On Expanding Standard Notions of Constructivity

Liron Cohen
Dept. of Computer Science
Cornell University
Ithaca, NY, USA

Ariel Kellison
Dept. of Computer Science
Cornell University
Ithaca, NY, USA

Abstract

Brouwer developed the notion of *mental constructions* based on his view of mathematical truth as experienced truth. These constructions extend the traditional practice of constructive mathematics, and we believe they have the potential to provide a broader and deeper foundation for various constructive theories. We here illustrate mental constructions in two well studied theories – computability theory and plane geometry – and discuss the resulting extended mathematical structures. Further, we demonstrate how these notions can be embedded in an implemented formal framework, namely the constructive type theory of the Nuprl proof assistant. Additionally, we point out several similarities in both the theory and implementation of the extended structures.

1 Introduction

Plane geometry and computability theory share a constructive foundation. Since its inception in the Elements, Euclidean *plane geometry* has been conceived of as a theory based on the ruler and compass constructions. Similarly, *computability theory* is founded on functions for which there is an effective method, or computation, for obtaining the values of the function. Just as standard geometric constructions are informally perceived as those of the straightedge and compass, the standard computations are informally perceived as those computable by some pen and paper method.

Intuitionistic mathematics as conceived by Brouwer (see e.g. [12, 25, 32]) extends the standard notions of constructions by admitting also those constructions corresponding to human experiences of mathematical truths. Brouwer adds to the effective, algorithmic constructions mental constructions made by the *idealized mathematician* (or the *"creative subject"*).

According to Brouwer, mathematical truths are experienced, and thus mental constructions are formed, based on temporal, rather than spatial, intuitions:

> From this intuition of time, independent of experience, all the mathematical systems, including spaces with their geometries, have been built up, and subsequently some of these mathematical systems are chosen to catalogue the various phenomena of experience. [11]

PL'18, January 01–03, 2018, New York, NY, USA
2018.

In computability theory, the notion of infinitely proceeding sequences of freely chosen objects, known as *free choice sequences* (which lay at the heart of intuitionistic mathematics), can only take place when regarding functions as constructed over time. In plane geometry, the construction of *points at infinity*, which are the intersection of infinitely proceeding parallel lines (which are the essence of projective geometry), has no basis in our intuition of space, which is fundamentally Euclidean and thus limits our capacity for experiencing geometric truths.

We here incorporate these two types of objects into their corresponding theories. Even though exploiting the intuitionistic notion of constructions in these theories exceeds their respective standard constructions, we demonstrate how they can be captured in a formal computational framework. Namely, we outline how these ideas can be expressed in the constructive type theory of the Nuprl proof assistant [1, 18].

2 Expanding the Function Space

2.1 Standard Computability Theory

Church-Turing computability constitutes as the standard notion of computation. This notion defines the computable functions as those for which there is an *effective method* for obtaining the values of the function. For those, Turing used the term 'purely mechanical', whereas Church used 'effectively calculable'.

> define the notion ... of an effectively calculable function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ-definable function of positive integers). [15]

This notion of computability is the one underlying the computational theories invoked by standard constructive type theories, which in turn are at the base of extent proof assistants such as Nuprl, Agda [8], and Coq [4]. Thus, the elements of the function type $A \rightarrow B$ are taken to be the *effective (computable)* functions from the type A to the type B. The canonical form of elements of this type are therefore terms of the form $\lambda x.t$.

2.2 The Creative Subject and Choice Sequences

In his exploration of intuitionistic mathematic Brouwer put forward a new notion of computation that exceeds the standard Church-Turing computability. He proposed accepting *non-lawlike* computations, i.e. computations for which there is no method governing them. This notion was captured using the concept of *free choice sequences* [9, 19, 30].

Choice sequences are never finished sequences of objects created *over time* by a *creative subject.* They can be lawlike in the sense that they are determined by an algorithm (i.e., standard computable functions), or lawless in the sense that they are not subject to any law (i.e., free). Free choice sequences are described as:

> new mathematical entities... in the form of infinitely proceeding sequences, whose terms are chosen more or less freely from mathematical entities previously acquired... [9]

Hence, a free choice sequence is infinitely proceeding, i.e. it comes into existence by a never ending process of picking natural numbers. Therefore, it is never fully completed and can always be extended. The choices of values in a free choice sequence are made freely, that is, they are not governed by any rule. For instance, one might think of the results of tossing a dice time and time again as a free choice sequence. While this clearly steps out of the realm of sequences constructed by an algorithm, there is a mental conception of how to create such sequences. The ideal mathematician, or the creative subject, can simply pick numbers as time proceeds.

Choice sequences were originally introduced by Brouwer in order to explain the structure of the continuum. In contrast to Bishop's account of the constructive reals, Brouwer's intuition was that the continuum can not be seen as constructed by discrete elements, rather the continuum should have the property that it cannot be "pulled apart".[1] Brouwer developed the intuitionistic continuum by defining a real number as a choice sequence of nested rational intervals. A key point is that the choice sequence itself is the real number, and not its limit. Brouwer's interpretation of the continuum is then given by the concept of a spread, which can be thought of as a totality of choice sequences. In the spread one is not able to refer to any specific path (i.e., an individual choice sequence) only to a subspread. This is reflected in the central axiom for free choice sequences (known as 'Axiom of Open Data' [31]) which roughly speaking states that if a property holds for a free choice sequence, then there is a finite initial segment of that sequence, s, such that this property holds for all free choice sequences with s as initial segment. Thus, the axiom does not provide information on a specific choice sequence, rather on the subspread determined by an initial segment, therefore in a sense constitutes as a continuity principle.

The existence of free choice sequences has major implications which lay at the heart of intuitionistic mathematics. For example, the Axiom of Open Data (more precisely, the continuity principle for numbers which follows from it) was used by Brouwer in order to prove that all real-valued functions on the unit interval are uniformly continuous [10, Thm.3]. The bar induction principle, which is a powerful intuitionistic induction principle (equivalent to the classical transfinite induction) is another consequence of the introduction of free choice sequences [13, 14], which has also been explored in the context of the Nuprl proof assistant [29].

[1] As put nicely in [7]: "In Brouwer's case there seems to have been a nagging suspicion that unless he personally intervened to prevent it, the continuum would turn out to be discrete.".

3 Expanding the Point Space

3.1 Standard Constructions for Euclidean Geometry

What does it mean for a geometric proposition to be true? Certainly, a geometric proposition is true if it is validated by a ruler and compass construction. Conviction in geometric truths of this nature requires, for a formal analysis, axioms expressed using the formalization of these traditional tools. Axiomatizations governed by ruler and compass constructions have led to the conception of an abstract *human geometer*, who applies, in an algorithmic way, the traditional geometric tools to "geometric things." This is notably similar to the informal correspondence between computable functions and algorithms that can be carried out by some idealized *human computer* by pen on paper [2]

The analogy between the human computer and the human geometer becomes less apparent when it is necessary to identify what exactly is meant by "geometric things." While the human computer uses only purely mathematical structures as inputs when evaluating functions, the abstract geometer may not be required to carry out constructions on objects with definite mathematical properties. For example, in Euclid's *Elements*, points are defined only as "that which has no part" [20]. Therefore, Euclidean ruler and compass axiomatizations are historically *synthetic* – they do not stipulate what the primitive objects are – and therefore admit both "pre-mathematical" and purely mathematical models [22].

Numerous works have developed axiomatic systems for geometry that rely on straightedge and compass (or similarly familiar) constructions, e.g. [2, 3, 24, 28]. Euclidean geometry based on straightedge and compass constructions has been implemented in the Nuprl proof assistant as well [5].

3.2 Extending the Euclidean Point Space: The Projective Extension

A commonplace extension of the Euclidean plane is the *projective extension.* A projective plane is constructed from the Euclidean plane by adding to the Euclidean points *points at infinity* corresponding to the perception that parallel lines intersect at a point on the horizon.[3] This extension is trivial classically, but requires great effort in an intuitionistic

[2] We borrow this analogy from, e.g. [26].

[3] Actually, one also has to add to the Euclidean lines a line at infinity to which all the points at infinity are incident. The formalization of this line is straightforward once the extension of points is established, therefore we elide this treatment in what follows.

setting. [4] The projective extension exceeds the notion of standard constructions in Euclidean geometry. Nonetheless, the projective extension can be captured within a formal computational framework as we outline in the next section.

Remark. Note that, even in the Euclidean plane, an intuitionistic treatment admits constructions that exceed the ruler and compass [2]. In order to ensure that the intuitionistic continuum serves as a model for geometry, it is necessary to make use of the *apartness relation* on points (or some equivalent notion), see Sec. 4.2.1. Apartness, together with its corresponding axiom of co-transitivity on points, provides a "global notion" that imposes a continuous behavior on the topological space of synthetic points [33]. A result of this extension of the traditional ruler and compass constructions in the Euclidean plane is a surprisingly concise capturing of Euclid's propositions. Specifically, in our Nuprl implementation of Euclidean geometry with the apartness relation, we were able to prove a constructive version of Proposition 2 from Book I of the Elements in its full form. Proposition 2 constructs the rigid compass from the collapsing compass (the collapsing compass is the "axiomatic" compass taken by Euclid). In [2], Michael Beeson shows that the full form of Proposition 2 is not provable in a system of Euclidean geometry using constructive logic without apartness.

4 Type Theoretic Account

In this section we demonstrate how the aforementioned extended notions of constructions can be implemented in a formal system, namely the Nuprl proof assistant. We start by outlining the key components of the implementation of each theory, and then discuss the similarities between them.

4.1 Implementation in Nuprl

The Nuprl proof assistant implements a type theory called *Constructive Type Theory* (CTT), which is a dependent type theory, in the spirit of Martin-Löf's extensional theory [23], based on an untyped functional programming language. It has a rich type theory including equality types, W types, quotient types, set types, union and (dependent) intersection types, PER types, approximation and computational equivalence types, and partial types.

The quotient type [17, 27] is of particular use to us in this work. Given a type T and an equivalence relation E on T we can form in Nuprl the quotient type $T//E$ whose elements are the elements of T, but the underlying equality of the type is redefined by E. That is, two elements $x, y \in T$ are equal in the quotient type $T//E$ provided $E(x, y)$.

[4] Heyting was the first to publish on the projective extension, and did so under the premise of exemplifying the complexities of intuitionistic axiomatizations [21].

4.1.1 Choice Sequences

Recently we have integrated choice sequences into the constructive type theory implemented by Nuprl proof assistant [16], thus showing that CTT is expressive enough to extend computation to Brouwer's broader notion that includes non-lawlike computability. The free choice sequences were there introduced into the function type $A \rightarrow B$, which previously exhibited only law-like sequences.

Choice sequences were realized essentially as global top-level definitions, whose contents are lists that grow over time. This was implemented using the library underlying Nuprl as a state in which choice sequences are stored, so that the choices of values that have been made to a particular choice sequence at a given point in time can be recorded. A choice sequence entry in the library is simply a list of terms that can be expanded by adding more values. This dynamic nature of libraries is accounted for in the extended framework using a Beth-like semantics.

4.1.2 The Projective Extension

Recall that a synthetic axiomatization of geometry does not stipulate what the primitive objects are. The constructive geometry implemented in the Nuprl proof assistant is synthetic, and the Euclidean point type, P_{eu}, therefore retains an abstract character.

The new type of points in the projective extension, i.e. points at infinity, are formalized using Nuprl's quotient type. Let L_{eu} be the type of Euclidean lines, which are constructed from pairs of distinct elements of P_{eu}. The standard parallelism relation, Par, forms an equivalence relation on this type. The type of points at infinity, P_∞, is then formed by the quotient type:

$$P_\infty := L_{eu}//Par.$$

The extension of the point space is achieved by forming the disjoint union of the Euclidean points and points at infinity:

$$P := P_{eu} \sqcup P_\infty.$$

The computational interpretation of the disjoint union type allows us to discriminate between its internal elements, i.e. argue by cases (vide infra).

4.2 Common Type Theoretic Features

4.2.1 The Notion of Equality

In Nuprl's type theory, each type comes with its own equality relation (extensional equality in the case of functions), and the typing rules guarantee that well-typed terms respect these equalities. However, since free choice sequences are non-lawlike infinitely proceeding entities and Euclidean points are the atomic, indecomposable elements of the plane, this built-in syntactic equality does not suffice for capturing the intuitive notion of identity on those types (much like in the case for the reals). Having to construct the notion of equality for the types is the trade off for the introduction of this new broader notion of constructions.

To constructively build the equality on these types we use another primitive notion associated with the types which is strongly connected to equality: the notion of distinctness. Stating that two choice sequences α, β are distinct can be done in the following way:

$$\alpha \# \beta := \exists n \in \mathbb{N}. \alpha(n) \neq \beta(n)$$

This definition also exhibits constructive content due to the type theoretical interpretation of the existential quantifier. That is, to establish that two choice sequences are distinct an evidence for a position in which their values differ must be constructed.

In geometry, the "distinct" terminology is replaced by "apart". Recall that the projective points were formed by the disjoint union between two types: Euclidean points (P_{eu}) and points at infinity (P_∞). The tagging on these types allows us to define the apartness relation on projective points by cases. Firstly, all points at infinity are taken to be apart from all Euclidean points. When both projective points are Euclidean points, the projective apartness relation is the Euclidean apartness relation on points ($\#_{eu}$); and when both projective points are points at infinity, the apartness relation coincides with their corresponding Euclidean lines being non-parallel.

Note that just as the apartness for choice sequences referred to the underlying structure of the natural numbers from which they are constructed (or any other underlying structure of the sequences value type), the apartness of the points at infinity refers to the structure of the Euclidean lines from which they are constructed (which, in turn, is based on Euclidean points).

Instead of a primitive equality, the negation of the distinctness (apartness) relation on choice sequences (projective points), i.e.

$$a \equiv b := \neg a \# b,$$

forms an equivalence relation, which is then respected by the other primitive concepts. This is obviously not the primitive, built-in equality that generally comes with the definition of a type in type theory, but it allows for a practical, meaningful way of reasoning about the relations between the elements of the types.

In the case of choice sequences, note that constructively the negation of the statement that two choice sequences are distinct does not entail a notion of extensional equality on choice sequences. Because free choice sequences come into existence by an infinite, never terminating construction, there is no way in which one could ever determine that for *every* natural number n, the n'th elements in two given free choice sequences are equal (given that only the extensional data, i.e. the values, of the sequences are available to us). This is another justification for why choice sequences cannot be thought of individually, but only as elements of the totality (or the spread in the Brouwerian account of the continuum).

Remark on the Euclidean Apartness Relation ($\#_{eu}$). Euclidean points are primitive, and thus have no underlying notion to refer to. The Euclidean apartness relation on points is therefore also primitive, in contrast to the projective apartness relation. The relation $a \#_{eu} b$ is realized in the model of the reals by the existence of a natural number n such that a and b are separated by more than $\frac{1}{2}^n$. A subtle point of the Euclidean apartness relation is that using the quotient type, i.e using $P_{eu}//\equiv$, in order to make equivalence coincide with equality has undesired consequences. If equality and equivalence were to coincide, the co-transitivity of apartness:

$$\forall a, b \in P_{eu}. \; ((a \#_{eu} b) \rightarrow \forall c \in P_{eu}.(\; c \#_{eu} a \; \vee \; c \#_{eu} b))$$

would be a function that *respects the equivalence relation* and decides, for any two separated points a and b and any other point c, whether c it is apart from a or b. Any function respecting equivalence, as a corollary of Brouwer's uniform continuity theorem (which is provable in Nuprl), is constant. As a result, we would not be able to prove that the plane constructed from the real numbers satisfies the co-transitivity axiom, which is a salient property of the apartness relation.[5]

4.2.2 The Underlying Computation

A critical component in a constructive type theory is the underlying computation system, which is essentially the untyped programming language underlying the type theory. The data and the programs of the computation system are given by (closed) terms, which can be either canonical or non-canonical. Terms having a canonical operator are called values. Computation is defined as a sequence of rewritings or reductions of terms to other terms according to very explicit rules. Canonical terms, such as $\lambda x.x$ reduce to themselves. A non-canonical term such as the one corresponding to function application $(\lambda x.x)y$ reduces in one step to y using the rule $(\lambda x.t)a \mapsto t[x := a]$.

In Nuprl, the only thing one can do with a function is to apply it. This has the consequence that the function type in Nuprl is essentially defined in terms of its deconstructor, the application of a function. To support the existence of free choice sequences a new case to the application rule for a free choice sequence was added. Accordingly, the inference rule for function application has been modified so that $f(a)$ might be computed to a value also in case f is a choice sequence, not only if it computes to a λ-term. In the former case the computation was done by looking up the value in the free choice sequence entry in the library. This case-based formulation of the computational rule states that, even though in [16] the extension of the function space was done somewhat differently, it may be perceived as the disjoint

[5]This illustrates Bishop's claim that forming such a quotient is "either pointless or incorrect"[6, p.65]. For models with decidable equality, forming the quotient $P_{eu}//\equiv$ would be "pointless". For a model of the reals, forming the quotient is incorrect.

union of the type of law-like choice sequences and the type of free choice sequences.

While functions are governed by their application, points in geometry are governed by their associated constructions. The constructive reading of the Straightedge-Straightedge construction postulate supplies the Skolem term $SS(a, b, x, y)$ to represent the construction of the point z. In the case of points at infinity, there are no corresponding λ-terms in constructions; these points are the quotiented elements of the type of projective points and therefore axioms asserting the existence of such points carry no computational content. In this way, constructions on projective points in the case of points at infinity are similar to the computation rule in the case of a free choice sequence, which has to refer to the corresponding entry in the library as it has no underlying λ-term to refer to.

5 Conclusions

In this paper we outlined how Brouwer's generalized notion of constructions can be incorporated into the constructive theories of computation and geometry within a formal framework. We show that the extensions of both theories rely on common intuitive notion of mental construction, and compare the main features in their implementations into the Nuprl proof assistant. We aimed to demonstrate that the philosophical idea of mental constructions put forward by Brouwer can be captured within an implemented formal system, and therefore has great potential if further explored in the context of computerized systems.

References

[1] S.F. Allen, M. Bickford, R.L. Constable, R. Eaton, C. Kreitz, L. Lorigo, and E. Moran. 2006. Innovations in computational type theory using Nuprl. *Journal of Applied Logic* 4 (2006).

[2] Michael Beeson. 2009. Constructive Geometry. In *Proceedings of the 10th Asian Logic Conference*. 19–84.

[3] Michael Beeson. 2015. A constructive version of Tarski's geometry. *Annals of Pure and Applied Logic* 11 (2015).

[4] Yves Bertot and Pierre Casteran. 2004. *Interactive Theorem Proving and Program Development.*

[5] Mark Bickford, Rich Eaton, and Ariel Kellison. 2018. Nuprl Theory: euclidean plane geometry. (2018). http://www.nuprl.org/LibrarySnapshots/Published/Version2/Mathematics/euclidean!plane!geometry/index.html

[6] Errett Bishop. 1967. *Foundations of Constructive Analysis.*

[7] Errett Bishop and Douglas Bridges. 1985. A Constructivist Manifesto. In *Constructive Analysis*. Springer, 4–13.

[8] Ana Bove, Peter Dybjer, and Ulf Norell. 2009. A Brief Overview of Agda - A Functional Language with Dependent Types. 73–78.

[9] L.E.J. Brouwer. 1919. Begründung der mengenlehre unabhängig vom logischen satz vom ausgeschlossen dritten. zweiter teil: Theorie der punkmengen. *Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam* 12, 7 (1919).

[10] L.E.J. Brouwer. 1927. *From frege to Gödel: A Source Book in Mathematical Logic, 1879–1931.* Harvard University Press, Chapter On the Domains of Definition of Functions.

[11] L.E.J Brouwer. 1975. 1909 The Nature of Geometry. In *Philosophy and Foundations of Mathematics*, Arend Heyting (Ed.). North-Holland, 112–120.

[12] L.E.J. Brouwer. 1975. Consciousness, philosophy, and mathematics. In *Philosophy and Foundations of Mathematics*. Elsevier, 480–494.

[13] L. E. J. Brouwer. 1954. Historical Background, Principles and Methods of Intuitionism. *Journal of Symbolic Logic* 19, 2 (1954), 125–125.

[14] L. E. J. Brouwer. 1981. *Brouwer's Cambridge Lectures on Intuitionism.* Cambridge University Press. 214–215 pages. Edited by D. Van Dalen.

[15] Alonzo Church. 1936. An unsolvable problem of elementary number theory. *American journal of mathematics* 58, 2 (1936), 345–363.

[16] Liron Cohen, Vincent Rahli, Mark Bickford, and Robert Constable. 2018. Computability Beyond Church-Turing via Choice Sequences. In *33nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. To appear.

[17] Robert L. Constable. 1983. Constructive Mathematics as a Programming Logic I: Some Principles of Theory. In *Fundamentals of Computation Theory (LNCS)*, Vol. 158. Springer, 64–77.

[18] Robert L. Constable, Stuart F. Allen, Mark Bromley, Rance Cleaveland, J. F. Cremer, Robert W. Harper, Douglas J. Howe, Todd B. Knoblock, Nax P. Mendler, Prakash Panangaden, James T. Sasaki, and Scott F. Smith. 1986. *Implementing mathematics with the Nuprl proof development system.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[19] Michael A. E. Dummett. 2000. *Elements of Intuitionism* (second ed.). Clarendon Press.

[20] Euclid., Thomas Little Heath, and Dana. Densmore. 2002. *Euclid's Elements : all thirteen books complete in one volume : the Thomas L. Heath translation.* Green Lion Press. 499 pages.

[21] A. Heyting. 1959. Axioms for Intuitionistic Plane Affine Geometry. *Studies in Logic and the Foundations of Mathematics* 27 (1959), 160–173.

[22] Arend Heyting. 1966. Axiomatic Method and Intuitionism. In *Essays on the foundations of mathematics : dedicated to A.A. Fraenkel on his seventieth anniversary* (2d ed. ed.), Yehoshua Bar-Hillel and Abraham Adolf Fraenkel (Eds.). Magnes Press, Hebrew University, Jerusalem, 237–245.

[23] Martin-Löf. 1982. Constructive Mathematics and Computer Programming. In *6th International Congress for Logic, Methodology and Philosophy of Science*. 153–175.

[24] Nancy Moler and Patrick Suppes. 1968. Quantifier-free axioms for constructive plane geometry. *Compositio Mathematica* 20 (1968), 143–152.

[25] Joan Rand Moschovakis. 2009. The Logic of Brouwer and Heyting. In *Logic from Russell to Church*. 77–125.

[26] Alberto Naibo. 2015. Constructibility and Geometry. In *From Logic to Practice: Italian Studies in the Philosophy of Mathematics*, Gabriele Lolli, Marco Panza, and Giorgio Venturi (Eds.). Springer International Publishing, 123–161.

[27] Aleksey Nogin. 2002. Quotient types: A modular approach. In *International Conference on Theorem Proving in Higher Order Logics*. Springer, 263–280.

[28] Victor Pambuccian. 1993. Ternary operations as primitive notions for constructive plane geometry III. *Mathematical Logic Quarterly* 39, 1 (1993), 393–402.

[29] Vincent Rahli, Mark Bickford, and Robert L Constable. 2017. Bar induction: The good, the bad, and the ugly. In *Logic in Computer Science (LICS), 2017 32nd Annual ACM/IEEE Symposium on*. IEEE, 1–12.

[30] A.S. Troelstra. 1977. *Choice Sequences: A Chapter of Intuitionistic Mathematics.* Clarendon Press.

[31] Anne S. Troelstra and Dirk van Dalen. 1988. *Constructivism in Mathematics An Introduction.* Studies in Logic and the Foundations of Mathematics, Vol. 121. Elsevier.

[32] Mark van Atten. 2004. *On Brouwer.* Thompson/Wadsworth, Toronto, Canada.

[33] Dirk van Dalen. 2013. The Breakthrough. In *L.E.J. Brouwer âĂŞ Topologist, Intuitionist, Philosopher.* Springer London, 357–394.