Filozofia Nauki (The Philosophy of Science)
ISSN 1230-6894 e-ISSN 2657-5868
2023: 1-15 (*online first*)
DOI: 10.14394/filnau.2023.0002

AARON LERCHER*

CURRY'S CRITIQUE OF THE SYNTACTIC CONCEPT OF FORMAL SYSTEM AND METHODOLOGICAL AUTONOMY FOR PURE MATHEMATICS**

Abstract

Haskell Curry's philosophy of mathematics is really a form of "structuralism" rather than "formalism" despite Curry's own description of it as formalist (Seldin 2011). This paper explains Curry's actual view by a formal analysis of a simple example. This analysis is extended to solve Keränen's (2001) identity problem for structuralism, confirming Leitgeb's (2020a, b) solution, and further clarifies structural ontology. Curry's methods answer philosophical questions by employing a standard mathematical method, which is a virtue of the "methodological autonomy" emphasized by Curry (1951, 1963) and more recently with greater clarity by Maddy (1997, 2007).

Keywords: Haskel Curry, mathematical structuralism, philosophy of mathematics

1. INTRODUCTION

Haskell Curry's student Jonathan Seldin (2011) observes that Curry's view of the nature of mathematical subject matter is best understood as a form of structuralism, similar to other structuralist views (Resnik 1997, Parsons 2008, Hellman and Shapiro 2019). Curry's methods are frequently misunderstood as "formalist," since Curry (1951) himself uses that word to describe his approach.[1]

*Louisiana State University Libraries, Baton Rouge, LA 70806, USA, e-mail: alerche1@lsu.edu, ORCID: https://orcid.org/0000-0002-6994-5833.

**The author thanks John Seldin, Roger Hindley, Hannes Leitgeb, and an anonymous reviewer for *Filozofia Nauki*.

[1]Curry (1951) was written in 1939, but publication was delayed by World War II. Curry states in the Preface, "this monograph represents the views which I held in 1939; it does not represent accurately the views which I would defend right now" (Curry 1951: v).

Curry's mature view can be found in his 1963 *Foundations of Mathematical Logic*. But his mature view is difficult to discern even there. Seldin ascribes the common misunderstandings of Curry's view to several factors, including Curry's idiosyncratic vocabulary. Curry sought to avoid philosophical debate by introducing neologisms to describe his ideas, among which are his distinctive notions of a "formal system," and that of mathematical objects or "obs." Curry attempted to avoid philosophical debate while also giving forceful expression to his views. That combination causes confusion.

Curry's avoidance stems from his strong support of the autonomy of mathematical methodology. Curry's approach to formalization fits among an array of disparate methods (Maddy 1997, 2007, Franks 2009, Simpson 2009) that all apply standard mathematical methods to solve ontological problems, and so minimizing philosophical partisanship. This use of standard mathematical methods to solve philosophical problems can be called "methodological autonomy," to which I will briefly return in the conclusion. Curry claims that his view of mathematics shows how mathematics is "independent of any except the most rudimentary philosophical hypotheses" (1951: 3), and so gives mathematics a "pre-philosophical" independence from metaphysical assumptions (1951: 5).

Curry's criticism of the syntactic conceptions of the formal system is an easy way to explain Curry's actual views. Curry criticizes the syntactic conceptions of the formal system throughout his career and with increasing clarity (1950, 1958, 1963). Curry's method for understanding syntactic systems is to analyze these by means of a formal system in his sense. So, this paper first reviews Curry's criticisms of syntactic ideas, and then in the following section, it presents a rudimentary formal system in Curry's sense. The following section shows how this analysis solves Jukka Keränen's (2001) identity problem for structuralism. Keränen (2001), and more briefly John P. Burgess (1999), pose the problem that structural ontologies appear to induce identities between non-identical elements of structures, if these structures permit non-trivial automorphisms (Parsons 2008: 107–109, Hellman and Shapiro 2019: 58–61).

This paper then presents a term system for unlabeled graphs. Unlabeled graphs arise in philosophical debate over structural ontologies for mathematics as an apparent example of Keränen's (2001) identity problem. But Hannes Leitgeb (2020a, b) finally formalizes unlabeled graphs to prove that Keränen-style identities do not arise for these. The term system in this paper confirms Leitgeb's solution in an easily generalizable way, and follows Curry's (1963) methods of formalization.

2. SYNTACTIC SYSTEMS AS INEXPLICIT

The customary syntactic way of thinking about formal systems views them as systems of character strings, which are classes of equivalent inscriptions, which are already a kind of abstract object (Curry 1963: 15–16). Curry sets aside the nominalistic aims of interpreting these character strings as mereological sums of particular inscriptions, instances of brain activity, etc. The character strings have the smallest components, which are then concatenated. Once one realizes that this concatenation operation needs to be associative, then one has already realized that there is an implicit algebraic structure in how the character strings are built up (Curry 1963: 51–52).

In syntactic presentations of formal systems, character strings are usually used autonymously as nouns for themselves. Customarily, the next step is to give syntactic formation rules for defining terms and formulas. However, due to the implicit algebraic structure of concatenation, this is imprecise and inexplicit. So instead, Curry's concept of an abstract formal system includes explicit operations applying to objects, forming other objects, and generating inductive classes of objects. The abstract objects defined this way are "obs."

In a syntactic presentation of a formal system, there are rules of inference, or transformation rules. Here again, Curry (1963: 64–67) criticizes how autonymous presentation obscures the algebraic structure. Nonetheless, Curry's concept of a formal system likewise includes axioms and rules of inference. But we usually want a formal system to be a deductively-generated class of sentences, a theory, which can be interpreted as having truth values. So, the syntactic approach leads us to focus on only certain kinds of syntactic formal systems, in which all the objects are abstract assertions generated from axioms by rules of inference, which Rudolf Carnap (1959) and others call a "calculus." All syntactic objects then fall under one unary predicate or type, marked perhaps by Frege's assertion turnstile, $\vdash$.

Curry takes an algebraic approach to the semantics of formal systems, and is not interested in the customary set theoretical interpretations. But this is for clarity of language, rather than for the sake of any commitment to constructivism.

Curry points out that some important formal systems are not calculuses, including Church's λ-conversion (1941), and Curry's own combinatory logic (Curry and Feys 1958). In these formal systems, well-formed expressions are nouns, not sentences (Curry 1950: 352). There are three operations for forming λ-terms from other λ-terms (Church 1941: 8–9), but no axioms or rules of inference. Moreover, the basic predicate denotes the binary relation of reducibility between the things denoted by such nouns, where reducibility is an analysis of a single step in an arithmetical or algorithmic calculation (Curry 1963: 51). Yet further, the formal language of λ-conversion does not capture a preexisting informal language, but has a more abstract origin (Curry, Hindley, and Seldin 1972: 5).[2]

Any Gödelized system is also not a calculus system. A Gödelized syntactic system is a system of numbers, such that these numbers represent formulas and deductions, and there is a primitive recursive relation between numbers representing deducibility (Curry 1958: 256–257). According to Curry's conception of abstract formal systems, the Peano axiomatic system and the corresponding Gödelized syntactic system represent the same underlying abstract formal system.

3. ABSTRACT ANALYSIS OF A SYNTACTIC SYSTEM

In order not to get lost in the symbols of a term system, let us first look at an exceedingly simple example. Suppose we think about palindromes in a syntactic way. Then any palindrome has a non-trivial isomorphism to itself, or automorphism, ϕ, mapping the left side to the right and vice versa, and mapping a letter in the middle, if any, to itself. But the left and right sides are indistinguishable by any features "internal" to the palindrome. We intended the automorphism to be non-trivial, but if we rely only on such "internal" properties, then it now seems that for any element a of a palidrome, $\phi(a) = a$. This is Keränen's (2001) identity problem.

In this section, we first define a syntactic formal system for palindromes, and then apply Curry's methods to define a more abstract and more precise system that can serve as an analysis of the syntactic system. The following section shows that Keränen's (2001) identity problem does not arise for a more precise system.

[2] λ-conversion can also be presented as a formal system of axioms and equations in the more usual way (Barendregt 1984, Troelstra and Schwichtenberg 2000, Hindley and Seldin 2008: 69–75) and thus framed in terms of assertions, and it is clearly understandable through syntactic accounts.

In a syntactic system, we suppose there is an alphabet of letters, plus one spacing character, and these form strings (or "words") by concatenation. We need to include the empty string because two mirror sequences of letters, with no letter in the middle, can form a palindrome. In order to use variables for unspecified strings or letters, then we need to think of these as belonging to the language we are using in informal reasoning, or U-language (Curry 1958, 1963: 28–29). Let $(A_i)^*$ be the string A_i in reverse order. Then there are two rules for generating palindromes:

1. Construct A_1 as a string, where A_1 is a letter or the empty string.
2. Given strings, A_1, A_2, construct the string, $A_2A_1(A_2)^*$.

These rules are in the general form of a rule in a syntactic calculus (Curry 1958) or assertional system (Curry 1963: 64–67):

$$\text{If} \vdash A_1, \vdash A_2, \ldots, \vdash A_m, \text{then} \vdash A_0,$$

where the A_i are constructed strings. This treats the syntactic system as asserting each string is constructed by an instance of a rule, indicated by the prefixed $\vdash$. Then palindromes are defined as the inductive class generated by repeated applications of this rule on the given alphabet. That's it.

The syntactic system consists of letters and strings, with a rule for generating strings. The syntactic concept of a rule is the same for generating such strings, seen as words, formulas, or sentences. There is no built-in distinction between the types of objects and of propositions. So it seems from a syntactic view to be appropriate to see the result of any rule as an assertion, symbolized by $\vdash$.

This is so simple that it seems entirely transparent. Yet the concatenation operation is merely assumed to occur in our visual perception, rather than being made explicit by symbolizing it. Moreover, the symmetry in the rule is redundant. This might seem an odd thing to say when the system is meant to generate palindromes. But since there is only the one symmetry, and no further distinction to be made on the basis of operations or rules that behave differently for the left and right sides, the symmetry contains no information. The palindrome system does not really have a geometric structure of having different sides or a half-rotation, or any algebra of negative versus positive operations. Also, how to talk about the empty string clearly? How to designate it autonymously?

So in the abstract palindrome system, for each letter in the alphabet, b_i, there is a unary operation which can be symbolized as an infixed predicate, $X \wedge b_i$. The empty string is a distinct abstract object, a. If association is defined unambiguously with association to the left, so that $A_1A_2A_3 \equiv (A_1A_2)A_3$, then the unary operation can also be written by concatenating b_i to the right of X.

The abstract ob-system, $\mathbf{P_1}$, then, consists of arbitrary atoms, designated by any nouns we want (letters a, b_i will serve), and one unary operation defined by each atom, which inductively generating a class of obs from the atoms. In order to explain the formal system, we need to use U-language free variables, X, Y:

1. Each a, b_i is an atom.
2. Given an atom, construct that same atom as a string.
3. Given string X, construct the string, $X \wedge b_i$.

The inductively generated class of strings is the subject matter of this abstract system. What else would it be? This answers the ontological question: what are palindromes? To be a pure mathematical object is to be an object in an abstract formal system, or what Curry calls an "ob" (Curry 1963: 54). My example is even more trivial than those Curry (1951: 17–27; 1958: 254–261; 1963: 52–56) gives because it is intended solely to illustrate the difference between a syntactic system and an abstract system. In the process of building up the abstract system of palindromes, the palindromic shapes might seem to have been abstracted away, but we need not think of anything having gone anywhere.

In the informal U-language, we make distinctions between nouns and verb phrases, relations, or predicates. So far, we have only nouns. There are not yet any axioms or rules of inference in the abstract system. Suppose we add one relation to the abstract system. This will be written as an infix as $\equiv$. Then in order to state the axioms, we again need to use U-language free variables, X, Y, Z. The axioms for $\mathbf{P_1}$ are in the form of axiom schemes, for every ob, X and Y, where X and Y are free variables:

1. $\vdash X \equiv X$.
2. $\vdash X \wedge (Y \wedge b_i) \equiv (X \wedge Y) \wedge b_i$.

The rules of inference are the usual rules for equality or equivalence:

1. If $\vdash X \equiv Y$, then $\vdash Y \equiv X$.
2. If $\vdash X \equiv Y$ and $\vdash Y \equiv Z$, then $\vdash X \equiv Z$.
3. If $\vdash X \equiv Y$, then $\vdash X \wedge Z \equiv Y \wedge Z$.
4. If $\vdash X \equiv Y$, then $\vdash Z \wedge X \equiv Z \wedge Y$.

Then associativity can be proved for this abstract system, $\mathbf{P_1}$.

Curry (1958) proceeds to show how to transform a syntactic calculus system into an abstract one with obs constructions isomorphic to constructions of strings by syntactic rules, and conversely how to transform an abstract system into a calculus. A syntactic system is "monotectonic" when every syntactic string has a unique construction by means of the system rules (Curry 1963: 41). The syntactic palindrome system as given is not monotectonic. The syntactic palindrome system, A_2, however, can be made into a monotectonic system by restricting in the second rule to a single letter. A monotectonic system is then immediately convertable into an abstract system with an isomorphism to a monotectonic system. So, Curry (1958: 261–266) proves that every syntactic system meeting certain conditions is monotectonic. One condition is that it must be decidable which rule applies to construct each string. Another condition is a generalization of the requirement that a well-formed formula has closed left and right parentheses. But in general, the associativity of concatenation has to be formalized in order to obtain an abstract formal system to which a given syntactic system is isomorphic (Curry 1958: 268–269; 1963: 60).

4. STRUCTURALISM

Curry (1963: 57) distinguishes an abstract ob-system from its "representations." A representation is anything isomorphic with a formal system which is an abstract ob-system. Even to begin reasoning about an abstract ob-system, one must use symbols, constituting a representation.

Curry's formalization procedure results in a representation with an isomorphism between it and the abstract system of which it is a representation. William W. Tait calls this sort of procedure "Dedekind abstraction" (1986: 369, n. 12; 2005: 87, n. 17). Dedekind does not identify the continuum with the class of what are today called Dedekind cuts. Rather he introduces an abstract system of "continuous space" (Dedekind 1963: 38), together with an isomorphism from the abstract system to its instantiation in the system of cuts. Likewise, Dedekind does not identify natural numbers with any infinite system except the abstract one. Dedekind (1963: 68, Definition 73) defines natural numbers as the infinite abstract system ordered by a one-to-one function onto a proper part of itself, and for which "we entirely neglect the special character of the elements." Dedekind's Theorem 126 (1963: 85–86) says that any simply infinite system has a unique isomorphism defining it recursively from the natural number system. So, it is sufficient to consider the natural numbers as an abstract system.

Curry uses Dedekind abstraction, making him into what Geoffrey Hellman and Stewart Shapiro (2019: 51–61) call a "sui generis structuralist," who does not require a set theoretical, category theoretical, or modal interpretation in order to confirm the reality of a structure.

For Curry (1963: 61) the main reason for talking about abstract ob-systems is that these are mathematical invariants: "Consequently it agrees with the tendency in mathematics to seek intrinsic, invariant formulations, such as vectors, projective geometries, topological spaces, etc." So, the basic consideration on behalf of Curry's abstract approach to formal systems is that it fits a trend of mathematical practice since Dedekind's time.[3] We can now see how Curry's structuralism deals with the identity problem posed by Keränen (2001), and briefly by Burgess (1999). The identity problem is that a non-trivial automorphism of a system can seem to induce identities between non-identical elements of that system.

Let's define a second system of palindromes, $\mathbf{P_2}$,with the same atoms as above, but with a different unary operation:

1. Each a, b_i is an atom.
2. Given an atom, construct that same atom as a string.
3. Given string X, construct the string, $X \vee b_i$.

Let $\mathbf{P_2}$ have axioms and rules that are the same as $\mathbf{P_1}$, except with $\vee$ instead of $\wedge$. We can visualize the $\wedge$ as generating a string to the right, while the $\vee$ operation

[3]Curry (1963: 15–16) distinguishes various levels of abstraction without taking any position on which are real. First is the abstraction of an expression or word from many utterances, thoughts, or inscriptions of that expression. Second is the abstraction from space and time limits on inductively defined processes, allowing for any finite number of steps. Mathematical abstraction of an inductive process of any finite length might be seen as an extension of the idealizations used in physical science. Third are completed infinities considered by set theory, which abstract from inductive definitions, and which Curry does not discuss in his 1963 book. Both abstract ob-systems and syntactical systems of expressions involve the first two levels of abstraction.

generates a symmetrical string to the left. So there is the obvious isomorphism, $\mathbf{P_1} \cong \mathbf{P_2}$, defined recursively by mapping $X \wedge b_i$ to $X' \vee b_i$, where X is mapped to X'. With respect to the internal structures of abstract ob-systems, $\mathbf{P_1}$ and $\mathbf{P_2}$, the respective operations, $\wedge b_i$ and $\vee b_i$, are not really different.

Then there is a third system, $\mathbf{P_3}$, with both operations. Even though it does not matter which operation is which, the operations, $\wedge b_i$ and $\vee b_i$, need to be distinguished in $\mathbf{P_3}$ and one must use distinct symbols for them. This is also true in cases Keränen (2001) and Burgess (1999) discuss, such as 1 and -1 in the additive group of integers. There is a non-trivial automorphism on $\mathbf{P_3}$, mapping $X \wedge b_i \vee b_i$ to $X' \vee b_i \wedge b_i$, and vice versa.[4] By this automorphism, we can infer that $\vdash X \equiv Y$ if and only if $\vdash X' \equiv Y'$, where X' is the isomorphic image of X. Nevertheless, the automorphism does not induce the undesired identity statement, $\vdash X \equiv X'$, in the combined system, $\mathbf{P_3}$, because $\mathbf{P_3}$ formalizes reasoning about identity.[5]

5. UNLABELED GRAPHS

Unlabeled graphs present an excellent example for investigating Keränen's (2001) identity problem for structuralism. Leitgeb (2020a, b) formalizes reasoning about unlabeled graphs. He concludes by dissolving the identity problem.

Curry's method allows us to do the same in a different way. Unlabeled graph terms are constructed from an empty graph, 0, by two operations: adjoining a vertex, a_i and for any two vertices, a_i and a_j, adjoining an edge, $b_{a_i a_j}$, connecting those two vertices:

1. Given X, construct $X \wedge a_i$
2. Given $X \wedge a_i \wedge a_j$, construct $X \wedge a_i \wedge a_j \wedge b_{a_i a_j}$

[4]Perhaps some will say these finally are the real palindromes! I would say rather that the question of which are the real ones has been answered by an analysis of three distinct formal systems.

[5]Curry never says that ordinary mathematical discourse in the U-language lacks meaning, although it is sometimes insufficiently precise. Keränen (2001: 315) argues that a structuralist metaphysics excludes "non-relational" properties and constants from the language used to talk about structures. So presumably, for his argument, "is an inverse of the generator of an infinite group generated by one element" would not be a relational property, or perhaps the word, "generator," is a surreptitious constant. But reference to such objects as generators or their inverses seems to be necessary for reasoning about groups.

The unlabeled character of graphs freely generated by these two operations is described by axioms to be given presently. Following Leitgeb (2020a), these unlabeled graphs have undirected edges, at most one edge for any two distinct vertices, and no loop edges from a vertex to itself. The rules of inference are the same as for the **P** systems above.

Unlabeled graphs are represented by unlabeled graph terms. The idea is to distinguish obs from their automorphic images. Graph terms allow us to make the needed distinction linguistically by distinct terms. Then we can check whether unwanted identities of subterms are induced by maps from one term to another.

Much like palindromes, the question of whether the subject matter is really unlabeled graphs will be answered as fully as mathematics requires, because we will be making all the required distinctions to capture our reasoning. In fact, we distinguish more graphs than we would visually, since distinct terms can be constructed with more than one copy of the same vertex or of the same edge. But in the formal system, such terms are equated by cancellation axioms.

The identity problem is about unwanted identities. Consider two unlabeled graphs, G_1 and G_2. The unlabeled graph, G_1, consists of a single vertex, a_1. G_2 consists of two disconnected vertices, a_2 and a_3, and no edge. In G_2, $a_2 \neq a_3$. But it might seem reasonable that $a_1 = a_2$. After all, as unlabeled graphs, each vertex has no other properties. But if so, then the same thought leads to $a_1 = a_3$, and then transitivity of equality brings a collapse of the two distinct vertices in G_2. So, no reasonable formal system for unlabeled graphs can include such identities. To spell out the details, notation can be simplified. Every graph construction begins with the empty graph, 0. So we might as well not mention it. Also, both adjoining operations are $\wedge$-operations so we might as well write these by concatentation, as long as we do not neglect to include explicit axioms and rules for associativity.

So the construction operations for obs can be restated. We assume a countable supply of obs: $a_0, a_1, a_2, \ldots$ Assume that $i \neq j \neq k \neq l$. Operations:

1. Given X, construct Xa_i
2. Given Xa_ia_j, construct $Xa_ia_jb_{a_ia_j}$

Axioms 1–4 are commutativity axioms. Commutativity axioms capture the unlabeled character of these constructions: the order in which vertices or edges are constructed does not matter. Axioms 5 and 6 are cancellation axioms. Axiom 5 captures both the uniqueness of each vertex despite it not having a name. Axiom 6 applies to edges analogously, since we think of these graphs as having at most one edge between any two vertices. Axioms 7–9 for associativity and reflexivity of equivalence need no further explanation. Assume the following axiom schemes:

1. $a_i a_j \equiv a_j a_i$ for any graph with two applications of the vertex-adjoining operation.
2. $b_{a_i a_j} \equiv b_{a_j a_i}$ for any graph with an application of the edge-adjoining operation.
3. $b_{a_i a_j} b_{a_k a_l} \equiv b_{a_k a_l} b_{a_i a_j}$ for any graph with two applications of the edge-adjoining operation.
4. $b_{a_i a_j} a_k \equiv a_k b_{a_i a_j}$ for any graph with any combination of vertices and edges.
5. $a_i a_i \equiv a_i$, cancels repeated adjunctions of the same unlabeled vertex.
6. $b_{a_i a_j} b_{a_i a_j} \equiv b_{a_i a_j}$, cancels repeated adjunctions of the same unlabeled edge.
7. $X(Y a_i) \equiv (XY) a_i$, makes associativity explicit.
8. $X(Y b_{a_i a_j}) \equiv (XY) b_{a_i a_j}$, makes associativity explicit.
9. $X \equiv X$.

Rules of inference are the same as for $\mathbf{P_1}$ above:

1. If $X \equiv Y$, then $Y \equiv X$.
2. If $X \equiv Y$ and $Y \equiv Z$, then $X \equiv Z$.
3. If $X \equiv Y$, then $XZ \equiv YZ$.
4. If $X \equiv Y$, then $ZX \equiv ZY$.

We have not defined automorphisms of unlabeled graphs formally. For our purposes here, a fully formal system of morphisms of unlabeled graphs is unnecessary. Instead, informally, an automorphism is a one-to-one map f from a graph to itself, sending:

- $f : a_i \mapsto a_j$, and

- $f : b_{a_i a_j} \mapsto b_{a_k a_l}$,

such that
$$f(b_{a_i a_j} b_{a_j a_k}) \equiv f(b_{a_i a_j}) f(b_{a_j a_k}).$$

That is, an isomorphism on terms is a bijection that preserves connectivity. Axiom 3 is the only relevant axiom or rule to check whether any proposed f preserves connectivity, in a setting with the other commutativity axioms. Anything that might intuitively be an automorphism of unlabeled graphs has to be sufficiently explicit to allow this to be checked.

There might yet be a worry as to whether commutativity axioms, plus the other axioms and rules capture all the unlabeled graph automorphisms. For instance, what about permutations of the indices, i, j, etc.? Such permutations would make preserving connectivity nontrivial. But I think that the permutation of indices is contrary to the unlabeled character of these graphs. Consider the graph, $a_i a_j$, of two unconnected vertices. Axiom 1 says that this is automorphic with the graph $a_j a_i$. Permuting the indices would be redundant, as well as contrary to the unlabeled character of the graphs.

This level of detail is worked out, however, merely to show that no deduction in this formal system has a conclusion of form, $a_i \equiv a_j$ or $a_i \equiv f(a_j)$. This is easy to see by inspection of the axioms and rules, since no axiom states the equivalence of two distinct vertices, and no rule will permit the inference that two distinct vertices are equivalent. So, this confirms Leitgeb (2020a, b) by using Curry's methods of formalization.

In order to differentiate between automorphisms that induce unwanted identities among subterms (or elements) and those that do not, we have treated automorphic images of terms as distinct terms. Then we can ask whether unwanted identities occur. On the other hand, it is often mathematically useful to treat isomorphic terms as identical (as "unique up to isomorphism"), and thus as representing the same graph. The latter approach is not useful here. But we may want to identify isomorphic graphs if instead of studying unlabeled graphs as distinct objects, we were studying the order structure on subgraphs.[6]

This approach can easily be generalized, for instance, to Euclidean geometry. Euclid postulates that for any two points, there is a line through those two points.

[6]Leitgeb handles this by assuming both Axiom G2 (Leitgeb 2020a: 334) that no two graphs share the same vertices, together with Axiom E2 (Leitgeb 2020a: 335) that for any graph there is a second graph with a subgraph isomorphic with the first graph plus one additional vertex in the second graph.

For those two points in reverse order, then, Euclid's postulate constructs the same line. Reversing the order of the points induces an automorphism. But there is nothing in this that suggests a valid inference that those two points are identical.

6. METHODOLOGICAL AUTONOMY

Formalization is a standard mathematical method for replacing imprecise language with precise language as this becomes necessary. Here it is used for proving and disproving identity statements. So, we need a formal system in which there are identity statements. The definition of an abstract ob formal system in Curry's sense includes operations whose inductive closure can reasonably be seen as the abstract subject matter of that formal system. So, Curry's approach is a mathematically autonomous way of answering the question about the nature of the ontology of the subject matter of pure mathematics.

Penelope Maddy's (2007: 367–377) discussion of "thin realism" explains more about what it means for a mathematical practice to be autonomous. Thin realism says that set theory speaks for itself, determines goals of inquiry for itself, and can be taken to define what it is to be a set. Then sets are exactly the kind of thing that set theoretical practice, when done well, says they are. There is no definite boundary to this claim of authority. We want to bring specialized precision to bear on philosophical problems, and to answer our questions clearly without imposing putative answers that do not really answer the questions we are asking.

Maddy distinguishes the ontological claim of thinness from the semantic idea of disquotationalism, which might seem similar. Moreover, thin ontology discourages one from jumping to conclusions, such that the continuum hypothesis has indeterminate truth value, just because it has not been decided.[7] If so, there can be tolerance of other methodologically autonomous mathematical practices, as long as these likewise pose no danger to any mathematics.

Curry's method is not autonomous in an absolute sense. Curry, Hindley, and Seldin (1972: 7–8) cite a Dutch graduate student, Fleischhacker, for posing the problem of decidability for being a word in a symbol system, when the class of symbols itself might not be a recursive set in relation to the universe. Any human must use symbols to reason about an abstract ob-system. The philosophical

[7]Since an autonomous practice does not have to justify its authority, Maddy (2007: 377–382) also countenances an "arealist" stance of disengaging from ontological questions altogether as unnecessary for the practice of pure or applied mathematics.

questions remain about the relationship between abstract structures and the natural world we live in. These questions do not vanish even if the ontology of abstract structures is clarified.

I do not, however, think that it matters that Curry's view does not explain the ontology of informal mathematics, which Seldin (2011: 94) considers to be one of the main criticisms of Curry's view. Rather, it seems to me that informal mathematics is a vast collection of unsolved problems of formalization. If many of these problems are trivial, so much the better for the sake of autonomy.

What kinds of mathematical methods and results can answer ontological questions autonomously? To the extent these methods are mathematically autonomous they might evade philosophical criticism. Since these are autonomous methods, we cannot expect mathematical reasons to decide for or against any such method. We can expect many such methods. Reverse mathematics (Simpson 2009) directly answers some ontological questions by giving proofs. So reverse mathematics is an autonomous method. Curtis Franks (2009) argues that the primary aim of Hilbert's program is mathematical methodological autonomy from philosophical critique.

REFERENCES

Barendregt H. P. (1984), *The Lambda Calculus*, revised ed., Amsterdam: North-Holland.

Burgess J. P. (1999), Book Review: "Stewart Shapiro, Philosophy of Mathematics: Structure and Ontology," *Notre Dame Journal of Formal Logic* 40(2), 283–291. https://doi.org/10.1305/ndjfl/1038949543

Carnap R. (1959), *The Logical Syntax of Language*, transl. by A. Smeaton, Paterson, NJ: Littlefield, Adams. [1934 German publication].

Church A. (1941), *The Calculi of Lambda-Conversion*, Princeton: Princeton University Press. https://doi.org/10.1305/ndjfl/1038949543

Curry H. B. (1950), "Language, Metalanguage, and Formal System," *The Philosophical Review* 59(3), 346–353. https://doi.org/10.2307/2181990

Curry H. B. (1951), *Outlines of a Formalist Philosophy of Mathematics*, Amsterdam: North-Holland.

Curry H. B. (1958), "Calculuses and Formal Systems," *Dialectica* 12(3–4): 249–273. https://doi.org/10.1111/j.1746-8361.1958.tb01462.x

Curry H. B. (1963), *Foundations of Mathematical Logic*, New York: Dover Publications.

Curry H. B. and Feys R. (1958), *Combinatory Logic*, vol. I, Amsterdam: North-Holland.

Curry H. B., Hindley J. R., and Seldin J. P. (1972), *Combinatory Logic*, vol. II, Amsterdam: North-Holland.

Dedekind R. (1963), *Essays on the Theory of Numbers*, New York: Dover Publications. [1872 and 1887 German publications].

Franks C. (2009), *The Autonomy of Mathematical Knowledge: Hilbert's Program Revisited*, Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511642098

Hellman G. and Shapiro S. (2019), *Mathematical Structuralism*, Cambridge: Cambridge University Press. https://doi.org/10.1017/9781108582933

Hindley J. R. and Seldin J. P. (2008), *Lambda-Calculus and Combinators: An Introduction*, Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511809835

Keränen J. (2001), "The Identity Problem for Realist Structuralism," *Philosophia Mathematica* 9(3): 308–330. https://doi.org/10.1093/philmat/9.3.308

Leitgeb H. (2020a), "On Non-eliminative Structuralism: Unlabeled Graphs as a Case Study, Part A," *Philosophia Mathematica* 28(3): 317–346. https://doi.org/10.1093/philmat/nkaa001

Leitgeb H. (2020b), "On Non-eliminative Structuralism: Unlabeled Graphs as a Case Study, Part B," *Philosophia Mathematica* 29(1): 64–86. https://doi.org/10.1093/philmat/nkaa009

Maddy P. (1997), *Naturalism in Mathematics*, Oxford: Oxford University Press.

Maddy P. (2007), *Second Philosophy: A Naturalistic Method*, Oxford: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199273669.001.0001

Parsons Ch. (2008), *Mathematical Thought and Its Objects*, Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511498534

Resnik M. (1997), *Mathematics as a Science of Patterns*, Oxford: Oxford University Press.

Seldin J. P. (2011), "Curry's Formalism as Structuralism," *Logica Universalis* 5, 91–100. https://doi.org/10.1007/s11787-011-0028-3

Simpson S. G. (2009), *Subsystems of Second Order Arithmetic*, 2nd ed., Cambridge: Cambridge University Press.

Tait W. W. (1986), "Proof and Truth: The Platonism of Mathematics," *Synthese* 69, 341–370. https://doi.org/10.1007/BF00413978

Tait W. W. (2005), *The Provenance of Pure Reason: Essays in the Philosophy of Mathematics and its History*, Oxford: Oxford University Press.

Troelstra A. S. and Schwichtenberg H. (2000), *Basic Proof Theory*, 2^{nd} ed., Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139168717