An Algorithmic Perspective on Denotation

Mihail Radu Solcan

"The present King of France is bald."
(Bertrand Russell)

Mihail Radu Solcan (1953–2013) Professor at the Faculty of Philosophy, Bucharest University. Author, among others, of the books **Freedom, Minds and Institutions** (2003) and **Eseul filosofic** (The philosophical essay) (2004).

1. The Computer As a Platonic Cave

Computers draw mixed reactions from philosophers; in some perspectives, computers are a poor source of inspiration for philosophy, while in others there are reasons for far more optimistic views.

The computing machinery, in the optimistic position, can be used as a tool for building philosophical models and making "thought" experiments. Models, such as the Platonic allegory of the cave, were built "by hand," but now they can be produced on a very special machine, the "philosophical computer."[1]

A philosophical model, such as Plato's Cave, has also inspired research in artificial intelligence. For example, one such program creates artificial windows (in windowless rooms).[2]

The above examples illustrate the uses of the computers *by* philosophers and computer scientists *for* modeling and image streams. Here, we stress the

A first draft of this paper is available on the web page of the author, at http://filosofie.unibuc.ro/~solcan/shelf/dap/dap.html.

idea of the computer *as* a model, namely as a new version of the Platonic cave. Let us have a look at Plato's text:

> *Imagine people living in a cavernous cell down under the ground; at the far end of the cave, a long way off there is an entrance open to the outside world. They've been there since childhood, with their legs and necks tied up in a way which keeps them in one place and allows them to look only straight ahead, but not to turn their heads. There is firelight burning a long way up the cave behind them, and up the slope between the fire and the prisoners . . . a low wall has been built. . . . Imagine also that there are people on the other side of this wall carrying all sorts of artefacts. . . . do you think they'd see anything . . . except the shadows cast by the fire on the cave wall directly opposite to them?*
>
> Plato, *Republic* 514a–515a[3]

There is a cave effect in the computer too; the images on the display are like the faint images of the real things in Plato's cave. Like the prisoners in the cave, computer users see the effects of the ideas, not the ideas themselves. The users send inputs from peripheral devices and then get an output on other peripheral devices. They have their backs turned to the ideas behind the computer programs.

Now, what are the ideas behind the data flow in the computer? According to Steven S. Skiena, "an algorithm is a procedure to accomplish a specific task. It is the idea behind any computer program."[4] The cave effect in the computer means that ideas are not manipulated as ideas by the device, but as effigies of ideas.

Embodied in a program, the algorithm is usually defined in terms of instructions. An algorithm is made up of clear and unambiguous instructions for transforming an input into an output.[5] In what follows we focus, however, upon the rules behind human actions, rather than on the movements of some mechanical device.

The theory of algorithms is a newcomer. It started as a by-product of the investigations in the 1930s of the decision procedures in logical and mathematical systems. A lot of interesting results have been achieved in the pure theory of algorithms.

In more practical terms, the design of an algorithm starts with an idea formulated in natural language. The algorithm is then translated into a more precise language and it is subjected to a process of stepwise clarification.

It is prudent to distinguish between the algorithm and the algorithmic process. Take, for example, three pairs of positive integers: (24, 12), (28, 21) and (30, 16). If we apply to them Euclid's algorithm we get three different *processes*.

In the study of an algorithm, there are two very significant aspects. First, we must show that the algorithmic process halts and returns a value. We have to prove that this result is correct.

On the other hand, any algorithm is time-consuming and it uses more or less of the resources of the computer. Thus it makes sense to have a look at the use of computing resources involved in an algorithm.

In the cave, there are actions: the people behind the low wall move all kinds of artefacts; the prisoners see the shadows of these artefacts. Likewise, the computing machinery involves actions. This is the most important aspect, because algorithms capture patterns of action and thus make possible their reproduction.

2. The Uses of Algorithms

The impact of the theory of algorithms is stronger in philosophical logic, but it is also felt in other areas of philosophy.

There is an indisputable impact of algorithm theory and computer programming upon philosophers interested in cognitive science. Daniel Dennett, for example, had a striking idea: he sees evolution as an algorithmic process.[6]

Jakub Szymanik contrasts traditional semantics with algorithmic semantics. The new semantics pays attention to the dynamics of language and knowledge. A restriction that might raise many questions is the assumption that in algorithmic semantics the universes are finite.[7]

There might be a lot of objections against a computational turn in philosophy; did the computational turn really bring something new and interesting? Isn't it a replay of a failed old approach involving the use of tests for the verification of meanings? In the following subsections, we briefly summarize two contributions connected with the use of algorithms.

2.1. Yannis Moschovakis on Algorithms and Meaning

In a series of papers, Yannis Moschovakis developed an original approach to the concept of algorithm. Moschovakis stresses the idea that algorithms are *semantic* objects. According to him this is quite obvious, but he wants to emphasize the contrast with such syntactic objects as the computer programs.[8]

Moschovakis has proposed a formal framework for understanding ideal, abstract algorithms.[9]

The basic idea in Moschovakis's approach is simple. Let us say, for example, that we examine a declarative sentence. This might be "Athens and Bucharest are in the same meantime zone." Then let us examine the algorithm for determining the truth of this sentence. The algorithm remains the same if I translate this sentence into Romanian and Professor Moschovakis translates it into Greek.

Anyone may use the same algorithm in order to determine the truth of the Romanian or the truth of the Greek sentence.

According to Yannis Moschovakis, an algorithm such as the one that we have mentioned above is Frege's *sense*. This is precisely the suggestion contained in the title of Moschovakis's paper (1994). Moschovakis argues that the algorithmic understanding of Frege's *sense* is presupposed by almost any interpretation of the work of Frege. Obviously, they do not explicitly use the concept of algorithm, but some idea like that of method or procedure for determining the denotation.[10]

2.2. Pavel Tichý and the Intensional Analysis

The work of Pavel Tichý corroborates and generalizes Moschovakis's observation concerning Frege's *sense*. It is possible to identify in logic and philosophy a clear tendency to connect intensions with algorithms. For Pavel Tichý, *intensions* are procedures. He distinguishes between the logical *type* of procedures and the type of the extensional entities.[11]

Tichý approaches directly the core of a famous method in analytical philosophy, the formalization of natural language, in order to get rid of the traps that it contains. He suggests that a very important step in the formalization process is the disentanglement of the references to extension from the references to intension.[12]

In Tichý's approach there is a strong emphasis on types. But does he really use the concept of algorithm? Of course, one cannot find the *word* algorithm in his 1971 paper, but this is not the point. He writes about the possibility of talking about intensions in terms of Turing machines. This would be different from Moschovakis's concept of algorithm, which is expressly *not* conceived in terms of computing machines. But the approach exploits the same idea of a connection between intensions and algorithms.

3. From the Ideal-Languages Analysis to Algorithmic Analysis

More than a hundred years ago, Bertrand Russell published a paper (Russell 1905) in which he explained his version of the logical analysis of the natural language. The basic idea was that constructions in a natural language contain traps. We avoid such traps with the help of translations into a logically flawless language.

Russell's analysis clearly separates the surface and the deep levels of the sentences formulated in a natural language. We reach the deep levels with the help of logic. Perhaps the most important feature here is the exactness, the precision that is obtained with the help of the translations into ideal languages.

Let us now go back to the very idea of *analysis* as it is illustrated in the seminal 1905 paper by Bertrand Russell. Consider, for example, the following sentence:

The author of *Waverley* also wrote *Ivanhoe*.

One might be tempted to treat "the author of *Waverley*" as the subject of the above sentence. For Russell, this is a trap and one should avoid it.[13]

According to Russell, in the case of the sentences of a natural language, we have to go beyond their surface. "The author of *Waverley*" is dissolved by the logical analysis. First, Russell outlines a strict interpretation of "the."[14] When "the" is involved, a uniqueness condition must be satisfied. Thus, there is one and only one author of *Waverley*. The other condition that must be satisfied is an existential condition. Therefore, when one talks or writes about "the author of *Waverley*," we have to expand logically this expression into "there is one and only one person who wrote *Waverley.*"[15]

The most famous example offered by Russell is, of course:

The present King of France is bald.

Applying once again the analysis outlined above, we realize that the above sentence is expanded into "there is one and only one person who is now King of France and that person is bald." The first part of the conjunction is (in the year 2012) false, therefore the whole sentence is false.[16]

There is a huge literature on the topic of Russell's theory. It does not make sense to summarize it here.[17] As it is natural with a century-old theory, Russell's theory is dismissed by many authors today.[18] We may just mention that for P. F. Strawson "communication is much less a matter of explicit or disguised assertion than logicians used to suppose";[19] there is a variety of contexts in which we use the natural language and different presuppositions help us to make sense of expressions in these contexts.[20]

The algorithmic approach entails substantial modifications of the analysis of a natural language. But, as in the case of Russell's paper (1905) we focus our analysis upon denotations. We treat, as Russell did, the surface of the natural language as very rough and tricky, but, unlike Russell, we do not look for a translation into an ideal language. Linguistic expressions are construed here as structured sets of clues; the clues, among other things, permit us to find out denotations.

Now, a problem that is going to stir up trouble is the denotation of (declarative) sentences. Intuitively, it makes sense to think that the algorithm for finding out the denotation of the expression "the author of *Waverley*"[21] is going to lead us to a person (or some collective of authors), but which is the denotation in the case of a sentence? Is it really a truth-value?

Let's consider the example of the sentence "the shop around the corner sells books." The sentence denotes a situation. Talking about truth means that the algorithm reports that this is the situation.[22] If the sentence has a denotation, then the sentence is true. Otherwise, it is false; false means that it has no denotation.

Algorithms for finding out denotations are built on the basis of the clues contained in the linguistic expressions. Various natural languages offer rather different clues. The refinement of these algorithms leads to a clarification; it is the counterpart of the traditional logical analysis.

4. Algorithmic Analysis and Denotation

In contrast with the approach that establishes a correspondence between algorithms and meaning or sense, we do not treat algorithms as meanings. They are just ways of computing a denotation. Consider, for example, the following case: I number the shelves in my personal library; then I examine the expression "the author of the third book on the 22nd shelf." The denotation happens to be again the same as that of "Thucydides." But is this changing something that we might call *meaning*? This would be an endless process with no connection with a plausible idea of meaning. An algorithm is not a sense, it helps us to make sense of an expression.

Let us imagine that I am looking at a picture and I say that I am contemplating the picture of the author of the third book on the 22nd shelf in my library. This is a statement about my actions. It is not expanding any meaning, despite the fact that someone might use my words as *clues* for finding out that I am talking about a person whom she calls "Thucydides."

In Russell's classical analysis, as well as in the tradition that it inaugurated, the focus is on knowledge about the denotation. Russell's subtle stress is on descriptive knowledge.[23] What knowledge could exist about the present King of France? Obviously, no knowledge, except the knowledge of the fact that there is no individual who is now (legally) the King of France.

It is not difficult to identify the alternative to the focus on descriptive knowledge. We may focus upon the *procedures* we use for the discovery of the denotation.[24] This is what we propose here.

We shall illustrate the shift from the focus on descriptive knowledge to the focus on action with the help of an imaginary situation. Let us suppose that we are transported back in time, in Ancient Greece. We are still mainly animated by the passion for knowledge. We examine the following sentence:

The King of Sparta is lame.

First, the denoting phrase, as Russell would say, "the King of Sparta" is deficient. Existence, in this case, is not the problem. The problem is created by the fact that Spartans had two kings. The uniqueness condition is not satisfied.

If we think from the point of view of action, the phrase "the King of Sparta" leaves us in an unclear situation. Suppose that you have to give something to "the King of Sparta." Which one?

The mystery of the lame king is solved if we refer somehow to Agesilaus. Agesilaus was a Spartan King and he was lame. Xenophon wrote about him. There is, no doubt, knowledge involved in all this. But, if we put action in focus, we see linguistic constructions rather as sets of clues than elements of a system of truth-bearers. We use clues that can lead us to Agesilaus or the author who wrote about him. "Xenophon" and "the author who wrote about a lame King of Sparta" are good clues, for example, for a keeper of a collection of ancient books.

Before we go into the details of the algorithms involved, we should observe that we must distinguish between various actions in which we use sets of linguistic clues. For example, I do not live (now, at least) in France. I might go to France or use some other method to find out facts concerning present-day France. I might, on the other hand, listen to stories about a mythical France. With respect to the mythical France, I would *imagine* a lot of things, an action which is very different from that of searching and finding mere facts.

4.1. Looking for the Present King of France

Russell is like someone who has sent a search team in France and expects a report. Writing the report is like the postcondition of an algorithm. But, in order to be able to write the report, the team has to fulfill a precondition. The algorithm would be:[25]

Precondition: *The existence of a unique individual who is King of France.*

The body of the algorithm.

Postcondition: *A report about the King of France.*

The problem with this algorithm is that the precondition (the preliminary requirement) conflicts with the postcondition. It would be impossible to generate any genuine report if there is no King in France.[26] Let us modify the algorithm:

Precondition: *The existence of at least one individual in France.*

***While** you have not found a King of France or you have not examined all the individuals in France*

If** you have found a King of France **then

***For all** the other (yet unexamined) individuals in France*

Check if there is another king.

***End for*.**

If** there is no other king **then

Check if the King of France is bald.

Make a report about the situation.

Else

Report that France has two or more kings.

Report that the sentence "There is one and only one individual who is now King of France" does not denote anything.

End if.

Else

Report that the sentence "There is an individual who is now King of France" does not denote anything.

End if.

Examine the next individual in France.

End while.

Postcondition: *A report about the denotation of the sentence "the present King of France is bald."*

The above algorithm is obviously time-consuming. There are a lot of individuals in France. In the worst case, the search team has to examine the situation of each of all these individuals. The best case would be to find, after examining two individuals, that France has two kings (like ancient Sparta). This algorithm is quite awkward. Most of us would probably proceed like this:

Precondition: *The existence of a clear (written or unwritten) constitution of France.*

If** there is no king in France (according to the constitution) **then

Report that the sentence "There is an individual who is now King of France" has no denotation.

***Else if** there is a King of France*

Find the king.

If** the king is bald **then

Report that the king is bald.

Else

Report that the king is not bald.

End if.

End if.

Postcondition: *A report about the baldness of the King of France.*

4.2. Dreaming about a King of Present-Day France

Fiction seems to generate a lot of problems, if we are focused on knowledge. Suppose that we talk about an author who lives in present-day France and writes in Ancient Greek. There are persons in France who write in Ancient Greek, but the individual we imagine is an entirely fictitious person. The existence condition is not satisfied. Probably, there are at least some difficulties with the uniqueness condition too. Anything we say primarily[27] about this author is bound to be false. It would be however quite ridiculous, in a literary text, to use secondarily the denoting phrase and announce repeatedly that we are talking about a fictitious character.

Would things look differently in algorithmic perspective? Let us reexamine the algorithms discussed before. The precondition must be modified: we must allow some room for imaginary situations. The postcondition also has to undergo some changes: if we want a report, then we must admit reports about imaginary situations. The most spectacular changes, however, take place in the algorithm itself.

In an imaginary situation, it does not make sense to look for "the King of France." We add such an individual to the imaginary situation. We change the state of the world in order to make room for that individual.

Let us look to a possible algorithm:

Precondition: *The possibility to imagine that there is at least another person added to the set of persons living in France.*

Start *with the introduction of a new person.*
Imagine that this is a king of France.
Make sure that there is no other king of France.
If *you have imagined that the King of France exists* ***then***
Imagine that the King of France is bald.
Make a report about the situation of the imaginary King of France.
End if.

Postcondition: *A report about the imaginary situation of the King of France.*

It is not surprising that this algorithm is less time-consuming than an algorithm which would force us to examine the situation of all the individuals in France. It is similar to the algorithm in which we examine the constitutional situation, but there are fewer decisions to be taken. We might even eliminate the "if" from it; the algorithm is just a series of actions to be taken by someone who imagines a France with a King. This is natural: the whole operation is like an imaginary revision of the Constitution.

The advantage of the algorithmic perspective is that we avoid the approach in which fictional prose is made up only of false sentences.

4.3. Looking for Ghosts

Russell rejected Meinong's objects. There are now many authors who have gone back to Meinong and try to figure out what a logical system that accommodates "round squares" and similar things looks like. Our idea is rather different. Let us look at what happens if we modify the algorithm for the King of France and try to find out round squares.

Let us examine the following algorithm:

Precondition: *The examination of all possible squares.*

While *you have not found a suitable square or you have not examined all the squares* **do**

Check the current square.

If *the square is round* ***then***

You have found a suitable square.

Else

You did not find a suitable square.

End if.

Examine the next square.

End while.

Postcondition: *A report about squares.*

Despite the superficial similarity with the algorithm which checked the situation of all the persons in France, this algorithm has a nasty flaw. The algorithmic process will never halt.

First, if we assume that "squares" refers to normal, usual squares, there is an infinity of such squares and there is no chance to find a round square among them (as Russell pointed out long ago, this would be logically contradictory).

Second, if we enlarge somehow the set of possible squares with some unusual squares, there would be no procedure to check those squares. You have to believe in some unusual checking procedures if you say that you check such squares. This would push you out of the realm of mathematics into the world of linguistic tricks. After all, those who believe that "round square" refers to some kind of object do not believe that such objects subsist.

It is impossible to halt the above algorithm, it loops for ever. The algorithm is obviously wrong. There is nothing however that prevents us from talking about this algorithm.

Would it be possible to use the same strategy as in the case of the King of France and examine the "constitution"? This would engage us in a debate about the "logical legislation" that applies here. If we stick to good old classic logic, "round square" leads to a contradiction and it should be avoided. If we adopt some fancy logic, the situation might look quite different, but it is not the aim of this paper to engage in such debates.

One way or another, the solution that we suggest is: talk about the algorithm! Do not bypass the algorithm and try to discuss the denotation as an object in itself. The denotation makes sense only if we can (or we cannot) illuminate it with the help of some algorithm.

4.4. What Did the King Want to Know?

According to Bertrand Russell, philosophical theories are tested using logical puzzles. The use of puzzles resembles the use of experiments in natural sciences. The capacity to deal with puzzles is similar to the capacity of a theory in physical science to pass empirical tests.[28] Let us now consider the famous sentence proposed by Bertrand Russell:[29]

George IV wished to know whether Scott was the author of *Waverley*.

According to the approach adopted here, the denotation of a (declarative) sentence is a situation. George IV offered a series of clues concerning a certain situation. First is the name "Scott." Then "the author of *Waverley*" contains clues for an algorithm. This algorithm computes a denotation. King George IV wanted to know whether the result of this computation is the same as the denotation of the name "Scott." What happens with the puzzle? The logical difficulty appears when we substitute "the author of *Waverley*" with "Scott." The substitution is based on the assumption that both the denoting phrase and the proper name have the same denotation. However, the focus in the King's question is on the algorithm computing the denotation of a sentence. The name "Scott" raises its own problems. This paper has no intention of discussing them. We assume that, in the puzzle, the King already knows how to compute the denotation of the name "Scott." He might have known that Scott wrote *Ivanhoe* or something similar. This seems to be Russell's opinion. He might have known how to use the label "Scott" as a clue in order to find a certain individual.[30]

In the algorithmic interpretation, the puzzle involves a logical error. The question is *about* an algorithm. Does the algorithm lead to a certain result? The denotation of the sentence "Scott was the author of *Waverley*" is not in focus.

What happens if we interpret the whole substitution as the replacement of an algorithm with another algorithm? The logical mistake is still there. We may replace one algorithm with another algorithm if we are just interested in getting the same result in a different way. The replacement is wrong if we want to know where the algorithm is leading to. The substitution begs the question.

Somebody may object to the approach above and wonder if the restrictions on substitutions are not too drastic. Actually, they depend on whether the denotation or the algorithm is in focus. Consider the following sentence:

X wants to know when the author of *Waverley* lived.

Here the denotation is in focus and, if "the author of *Waverley*" has the same denotation as the name "Scott," the substitution is possible. X implicitly wants to know when Scott lived, even if she would not know that the author of *Waverley* was Scott.

It is a perfectly benign strategy to answer a question like "when did the author of *Waverley* live" by making first a substitution. Then one may look into an encyclopedia and come up with a time interval: 1771–1832.

In other contexts, the attitude towards the denotation puts indirectly in focus the algorithm. Let's consider the following belief of X:

X believes that Scott is the author of *Waverley*.

The denotation of the sentence introduced by "that" is a situation. The replacement, for example, of "the author of *Waverley*" with "the author of *Ivanhoe*" changes the situation; it also changes the algorithm which computes the denotation of the sentence, thus putting the algorithm in focus. When X believes something, she surmises that a certain algorithm works.

The logical analysis as formalization translates vague sentences from the natural language into a symbolic language. It is then easier to analyze the arguments involved.

The scheme of algorithmic analysis follows, up to a point, a similar pattern: it separates the clues which permit the formulation of an algorithm for the discovery of a denotation from the misleading clues.

The approach used here departs from Russell's approach far less than the theories which claim that natural language is all right and needs no translation into a symbolic language. Summing up, the approach we would favor is the following: do not look for the meaning, analyze the linguistic clues and the algorithms they support.

5. Two Kinds of Denotation

The algorithmic approach seems to blatantly contradict the flexibility of the natural language. This is not true, at least if we treat linguistic expressions as sets of clues, which are obviously very flexible. Next, the relationship with the computer should not suggest something "mechanical"; algorithms capture patterns of actions and then they might be implemented on a computer.

On the other hand, what happens if we restrict ourselves to clues? The rejection of the algorithms obscures the differences between different kinds of denotations.

5.1. Old Ideas in New Clothes?

The approach suggested here seems to go straight into a formidable obstacle. Let us say that someone exhorts you to write a good paper. What algorithm corresponds to this imperative?

There is obviously no direct algorithmic translation for "write a good paper!" because we cannot specify exactly the steps of the respective action. But this does not prevent us from *understanding* the imperative. First, it is (relatively) easy to use clues in "write a paper" for designing a sequence of actions that leads to the production of a paper. What is going to count as a *good* paper is much less definite. We think that branching becomes inevitable in this case. There are many ways in which a paper could be good.

5.2. Effective vs. Presumed Denotation

An algorithm for discovering a denotation is deeply connected with further actions that might be performed. Let's go back, for example, to the shop around the corner (the real one, not the one from the movie). Talking about the shop is, normally, not mere description; if books are sold in the shop, further actions might be envisaged (for example, buying a novel by the 2012 Nobel Prize winner Mo Yan). Is it possible to build an algorithm for all denotations? It depends. The clues contained in the linguistic expressions might seem to indicate a denotation, but—as we saw above—be very problematic. Let's say that someone talks about the "soul of the Moon." The clues are there, but how are we going to build the algorithm? The Moon does not talk, but the proponent might insist that its movements are a kind of language. An opponent would point out that the supposed algorithm is merely an extension of the previous talk about a soul of the Moon; it does not bring forth any idea behind the interactions with the soul of the Moon.

In contrast with the denotation of "the soul of the Moon," the denotation for "the Moon" is not only presumed to exist, but there are various algorithms for finding the celestial body near the Earth. In the second case, the denotation is an effective denotation.

One thing that has to be noted in the above example is that "the soul of the Moon" is not construed as a poetic expression; the assumption is that its denotation is on a par with the denotation of "the Moon." But it is possible to formulate at least one algorithm for finding the denotation of "the Moon"; it is this feature which makes the denotation of the expression "the Moon" an effective denotation. ❑

Notes

1. See Grim et al. (1998, 9–10).
2. See Suzanne M. Marchese and Francis T. Marchese, "Plato's Cave: an Image Stream Installation within an Office Setting," in *Proceedings of the Eighth International Conference on Information Visualisation* (London: IEEE Press, 2004), 954–958. The authors explicitly mention Plato's Cave (in section 7.1); they view their artificial windows as similar to the shadows in the Platonic cave.
3. Robin Waterfield's translation (Oxford: Oxford University Press, 1993), 240–241. Waterfield observes that "the shadows are cast by things which are themselves effigies of real things" (424).
4. See Skiena (1998, 3).
5. Knuth (2000, §1.1) first highlights the *finite* and *clear* character of an algorithm. An algorithm has input(s) and output(s). The output(s), the result(s) of the algorithm correspond to the value(s) returned by the algorithm. Knuth also emphasizes the significance of *efficiency*: the algorithmic process (the transformation of the input into the output according to the algorithm) must lead to a value in a reasonable time interval. The finite character of the process is not enough.
6. Dennett's idea is that all design in the universe is the result of "an algorithmic process that weds randomness and selection." Intelligence is the product of such an algorithmic process, not the originator of this process.
7. For many details on the use of algorithms in logical semantics see Szymanik (2009), especially chapter 1 "Algorithmic Semantics."
8. Computer programs are just texts. See Moschovakis (1994, 212).
9. See Moschovakis (1989).
10. Moschovakis writes that "the idea that the sense of a sentence determines its denotation is at the heart of the Frege doctrine and practically everyone who has written on the matter uses some form of computational analogy to describe it" (Moschovakis 1994, 216).
11. See Tichý's arguments in his 1971 paper.
12. Tichý's dual strategy for the formalization of natural language is clearly delineated in his paper on intensional analysis (Tichý 1971, 282).
13. Russell writes that "if I say 'the author of *Waverley* was a man,' that is not a statement of the form '*x* was a man,' and does not have 'the author of *Waverley*' for its subject" (Russell 1905, 488).
14. See Russell's paper (1905, 481).
15. Cf. Russell (1905, 489).
16. Russell (1905, 484) expressly stresses the idea that the sentence is false. It is not nonsense.
17. Recent introductions into the philosophy of language, like Devitt and Sterelny (2000), still provide ample space for Russell's theory and its criticism.
18. For critical comments see, for example, the introduction to the philosophy of language by Devitt and Sterelny (2000, chap. 3).

19. Strawson (1950, 333).
20. Russell wrote that he was "unable to see any validity whatever in any of Mr. Strawson's arguments" (Russell 1957, 385). Russell points out that he disagrees with philosophers who are "persuaded that common speech is good enough not only for daily life, but also for philosophy"; he is "persuaded that common speech is full of vagueness and inaccuracy, and that any attempt to be precise and accurate requires modification of common speech both as regards vocabulary and as regards syntax" (Russell 1957, 387).
21. The novel was first published anonymously; thus Scott was later identified as "the author of *Waverley.*"
22. Of course, we suppose that this is a *real* situation. In the 1940 movie about the shop around the corner, the situation is *imaginary* (the story is about an imaginary gift shop).
23. This is obvious from the start of Russell (1905, 479). Russell comes back to the significance of the knowledge without acquaintance in the final part of his paper (493). The fact that knowledge is in focus frames the whole study.
24. In the tradition inaugurated by Wittgenstein, one would talk about a change of focus from meaning to use. Our intention, though obviously parallel, is however rather different. We still think that Russell's insistence on the tricky nature of the natural language and the need to go beyond its surface should be taken seriously.
25. We use pseudocode, a mixture of natural language and special notations in order to formulate the algorithms. It is a technique borrowed from computer programming.
26. Moschovakis (1994, 225) writes about the "programmer's solution" to Russell's problem. He asked four friends the following question: *do you think that the King of France is bald?* Three of them answered "there is no King of France." We interpret this as a focus on preconditions. The problem with this answer is that it misses the point: which is the truth-value of the sentence "the present King of France is bald"?
27. Anything we say in sentences in which "the French author who writes in Ancient Greek" has a primary occurrence; i.e. is a direct ingredient of the sentence. Russell (1905, 490) stresses the importance of the distinction between primary and secondary occurrences (occurrences in sentences that are components of a complex sentence). He admits that the distinction is not quite clear in the case of natural languages. It is easy to make it in symbolic languages.
28. See Russell (1905, 484–485).
29. Russell (1905, 485).
30. John Stuart Mill compares such an operation with the chalk mark that thieves in *1001 Nights* leave on Ali Baba's house. Morgiana confuses them by making the same sign on all the houses in the neighborhood. The thieves are lost since they use the mark as a clue (that plays a key role in the algorithm they use for finding the house). See John Stuart Mill's *A System of Logic*, Book 1, chapter II, §5 (we have used the French translation of the sixth edition, *Système de logique*, vol. 1 [Paris: Alcan, 1889]).

References

Dennett, Daniel. 1995. Evolution as an algorithm. In *The Mind, the Brain and Complex Adaptive Systems*, eds. Harold Morowitz and Jerome Singer, 221–223. Boston: Addison-Wesley.

Devitt, Michael and Kim Sterelny. 2000. *Limbaj şi realitate*. Translated by Radu Dudău. Iaşi: Polirom. Romanian translation of *Language and Reality: An Introduction to the Philosophy of Language*. Oxford: Blackwell, 1999.

Grim, Patrick, Gary Mar, and Paul St. Denis. 1998. *The Philosophical Computer*. Cambridge, Massachusetts: The MIT Press.

Knuth, Donald E. 2000. *Arta programării calculatoarelor*. Vol. 1. Bucharest: Teora. Romanian translation of *The Art of Computer Programming*. Vol. 1, *Fundamental Algorithms*. Reading: Addison-Wesley, 1998.

Moschovakis, Yannis. 1989. The formal language of recursion. *The Journal of Symbolic Logic* 54(4): 1216–1252.

——. 1994. Sense and denotation as algorithm and value. In *Lecture Notes in Logic*. Vol. 2, eds. J. Oikkonen and J. Vaananen, 210–249. New York: Springer.

Russell, Bertrand. 1905. On denoting. *Mind* 14(56): 479–493.

——. 1957. Mr. Strawson on referring. *Mind* 66(263): 385–389.

Skiena, Steven S. 1998. *The Algorithm Design Manual*. New York: Springer.

Strawson, Peter Frederick. 1950. On referring. *Mind* 59(235): 320–344.

Szymanik, Jakub. 2009. Quantifiers in TIME and SPACE: Computational Complexity of Generalized Quantifiers in Natural Language. Ph.D. Thesis, University of Amsterdam.

Tichý, Pavel. 1971. An approach to intensional analysis. *Noûs* 5(3): 273–297.

Abstract

An Algorithmic Perspective on Denotation

The paper focuses on the algorithms we use for the discovery of denotations; these algorithms are built by exploiting clues contained in the expressions formulated in natural language. The paper analyzes examples from Russell's "On denoting" (1905). It stresses the idea that algorithms do not function as meanings. They must be considered from the perspective of the actions of the persons who use a natural language. In conclusion, the paper proposes a distinction between effective denotation, the output of an algorithm, and presumed denotation, the result of mere talk about linguistic clues.

Keywords

analytic philosophy, language, clues, computer, algorithms, actions, denotation