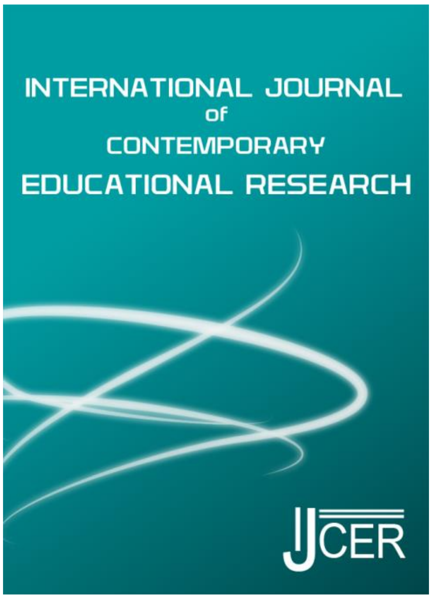

International Journal of Contemporary Educational Research (IJCER)

www.ijcer.net

Reflections on the Importance of Reference for Understanding Thinking

Reinhard Oldenburg[1]
[1] Augsburg University, Germany

To cite this article:

Oldenburg, R. (2015). Reflections on the importance of reference for understanding thinking. *International Journal of Contemporary Educational Research, 2*(1), 54-60.

International Journal of Contemporary Educational Research
Volume 2, Number 1, 2015 January, Page 54-60 ISSN: 2148-3868

Reflections on the Importance of Reference for Understanding Thinking

Reinhard Oldenburg[1*]
[1] Augsburg University, Germany

Abstract

An intentional form of the semantics of algebraic expressions in the tradition starting with Frege is popular in mathematics education. On the other hand, mathematical logic including predicate calculus and lambda calculus is dominated for more than 50 years by referential semantics. A third field of investigation is the semantics of mathematics as realized in programming languages and computer algebra systems. The paper explores the tension between these approaches and tries to clarify the role of reference both in he developed mathematics as well as in the learning process. Referential semantics simplifies theories but requires mental objects to be constructed to be useful. This links the topic to reification theory. A small collection of observations of learners' behavior adds support to the claim that reference is of some importance in the learning process.

Key words: Semantics, algebra, variables.

Introduction

The semantics of logic has been a subject of change for a long time. At least for mainstream mathematics this process has reached a stable state with the works especially of Tarski. Predicate calculus together with set theory has shown to be a powerful combination that fulfills the needs of mathematicians and moreover of computer scientists. In the present paper we investigate the fundamental role of reference in this theory of the semantics of logic language. A brief outline of this is described in section 1. Yet, in mathematics education it is more common to refer to the older semantical theory by Frege. Notably Arzarello et al. (1994, 2001) have dealt with this in detail and came to the conclusion that his intentional semantics is well suited for education (section 2). On the other hand, the modern referential view of logic has advantages in giving a short and concise description of the formal background of mathematics (section 3,4), but may be dangerous as model for what students should be presented (section 3). Hence, the central question that should be clarified is the relation between understanding and reference (section 5). We collect some simple observations that might help in giving an answer to this question (section 6), but find that further research is needed to clarify the situation.

Predicate Calculus, Lambda Calculus

The notion of reference plays a central role in modern formal mathematics (see e.g. Li 2010). In predicate calculus, formulae are build up from symbols for variables, functions and predicates, logical conjunctions (not, and, or, implication, …) and quantifiers (for all, exists). The well-formed formulae are described on a purely syntactical level. Meaning is given to them by fixing an interpretation that consists of a domain and an assignment of a value from this set to every free variable in a formulae and of functions and predicates over this domain for every predicate and function symbol that arises in this formula. Focusing on variables, an interpretation is thus a set of references from variables to the domain of the theory. If a formula is true for some, all or none interpretations applied to it, it is called satisfiable, tautology, or contradictory, respectively. The role of variables is to refer to objects from the domain. All variables are equal, but they may play different roles, depending on the quantifiers applied to them. But even then their function is to refer to an object.

Note that in each interpretation, the variable is assigned a unique object that is fixed in the interpretation. One may write this assignment as $v \to a$ (here v is a variable (i.e. a part of the logical meta-language) and a is an object of the domain). An interpretation consists of exactly one such assignment for each variable. Evaluating if

[*] Corresponding Author: *Reinhard Oldenburg, reinhard.oldenburg@math.uni-augsburg.de*

a formula is true or false in a given interpretation amounts to applying all these assignments as substitution rules and then checking in the domain if a statement is true. In general we use the term evaluation (w.r.t an interpretation) for the process of determining the object of the domain that an expression refers to.

Note that understanding change on the level of this formal language requires either the introduction of two variables to capture initial and final states or that of a function of formalized time, or, on the Meta level - the consideration of many interpretations.

Of course, this whole story only makes sense if the domain is well-understood. A prototypical example is arithmetic, where an axiom system assures that only domains are possible that are isomorphic to the standard set of integers. To work successfully in this logic theory one has to understand this set of integers, i.e. one needs to be able to decide the truth of predicates (such as 3|12) over this set. What the logic adds to purely having the integers as mental objects is the ability to make general statements over this set.

We conclude with a short look at lambda calculus which is an equivalent logical system that is essentially suited for understanding computation (Michaelson 2011). It is so to say the calculus of pure functions and used e.g. as a theoretical basis of functional programming languages. Variables appear as parameters of functions. Their role is to refer to the input given to the function. This short description is enough for our purpose as we need only the conclusion that with regard to variables, both logic systems, predicate and lambda calculus, don't differ much: Variables refer to something.

Semantics based on Frege

Arzarello et al. (1994) and many mathematics education researchers in the sequel work with an intensional semantics of algebraic expressions that derives from Frege's theory. Especially they draw on Freges distinction between sense and denotation (Frege used the German word 'Bedeutung', but it would be misleading to translate this as 'meaning', and most translators avoid this). His most cited example is that of the expressions 'venus', '*morning star' and* 'evening star'. They have, according to Frege, the same denotation (i.e. they refer to the same object) but their sense (i.e. all the connotation these expressions may have for you) is different. Thus, when evaluating an expression using a replacement $a \rightarrow b$ we may lose some information (namely the sense of a) when substitution b for a.

Now, we focus on the application of this to understand the use of the algebraic language. In the words of Arzarello who develops Frege's theory into more detail, one may say that the expressions $n(n+1)$ and n^2+n have

1. the same **denotation** (or the same reference), namely a function in n
2. different **algebraic senses**, i.e. one is expanded, the other factored. The algebraic sense “represents the very way by which the denoted is obtained by means of computational rules”.
3. different **contextual senses**, e.g. the first may have the sense of the area of a class of rectangles or it may be the product of consecutive integers.

This distinction is appealing because it can be used to explain certain problems of students, e.g. as being the result of students mixing up these levels or neglecting one.

Interestingly, the denotation, which might seem to be the easiest to characterize, proves to be delicate and is handled differently by authors who base their presentation seemingly on the same theoretical grounds. First note that there is some arbitrary choice made: One might have expected $n \cdot (n+1)$ or n^2+n to be a polynomials in n rather than functions. However, (Arzarello et al 1994, p. 110) says that functions are denoted: “For example, the expressions $4x + 2$ and $2 \cdot (2x + 1)$ express different rules (senses) but denote the same function.” Taking the denotation as functions may not be what one wants and on the next page they state “The denotation of a symbolic expression in algebra is the numerical set, possibly empty, which is represented by the expression itself.” In this understanding, the denotation (when working over the reals) of $n \cdot (n+1)$ and n^2+n would be the semi-open interval $[-\frac{1}{4}, \infty)$. Moreover, following this line of thought one has to say that x and $x+1$ have the same denotation when working over the integers or reals (but different denotations over the natural numbers).

The problems to define denotation in a coherent fashion seem to be rooted deeply in Frege's theory. We continue to explore these difficulties by focusing on the somewhat different presentation given by Drouhard & Teppo (2004) that draws on Frege as well. They put out, consistent with Arzarello et al. in the first of his explanations of denotation) the interpretation that $2 \cdot x+2$ denotes a function $\mathbb{R} \to \mathbb{R}$ and that $2 \cdot (x+1)$ denotes the same function (although they differ in sense). Furthermore, $2 \cdot x+2=1-x$ denotes a function from numbers to Boolean values $\mathbb{R} \to \{true, false\}$.

There are two severe problems with this approach:
First problem: Because 4+1 and 3+2 denote the same number, one may replace one for the other without affecting the truth of sentences, e.g. „4+1 is a prime“ will stay true after the substitution. On the other hand, the statement „$n(n+1)$ is a factored polynomial“ will become false upon replacing the expression by n^2+n. Thus, following Frege's intentional semantics, we must say that the truth of statements is not affected alone by what objects expressions refer to, but also what sense they are attributed. This is problematic, because 'sense' is a notion that is in itself not clearly defined. Thus, doing substitutions is a dangerous operation that requires deep analysis each time it is to be carried out – although it is a very frequent operation in mathematics. Quine (1960) has analyzed such situations and called them opaque contexts – which should be warning enough to chase students in this direction of thinking.

Second problem: Algebraic language is context free in the following sense of formal languages (i.e. context is not meant to be a real world context): The rules of the language apply to sub-expressions independent of their position in larger expressions, e.g. is you know that $a+b$ equals $b+a$ then you may replace one of these expressions with the other, independent of its position in some larger expression. And if you know that n refers to a number, then in $n+1$ the n refers to a number as well. This context freeness obviously simplifies a language very much and is reflected nicely in the view that algebraic expressions are trees where each leaf is a tree in itself that can be arbitrarily complex. In the Frege tradition, however, we are told that $x+1$ and $2 \cdot x-1$ are both functions R→R and these functions are different, so that we must conclude that $x+1=2 \cdot x-1$ is a false statement because the objects on both sides of the equal sign differ. To overcome this, and allow for equations to be solved for unknowns, Frege followers must scarify context-freeness. They have to claim (and Drouhard&Teppo do this explicitly), that in $x+1=2 \cdot x-1$ the part $x+1$ does not reference a function, rather as a part, it references nothing, but the whole refers to a function to the Boolean values. Thus, students can't learn one simple rule of what $x+1$ refers to, but they have to ask in each new context, what sense it has there. Thus they are urged to ask the question, Am I allowed to do this *here?*”.

To summarize: Taking Frege's intensional semantics as a basis of algebra gives the vague but useful notion of sense but comes at the cost of leaving denotation unclear and seemingly precise statements are bound to use the vague notion of sense as well.

Reference for simplicity!?

Within logic and moreover lambda calculus, many programming languages and so forth it has been realized that the complexity of the semantics of formal systems is reduced and streamlined if you assume that all symbols refer to something in a unique way. This does not only apply to variables for objects from the domain but also for function and predicate symbols. For example, one may hold the view that (depending on the domain) the equal sign = is not just a syntactical mean like parentheses but refers to something, namely a function from $\mathbb{R} \times \mathbb{R}$ to the set {true, false}. This point of view is very clearly articulated in one of the programming language with the slimmest yet most powerful semantics, namely Scheme (Abelson et al. 1998). In this language the equal predicate that the sign = refers to can handled like any other object (e.g. stored somewhere, passed as parameter). To be more precise, entering an expression in this language gives its evaluated value, e.g. entering `5` gives 5 and entering `(+ 5 2)` gives 7. After `(define a 3)` we have that `(+ a 1)` gives 4. The interesting thing is that entering = gives a procedure, e.g. in the racket implementation this prints out as `#<procedure:=>`. The symbol = is just a name. A standard name that comes defined in the initial environment while `a` in the example above was a name introduced by the user. Thus even the equal predicate (and as well operations such as +) are used in the same referential setup: These are symbols that refer to a procedure. However, not every part of the language has a reference – a single `(` does not have. Nevertheless, this approach greatly simplifies things: Symbols refer to something and that’s the value used upon evaluation.

So, a consequence could be to set out the goal to base school mathematics on such a simple and consistent referential semantics that is successful in a vast variety of areas. Every symbol used is then to be understood as a reference to some object and the logic of quantifiers on them may set students in the position to master all mathematical questions in a consistent, simple semantics system.

However, although logically consistent, such an attempt could very easily prove to fail. Its simplicity draws on the following facts that are given for most adult users and creators of mathematics but seem very questionable when looking at learners:

1. The domain of objects *S* needs to be clearly understood. As a quantification „for all *x* in *S* we have ….“ needs an overview of what the elements of *S* are. Yet learners may not have constructed the objects mentally as their own objects, and even if they have constructed some or all of them, it may turn out that the learner is still missing the overview of this domain.
2. Even if the domain of objects is mastered, it may be the case that functions on this domain are not yet constructed as mental objects, e.g. the learner may still have the process view without having it reified yet. E.g. it may be that learners can use the definition $f(x)=2\cdot x+1$ of a function N→N to calculate function values, but they may fail to see *f* as an object. Especially they may fail to see that the part *f* of these expressions refers to something. That learners often accept or produce such writings as $f(x)=n+1$ (thus not linking the variable in both sides of the equation) shows that they view this more like a ritualized way to express a calculation procedure than as an expression composed of parts that have individual meaning by their reference to some objects.

So, from his considerations it is not yet clear, what the relevance of reference is, expect that is desirable. We will explore some more aspects before trying to put bits together.

Technology: Computer algebra systems

The last section has already alluded to a connection to technology and this will be addressed in more detail in this section. When you enter an expression in a computer algebra system (see e.g. Davenport 1988) it builds up a certain structure in the computer's memory that we call an object (it may be a number, a list, a polynomial or some other thing). If one assigns the expression to a variable, then the variable refers to it and in evaluation will henceforth give this object. Thus, we have a referential structure as the working model of such a system. However, it would be absolutely infeasible to have objects used in the way suggested by Frege's theory. Instead, for all modern systems the expression $n\cdot (n+1)$ and n^2+n denote two different objects. Thus, sticking with Frege's theory would mean to explain students, that these expressions denote the same objects, but that the CAS does treat them as different. Or, put differently, it hinders students in synchronizing their mathematical objects and operations with that of the system, and as a result they are not able to learn from the system as a model of correct mathematical behavior. The only way out would be to say that the CAS deals with the sense of the expression, not with its denotation. But certainly, a computer can't work with the contextualized sense, at best with the algebraic sense. This notion is left a bit vague by Arzarello and Drouhard, at least it seems to be not so clearly defined as to become a criterion for checking if a computer algebra system works correctly. For example, one may ask if 1 seen as a number and 1 seen as a constant polynomial have different algebraic sense – if so, it can't be detected by standard computer algebra systems (although, there are strongly typed systems that make this distinction). To say that CAS works with algebraic sense would then require to say that the algebraic sense is identical. The same applies to examples such as $1+x$ as being either a polynomial or a rational function with unit denominator. One of the explanations Arzarello et al. give is that "it represents the very way by which the denoted is obtained by means of computational rules.". But as most CAS treat $x+y$ and $y+x$ as identical objects they can't represent the difference of the algebraic sense between these two writings. I think that the only definition of algebraic sense that is compatible with the use of computer algebra would boil down to say that two expressions have the same algebraic sense if they are represented in the CAS by the same object. But this would eliminate the need for sense as it replaces it with denotation.

Given these problems, I suggest that the easiest way to deal with the problems is to adopt for mathematics in general the way algebraic objects are dealt with in computer algebra systems. This is not to be mistaken as saying that technical decisions of computer algebra makers should have normative power for teaching mathematics. In the contrary, the science of computer algebra systems has established, that the best way to build such systems is to stick close to the ideas of formal mathematics such as predicate and lambda calculus. Thus, supporting a completely different view of mathematical objects would mean to increase the distance between school mathematics on one side and logic, mathematics and computer algebra on the other side at the same time.

To summarize the joint result of the last three sections: We propose that writings of compound signs such as $2\cdot x+1$ or $2\cdot x+1=3\cdot x$ refer to mathematical objects which are expressions. Thus, besides standard domains like integers, rationals and polynomials we also consider expressions as a valid domain in predicate calculus. These expressions may – if needed – further be mapped to specific domains such as polynomials, rational functions etc. but as they are created by writing them down $2\cdot(x+1)$ and $2\cdot x+2$ denote different objects. By mapping them to specific other domains, such as polynomials, different expression objects may be mapped to the same polynomial object. This allows for a fine grained understanding of identity. In the large domain of expressions we can attribute properties like 'expanded' to these objects rather than to have the need to speak about the vague notion of sense. This is a consistent and clear referential view of mathematics, compatible with mathematical logic and computer algebra systems. Yet, it has to be discussed, if this view is adequate for learners of mathematics. There are some subtle points to be clarified but first let's look at an observation that illustrates that thinking about expressions may be rather close to students intuitive conceptions:

> For instance, a child asked by an interviewer to write down the length of a space-ship's path composed of *y* 1l-light-years long segments said: "What, shall I write what I would do?"; and after she eventually contrived the formula $11\cdot y$, she exclaimed to the interviewer: "What, is that all it was? Why didn't you say so? I thought you wanted an answer." Thus, for this child the expression was a mere prescription for the sought-for quantity, not the quantity itself. (Sfard&Linchevsky 1994, p. 207)

This shows that this student feels comfortable with $11\cdot y$ as an expression but struggles with Frege's view that this denotes a number.

Understanding by form and by reference

As explained above, modern logic and mathematical software and programming language are extensively based on the idea of reference. When a new mathematical subject is created, researchers define their new domain of objects to be able to use the referential semantics of logic. The assumption that such objects exists at least relatively to some frame (ontological commitment in the language of Quine (1960)) is the first step in doing mathematics. As mentioned at the end of section 3, this view may not be adequate for learners who have not yet constructed the mental objects that variables should refer to. Consider, as an example, functions. There are various theories that describe the learning process of functions. One theory that is prominent in Germany is the theory by Vollrath who describes four steps in the learning of the function concept. Only the last one considers function as objects. A three step theory by DeMarois (1998) similarly puts functions objects on the highest layer. So, if these theories are correct, there must be a way for students to gain meaning other than the referential semantics of developed mathematics. How this way may look like is a difficult question. Besides by Frege, non-referential theories were influenced by Wittgenstein's language games. This is certainly attractive for natural languages, but the approach lacks the rigor needed for formalized proofs. For the purpose of understanding learning however, it may be quite adequate. One may say, e.g., that students learn the rule that after $f(x)$ one puts the rule of some calculation, or they may view the vector analytic description of a line $\vec{x} = \vec{a} + t\vec{b}$ as a ritualized way to give a point on a line and the direction of a line without seeing any referential meaning (of *x* standing for a vector to a point on the line or of seeing the whole as a logical statement with free variables that can be used, e.g. to substitute a vector for *x* and determine if there is a solution for *t*, i.e. if the point is on the line.). Such language games allow processes to take place and thus they form the basis of reification and the creation of mental objects that can then serve as the basis for referential understanding. The roles of diagrams in thinking can also be understood from this purpose. In (Oldenburg 2011) we put out the thesis that the inscriptions used by some programming languages are ideally suited to support the creation of mental objects. The diagrams are thus means to provide objects.

This gives a perspective on what might be a sensible didactical approach: Language games may be good starting points for the novice but ideally they are structured in a way to ease the creation of mental objects and form the domain of a referential understanding. The various forms of process-object theories such as reification theory (Sfard&Linchevski 1994) and procept theory (Tall 2012) may inform on how this object creation may take place. So the conclusion of this very sketchy paragraph is that successful learning is likely to happen if it is geared towards the creation of mental objects.

Then, how important is reference really for understanding?

Here we collect some hints, that reference is an important key in student's use of algebraic formalism. The first observation here is one from Meyer (2013). He led 11th grade students solve a problem of a number triangle, which is a complex arrangement of 7 numbers obeying several rules like that the sum of two adjacent numbers must be the number in a neighboring field. The details are not important, but it is of interest to look at the transcript (which was found to be interesting by Meyer for completely other reasons) and ask what activities triggered the use of a symbolic variable as a problem solving tool. The student Frank interacts with another student and they talk a lot about numbers, experiment and try out, but they never use any algebraic concept. Then the interesting break in thought happens:

> Frank: "...This is two times 7, basically (writes 2·7 above the number triangle). And here we have one times 9 (writes 9 into the lower outer field) [...] and here (*points at the left inner field*) we have one part, *I mean, one x of 9*, I mean, times *x*, I don't know how to put it, say *x* from 9, and ..." (He intended 9-*x* and finally got to this. Highlights (italics) by RO)

The use of a symbolic variable was thus triggered by the embodied action of pointing to a place and this brought up the use of a variable. Thus, at least in this case, referencing is deeply linked to using variables in a sensible way!

Another (this time negative) example is based in the area of analytic geometry. Certain curricula define vectors geometrically as equivalence class of arrows of the same length and direction. As these objects (equivalence classes) are not so easy to construct mentally, many students pretend that the vectors defined by the arrows attached to parallel edges of a cube to be different. Thus, they often fail to setup adequate vector equations to solve problems, because of the inadequate referential system they have set up.

A third example is given by the composition of functions, especially in the case of function and inverse function. As long as there is no reference to functions as objects, students hardly make sense of a recipe for finding the inverse function such as interchanging *y* and *x* and isolating.

The conclusion here is that referential understanding is at least of some importance for the learning of math and the lack of referential understandings may be an obstacle. However, further research is needed to allow more specific statements.

Conclusion

To get access to the advantages of referential semantics students must construct mental objects to refer to. Thus, it is an important question of educational research how to foster the development of concepts. There are some important contributions in this direction: As mentioned above Sfard, Dubinsky, Tall and others have developed various theories of how objects arise from processes and Dörfler (2005) and others have looked into how inscriptions of mathematical symbols may actually be the mathematical objects that are dealt with. These are important contributions that one can build on (see e.g. Oldenburg 2011) but still establishing a domain to be suitable for use as the domain of a referential logic theory is a lot of work: One must define (and create) the objects and one must be able to decide identity of objects. This is a non-trivial task and students may fail to do cope with this. We think that further research if necessary to clarify how objects are created and to what extend this is a necessary precondition for understanding (or for particular forms of understanding). Especially the paper urges the math education community to rethink if Frege's semantics is an adequate foundation, especially if technology is used in the learning process.

References

Abelson, H., Dybvig, R. K., …., Wand, M. (1998). *The revised^5 report on the algorithmic language Scheme*. MIT 1998. http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html

Arzarello, F., Bazzini, L., Chiappini, G. (1994). Intensional semantics as a tool to analyze algebraic thinking. *Rend. Sem. Mat. Univ. Poi. Torino*, Vol. 52, 2.

Arzarello F., Bazzini, L., Chiappini, G. (2001). A model for analyzing algebraic processes of thinking. In R. Sutherland et al. (Eds.), *Perspectives on School Algebra*, pp. 61-81. Dordrecht: Kluwer Academic Publishers.

Davenport, J., Siret, Y., Tournier, E. (1988). *Computer Algebra: Systems and Algorithms for Algebraic Computation*. London: Academic Press.

DeMarois, P. (1998*) Facets and Layers of the Function Concept: The Case of College Algebra*. PhD thesis, University of Warwick, England.

Dörfler, W.(2005): Diagrammatic Thinking. In: M. Hoffmann (Ed.) *Activity and sign – Grounding mathematics education*. Berlin: Springer.

Drouhard, J.-P. & Teppo, A. (2004). Symbols and Language. In K. Stacey, H. Chick, M. Kendal (Eds.) *The Future of the Teaching and Learning of Algebra: The 12th ICMI Study*. (Pp 226 - 264), Dordrecht: Kluwer.

Li, W. (2010). *Mathematical Logic*. Berlin: Birkhäuser.

Meyer, A. (2013). Algebraic thinking and formalized mathematics – formal reasoning and the contextual. Paper presented at Cerme 2013.

Michaelson, G. (2011). *An Introduction to Functional Programming through Lambda Calculus*. New York: Dover.

Oldenburg, R. (2011). Reification and symbolization. *Proceeding of Koli Calling 2011*. ACM Digital library.

Quine, W. v. O. (1960). *Word and Object*. Cambridge: MIT Press.

Sfard, A., Linchevsky, L. (1994). Gains and pitfalls of reification. *Educational Studies in Mathematics*, 26: 191-228.

Tall, D. (2012). *How Humans learn to think Mathematically*. Cambridge: Cambridge.

Vollrath, H.J. (1989). Funktionales Denken. *Journal für Mathematikdidaktik* (JMD), 10, 3 – 37.