these examples.

Therefore, it is not possible to give a formal definition of the ‘true’ predicate in this way, which is also proven by some other different antinomies. However, all of them have two conditions in common as Tarski pointed out:

- the languages that are used to construct antinomies contain ‘true’ predicates
- the validity of logical basic laws

To find a solution for the problem of antinomies one of these conditions must be eliminated. Since it is not possible to eliminate or give up the validity of logical basic laws, the elimination of the first condition is inescapable.

2.3 Division of Object and Meta language

For this reason, Tarski divided the natural language in two languages. The first one can be used to describe anything in the objective world. This is called the object language. It does not contain any ‘true’ predicates and cannot say anything about other statements. An example is:

“The table is white.”

The second one can be used to say everything. This is called Meta language and contains ‘true’, ‘false’ predicates etc. that might be used in order to say something about other statements (that could also formulated in object language).

“The previous example is not true.”

In addition, a statement in object language is called statement of order one whereas a statement in Meta language is at least of order two. If a statement refers to another, which is already of a higher order the actual statement is one order above (table 1).

Table 1: Order hierarchy of statements

Statement	Order
“The sun is shining today.”	one
“The statement above is true.”	two
“The second statement here is true.”	three

Through this division of the natural language, the construction of antinomies is not possible anymore as Stegmüller presented on page 40 for the antinomy of a statement such as

“The statement on page 8 is not true.”

By using object and Meta language, the order must be included in the statement:

“The statement of order one on page 8 is not true.”

Assuming there is a page 8 and only this statement is written there an empirical verification would show that there is no sentence of order one on page 8. The decision whether that statement is true or not depends now on the method of analyzes.

Since the method by B Russel is well accepted, the statement is false by using it because it contains the partial statement:

“ There is a statement of order one on page 8 that is false.”

In this way, it is possible to eliminate antinomies. However, this is not a proof that a definition of the ‘true’ predicate is working in that way. There are still sentences possible that are not determinable such as:

“The color is too late.”

In order to investigate a language and solve the truth problem in philosophy it is necessary to formalize languages and prevent those situations. This is done by the following three different techniques that are part wise mutually, also called “semiotics”:

3. SEMIOTICS

Semiotics, in general, is the study of signs and symbols. It can be used for every scientific investigation of language systems whereas there are two different approaches for different fields. Semiotics can be empirical or “pure”. The empirical semiotics is used in order to investigate historic traditional language systems. On the other hand, the “pure” semiotics helps to create new artificial language systems and investigate them as well. In both areas semiotics have three branches, namely (1) syntax, (2) semantics, and (3) pragmatics.

3.1 Empirical semiotics

Empirical semiotics is used in Linguistics and Philosophy.

3.1.1 Syntax

Syntax concentrates only on the formal structure of a statement. It is the study of rules, or “pattern relations”. For instance the sentence

“The color is late.”

is a correct English sentence construction in terms of “pattern relations” (subject + verb + adjective).

To find general laws that govern the syntax of all natural languages, modern research attempts to systematize a descriptive grammar.

3.1.2 Semantics

Semantics is the study of aspects of meaning. It analyzes the meaning of a statement only by its colloquialism and its content whereas two different sorts of meaning a significant expression may have:

- the relation that a sign has to other signs (sense)
- the relation that a sign has to objects and objective situations, actual or possible (reference)

For the example “The color is late.” the sense considers the relation between the subject “color” and adjective “late”. Does it make any sense? The reference investigates the meaning that this statement has in the objective world.

In addition, there are different syntactic levels of semantics:

- the meaning of each individual word is analyzed by lexical semantics
- relationships between objects within a sentences is referred by structural semantics
- combination of sentences as real or hypothetical facts
- texts of different persons that interacts somehow (discussion, dialog.)

The connection between these levels is realized by the Frege principle, which says that the meaning of a complex sign is a function of meanings of their sub meanings.

MEANING(the color is late) = f(MEANING (the),
MEANING (color), MEANING (is), MEANING (late))

3.1.3 Pragmatics

Pragmatics is the most extensive technique. It considers all factors of the environment such as (1) the speaker, (2) the colloquialism (statement structure), and (3) the content that is focused by the speaker.

3.2 Pure semiotics

To define new artificial language systems or formal languages in logic, mathematics, information theory and computer science “pure” semiotics also called formal semiotics is used.

3.2.1 Syntax

Syntax defines the formal grammar, or simple grammar. It provides sets of rules for how strings in a language can be generated, and rules for how a string can be analyzed to determine whether it is a member of the language or not.

3.2.2 Semantics

Semantics defines a mathematical model, which describes the possible computations of a formal language especially programming languages in computer science. The different approaches of semantics are called:

- Denotational semantics
- Operational semantics
- Axiomatic semantics

Whereas in logic other modern approaches of semantics are important e.g.:

- Model-theoretic semantics
- Proof-theoretic semantics
- Truth-value semantics
- Game-theoretical semantics
- Probabilistic semantics

3.2.3 Pragmatics

As pragmatics of empirical semiotics, pragmatics for formal languages considers the environment. Such environments are e.g. different compilers, operating systems or machines.

4. FORMAL SEMANTICS

In computer science 'pure' semiotics are used to define artificial languages respectively programming languages.

The syntax defines formal grammars that are often context-free, to describe the set of reserved words and possible token-combinations of a programming language.

Where semantics defines a mathematical model of computation by the following techniques:

- **Denotational semantics** is used to translate each phrase in the language into a denotation, i.e. a phrase in some other language. Denotational semantics loosely corresponds to compilation, although the "target language" is usually a mathematical formalism rather than another computer language. For example, denotational semantics of functional languages often translates the language into domain theory;
- **Operational semantics** is used to describe the execution of the language (rather than by translation). Operational semantics loosely corresponds to interpretation, although again the "implementation language" of the interpreter is generally a mathematical formalism. Operational semantics may define an abstract machine (such as the SECD machine), and give meaning to phrases by describing the transitions they induce on states of the machine. Alternatively, as with the pure lambda calculus, operational semantics can be defined via syntactic transformations on phrases of the language itself;

- **Axiomatic semantics** is used one gives meaning to phrases by describing the logical axioms that apply to them. Axiomatic semantics makes no distinction between a phrase's meaning and the logical formulas that describe it; its meaning is exactly what can be proven about it in some logic. The canonical example of axiomatic semantics is Hoare logic.

The distinctions between the three broad classes of approaches can sometimes be blurry, but all known approaches to formal semantics use the techniques above, or some combination thereof.

However, it would be wrong to view at these styles separately. In fact, all of them are highly dependant on each other as Winskel pointed out [4]. For example, showing that the proof rules of an axiomatic semantics are correct relies on an underlying denotational or operational semantics.

4.1 Dynamic and Static Semantics

Apart from the choice between denotational, operational, or axiomatic approaches, there are two more formal semantics introduced by Consel and Danvy [1]:

- **Static semantics** considers all properties that do not change during the execution.
- **Dynamic semantics** considers all properties that might change during the execution.

5. CONCLUSION

In this paper a rough overview is given about how important semantics are in philosophy and how this is related to computer science.

6. REFERENCES

- [1] C. Consel and O. Danvy. Static and dynamic semantics processing. In *POPL '91: Proceedings of the 18th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 14–24, New York, NY, USA, 1991. ACM Press.
- [2] W. Stegmüller. *Das Wahrheitsproblem und die Idee der Semantik*. Springer Verlag, Vienna, 1957.
- [3] A. Tarski. *THE SEMANTIC CONCEPTION OF TRUTH AND THE FOUNDATIONS OF SEMANTICS*. Philosophy and Phenomenological Research 4, University of California, Berkeley, 1944.
- [4] G. Winskel. *The Formal Semantics of Programming Languages*. Foundation of Computer Series, 1994.