Constructive mathematics, Church's Thesis, and free choice sequences.

D. A. Turner

University of Kent

Abstract. We see the defining properties of constructive mathematics as being the proof interpretation of the logical connectives and the definition of function as rule or method.
We sketch the development of intuitionist type theory as an alternative to set theory. We note that the axiom of choice is constructively valid for types, but not for sets. We see the theory of types, in which proofs are directly algorithmic, as a more natural setting for constructive mathematics than set theories like IZF.
Church's thesis provides an objective definition of effective computability. It cannot be proved mathematically because it is a conjecture about what kinds of mechanisms are physically possible, for which we have scientific evidence but not proof. We consider the idea of free choice sequences and argue that they do not undermine Church's Thesis

Keywords: constructive type theory · Church's Thesis · free choice sequence

Introduction

What makes constructive mathematics *constructive*? I believe it is two things: (i) the *proof interpretation* of the logical connectives, and (ii) restoring the older meaning of function as *rule or method* of which it had been stripped by the development of set theory in late 19C. Both steps are due to Brouwer (1908), whose point of departure was the paradoxes of set theory.

Brouwer's *intuitionism* also drew on his intuitions about free choice sequences for conclusions about properties of the continuum. Modern *constructivism* dates from Bishop (1967) whose treatment of the reals is straightforwardly constructive and doesn't make use of free choice sequences. Bridges & Richman (1987) give a thorough technical comparison of Brouwer's intuitionism, Bishop-style constructivism, and a third strand, *Russian constructivism*, due to Markov, which identifies function with recursive function, thus incorporating Church's Thesis.

Bishop & Bridges (1985) develop constructive analysis within the (informal) framework of set theory using intuitionistic logic and without the axiom of choice. A formal system of constructive set theory (CST) along these lines appears in Myhill (1975). The intuitionist set theory (IZF) of Friedman (1973) is similar, for a full discussion of these theories see Beeson (1985), ch VIII.

An alternative to CST or IZF as a framework for the formal development of constructive mathematics, is *the intuitionist theory of types* of Per Martin-Löf (1973), and its descendants such as Homotopy Type Theory (Univalent Foundations, 2013). These type theories are based on propositions-as-types and differ radically from set theories, whose essential ingredient is some version of the axiom of comprehension.

In the following sections I will cover:

1. Sketch the emergence of propositions-as-types as an alternative to set theory, with a note on the conflicted status of AC (axiom of choice).
2. Church's thesis and constructivity
3. Remarks on free choice sequences.

1 From Frege to Martin-Löf

Propositional Functions

The *Begriffsschrift* of Frege (1879) broke from the analysis, current since Aristotle, of the proposition as comprising subject and predicate. Instead we have an n-ary *propositional function* applied to n terms

$$P(a_1, \ldots, a_n)$$

one of the a_i might be the grammatical subject, others direct and indirect objects, but in Frege's analysis the terms are all treated in the same way. Each term is a referring (or denoting) expression which has a reference (or denotation). In the case of a mathematical proposition this will be a mathematical entity like a number, a function, etc.

But what is the reference of a complete proposition, that is of a propositional function supplied with its arguments? For Frege it was a truthvalue, either True or False. If P is a complete (or saturated) proposition for which we have a proof we can write the *judgement*

$$\vdash P$$

asserting that P has the value True. This is the sole form of judgement in Frege's system. Note that the judgement is not manifest, that is valid on its face. To be justified in writing it we must have a sequence of valid steps in Frege's system whose last step is $\vdash P$.

Frege's analysis of meaning in terms of reference led to various difficulties leading him (Frege, 1892) to introduce a second notion of meaning, *sense*. So for example $\sqrt{16}$ and 4 have the same reference but different senses. But the exact nature of *sense* remained elusive.

Jumping ahead by 90 years, to Howard (1969), we can see in propositions-as-types, an elegant solution to Frege's difficulties. The reference, or denotation, of a proposition is not a truthvalue, but a *type*, namely the type of its proofs. But what is a type?

Types

Russell (1903) in Appendix B "The Doctrine of Types", defines a type as *the range of significance of a propositional function*, that is what we would now call its *domain.* This is different from a set, which is the *extension* of a propositional function, that is the collection of values for which it is true. Types were introduced in an attempt to block self-reference which appear to be at the root of the paradoxes which had been found in set theory, such as the Russell paradox.

Applying the doctrine of types to Frege's analysis of propositions introduces another form of judgement, which we will write

$$a_i :: T_i$$

saying that term a_i has type T_i. Note that this is a manifest judgement, it should be verifiable on its face as a condition of well-formation of the formula in which it stands. To make this judgement we may need context, because in mathematical reasoning we introduce variables, always with their types e.g.

Let n be a natural number ...
let f be a function in $R \to R$ ...
So the general form of a typing judgement is

$$\Gamma \ \vdash a :: T$$

where Γ is a sequence of hypotheses introducing variables with their types. This is a manifest judgement, whose validity can be mechanically checked, as in the type systems of functional programming languages such as Haskell or Agda.

Types versus Sets

Types and sets are quite different in behaviour

– set membership $e \in S$ is not in general decidable but requires proof; that is $e \in S$ is a *proposition* not a judgement.
– basic operations on sets include union $S \cup T$ and intersection $S \cap T$, which usually make no sense on types; the natural operations on types are cartesian product $A \otimes B$, disjoint sum $A \oplus B$, the function type $A \to B$ and the dependent versions of product and function types.
– sets are equal iff they have the same elements — this is *the axiom of extensionality*; the situation with types is more complicated — including that in constructive type theories we must distinguish propositional equality from definitional equality. In Homotopy Type Theory, types are propositionally equal when they are *isomorphic*.
– set theory has the *axiom*[1] *of separation*, or *restricted comprehension*: if T is a set and P a property we can form the set $S = \{x \in T \mid P(x)\}$. This effectively erases the distinction between type and set implicit in Russell's doctrine of types. ZF set theory is typeless, or to put it another way there is only one type — everything is a set.
– the *axiom of choice* of set theory is constructively problematic, as we discuss later, while a choice principle is provable in the main versions of type theory.

[1] technically an axiom schema

Propositions as Types

The standard account of intuitionistic logic is the BHK (for "Brouwer, Heyting, Kolmogorov") or proof interpretation, see for example (Troelstra & van Dalen, 1988). Paraphrasing slightly[2] we have

1. A proof of $A \wedge B$ is given by presenting a proof of A and a proof of B.
2. A proof of $A \vee B$ is given by presenting a proof of A or a proof of B *and saying which has been given*.
3. A proof of $A \supset B$ is a rule or method for constructing from any proof of A, a proof of B.
4. Absurdity $\perp$ (contradiction) has no proof; a proof of $\neg A$ is a rule or method for constructing from any proof of A, a proof of contradiction.
5. A proof of $(\forall x : D)A(x)$ is a rule or method which for any $d : D$ constructs a proof of $A(d)$.
6. A proof of $(\exists x : D)A(x)$ is given by providing a $d : D$, and a proof of $A(d)$.

From 2 we see why the law of the excluded middle is rejected. To assert $P \vee \neg P$ in the general case would require a universal decision procedure for mathematical propositions.

Given the above definitions, propositions–as–types (aka the Curry-Howard isomorphism) jumps out, once we are given the idea. We see that a proof of $A \wedge B$ is a pair (a, b) of $A \otimes B$; that a proof of $A \vee B$ is a left or right element of the disjoint union $A \oplus B$; that a proof of $A \supset B$ is a function in $A \to B$; that Absurdity is the empty type. The relation of proof to proposition proved is seen to be the same "::" judgement already met, of a term to its type.

The interpretations of rules 5 & 6 for the quantifiers require, respectively, dependent function and product types. In stating these rules I have the range of quantification, D, a type. Universal quantification over a set, $(\forall x \in S)A(x)$, can be translated as $(\forall x : T)P(x) \supset A(x)$ where P is the defining property of set S in type T. Similarly $(\exists x \in S)A(x)$ can be translated $(\exists x : T)P(x) \wedge A(x)$.

The coincidence of the types of closed terms in typed λ-calculus with the tautologies of intuitionistic implication, and of the terms themselves with natural deduction proofs of these formulae, was first noted in Curry & Feys (1958). Howard (1969) extends this to a lambda calculus with sums and products and full intuitionistic first order logic.

The *Intuitionist Theory of Types* of Martin-Löf (1973) adds equality types, natural numbers and induction, and a hierarchy of universes so that types have types. This is both a functional programming language and a formal system for constructive mathematics in the sense of Bishop (1967) — with fully formal proofs. It is strongly normalising and has decidable typing judgement (which is the relation between element and type aka that between proof and theorem).

This has given rise to a number of descendant constructive type theories, including Homotopy Type Theory, and computer systems for developing proofs

[2] I have used "rule or method for constructing" where Troelstra & van Dalen say "construction for transforming".

in constructive mathematics alias functional programs, including Robert Constable's Nuprl at Cornell, Agda (Norell, 2007) designed at Chalmers by Caterina Coquand and Ulf Norell, and COQ, developed at INRIA and based on the Calculus of Inductive Constructions, a development from Coquand & Huet (1988).

Mathematicians have a century of highly successful work based on set theory and it is unsurprising that constructive mathematics has to date, from Bishop (1967) on, often been done using constructive versions of set theory. I believe this will change as systems based on type theories like those above continue to develop and become more convenient to use.

It is in any case likely that the publication standard in mathematics (whether classical or constructive) will come to require the submission of fully formal machine checked versions of proofs.

A note on the Axiom of Choice

The principle of dependent choice

$$(\forall x : A)(\exists y : B)C(x, y) \supset (\exists f : A \to B)(\forall x : A)C(x, f(x))$$

is easily provable in Martin-Löf type theory and its descendents.

The same principle in set theory is an axiom, the *axiom of choice* (AC), whose constructive status is widely regarded as problematic.

Bishop (1967) remarks in his introduction that "choice is implied by the constructive meaning of existence". But in the development of constructive analysis Bishop is working within an (informal) set theory close to Myhill's CST, which does not include AC.

Goodman & Myhill (1978) give a proof that in ZF set theory AC implies the law of the excluded middle, $P \vee \neg P$. So AC is omitted from constructive versions of ZF for good reason.

Goodman & Myhill's proof uses two other axioms of set theory besides AC: the axiom of pairing, that the set $\{a, b\}$ exists for any (not necessarily different) a, b and the axiom of extensionality. Pairing is a special case of the axiom of separation.

The axiom of separation is not part of Martin-Löf type theory — by conflating types and sets it would destroy the decidability of the typing judgement. That the choice principle for types is constructively valid while AC (for sets) is not, is telling us that the latter is a stronger claim.

There have been proposals to add a subtyping construct, analogous to the axiom of separation, to constructive type theory. In the light of Goodman & Myhill's result, this looks suspect — and Thompson (1992) gives arguments why a subtype construction is not needed in type theory.

2 Church's Thesis and constructivity

In or around 1936, an objective definition of *effective computability* emerged: the general recursive functions of Herbrand, Gödel and Kleene, the λ-definable functions of Church, and those computable by Turing's "logical computing machines" were found to be one and the same class of number theoretic functions, of which further definitions have since been found.

These are the *partial recursive functions*, whose types are $N^k \rightarrow \overline{N}$, with N the type of the natural numbers and $\overline{N} = N \cup \{\perp\}$ where $\perp$ stands for undefined or non-terminating. Without loss of generality we can restrict ourselves to the type $N \rightarrow \overline{N}$; by the use of Gödel numbering, elements of N can be used to represent any finite input or output data. The *recursive functions* are the *total* partial recursive functions, whose type can be written $N \rightarrow N$.

Church's Thesis (jointly due to Church and Turing) is the conjecture that this class identifies what is computable by any realisable mechanism[3] By mechanism we here mean one that takes a symbolic input and produces a symbolic output; a device with analog components is not excluded provided it meets that condition.

The thesis provides an objective basis to computability results that underly computing science. It is shown that such and such is not recursive (aka not computable by a Turing machine etc.), and Church's Thesis allows us to say that the proposed function is *not effectively computable*. The earliest example of this mode of reasoning is Church (1936b) where he argues from the results in Church (1936a) that the decision problem for predicate calculus is *unsolvable*. It is striking that in stating his conclusion Church uses the word "unsolvable" without any qualifying adverb.

What is equally important and emerged together with Church's thesis is the *undecidability of termination*. Given a computation of type $\overline{N}$ there is no generally effective procedure for determining if it is $\perp$. From this other undecidability results are derived, including that for first order predicate logic.

Church's thesis is empirical

Gandy (1980) shows that assuming a small number of rather general physical principles — finite (although possibly changing) number of parts, no infinitesimal parts, no instantaneous action at a distance etc. — what can be computed by any machine is recursive.

Quantum computers lie outside Gandy's assumptions by the use of quantum entanglement. They make feasible certain computations that would be exponential on a conventional computer, for example factorizing an integer in linear time. But they do not allow the computation of anything that is not recursive. See Rieffel and Polak (2000) for a survey.

[3] In his analysis of Church's Thesis, Gandy (1980) distinguishes between two claims, which he calls T and M. Theorem T says anything that can be computed by an idealised human being following rules, with an unlimited supply of paper and a pencil, is recursive (alias computable by a Turing machine). Gandy calls it a theorem because, although not formally provable, it is intuitively clear that a Turing machine models an idealised human computer. Thesis M says anything that can be mechanically computed is recursive. This implies T but is stronger. Hodges (2006) offers evidence that Church and Turing held the stronger claim, and this is what I take as Church's Thesis in this paper.

Church's thesis reflects the fact that, according to our current understanding, our universe (or at least the part of it which can have causal effects on us) contains no infinities and no infinitesmals, the latter because space, time and matter are all quantised.

It is thus a conjecture of physics, rather like the second law of thermodynamics, for which we have strong scientific evidence but no proof. Deutsch (1997) takes exactly this view.

It should therefore not be built into constructive mathematics. Mathematics should have the ability to describe universes other than our own.

Are there non-computable functions?

In classical mathematics non-computable functions flow from the law of the excluded middle and its friend the law of double negation. It is enough to write a *specification* for a function, prove that it is not absurd, and then, as if by magic, we have the function without having provided a method of computing it. In his "Constructivist Manifesto" Bishop[4] sees these as an obscuring fog that must be blown away to reveal a leaner, algorithmic structure.

If we accept the constructive definition of function as *rule or method* then non-computable function is an *oxymoron*, like a round square or a four cornered triangle. See Greenleaf (1992) "Bringing Mathematics Education into the Algorithmic Age".

A non-recursive function is logically possible—because Church's Thesis might be false—but to demonstrate the existence of one you would have to produce a method for computing it that others can understand and use. Since CT is almost certainly true this is not expected to happen. So constructive mathematics is destined to remain in the position that it can neither produce an example of a non-recursive function nor a proof that none exist. I don't see this as a problem, it seems entirely reasonable.

A fallacious argument from cardinality is sometimes advanced to "prove" that non-recursive functions both exist and greatly outnumber the recursive ones:

1. $N \to N$ is uncountable, by diagonalisation (Cantor).
2. The partial recursive functions $N \to \overline{N}$ are recursively enumerable
3. The total recursive functions, as a subset of (2), are countable
4. Therefore "almost all" members of $N \to N$ are non-recursive

Classically this is unproblematic. Viewed constructively, the argument fails at step 3. An effective enumeration of a set yields an effective enumeration of a subset only if the subset is decidable. The proof of (3) from (2) requires a *totality* test and totality for partial recursive functions is not recursively decidable. The argument is circular; it assumes a non-recursive function to prove that non-recursive functions exist.

The underlying issue is that the classical theory of transfinite cardinals is almost entirely non-constructive. Constructively, a set that is not countable has

[4] Bishop & Bridges (1985) Ch 1.

a *more complex internal structure* than a countable one but that does not necessarily make it "bigger" nor prevent it from being a subset of the latter. The ordering of transfinite sets by cardinality (trichotomy) is equivalent to the axiom of choice, which is not constructively valid.

Note that the total recursive functions constitute a set whose countability we cannot prove from the given facts. This is different from a set whose countability is absurd (leads to contradiction) meaning it is provably *uncountable*. It is the absence of the law of the excluded middle that separates these two situations.

In constructive mathematics the set $N \to N$ is uncountable, by Cantor's diagonalisation proof[5]. This does not imply the existence of non-recursive functions, which is logically independent.

To show this we use the result, established by Bridges & Richman (1987), that Bishop's constructive mathematics, BISH in their terminology, sits in the logical intersection of classical mathematics (CLASS), the recursive mathematics of Markov (RUSS), and Brouwer's intuitionism (INT), although the last is not relevant here. Anything provable in BISH is provable in the other three systems.

Conversely, any proposition that would lead to a contradiction in CLASS, RUSS, or INT, cannot be provable in BISH.

We see straight away that there cannot be a constructive proof that there are non-recursive functions because that would lead to a contradiction in RUSS, which incorporates Church's Thesis as an axiom[6]. Nor can there be a constructive proof that all functions are recursive — that is impossible in CLASS by the cardinality argument.

From which we conclude that the existence of non-recursive functions can be neither proved, nor disproved constructively.

Even in recursive mathematics $N \to N$ is internally (that is recursively) uncountable by the usual diagonalisation proof, despite being externally countable from a classical viewpoint. There are no objective grounds for regarding the classical view as the "correct" one.

In summary: there is no constructive proof that non-recursive functions exist and constructive $N \to N$ is uncountable, regardless of their presence or absence.

Similar results apply concerning the status of non-recursive real numbers, which have been latched onto by some as a possible analog route for evading Church's Thesis (ignoring quantum mechanics).

In a survey of schemes for "Hypercomputation" by Stannett (2004), we find this statement[7]: "As is well known, a randomly selected real number has a 100% chance of being non-recursive.". An appeal to the, constructively invalid, cardinality argument, it raises another interesting question—how can a point in say, the interval $[0, 1]$ be "randomly selected"? Presumably by an infinite free choice sequence, to which topic we now turn.

[5] This is a proof by contradiction of non-existence, which is uncontroversial—what is not allowed, intuitionistically, is a proof by contradiction of existence.

[6] See Bridges & Richman (1987) Ch 3.

[7] Teuscher (2004) p152.

3 Free Choice Sequences

Do free choice sequences contradict Church's Thesis?

Bishop is dismissive of Brouwer's use of free choice sequences for analysis, and develops constructive analysis without them. He writes

> 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. He therefore introduced the method of free-choice sequences for constructing the continuum, as a consequence of which the continuum cannot be discrete because it is not well enough defined.[8]

However, this doesn't settle the question of whether free choice sequences exist, and if so, do they give us access to something non-recursive? Concerning existence, we do not need to get entangled in discussions about whether humans have free will and in what sense.

A genuinely random, i.e. not rule-governed, sequence of bits can be generated from quantum uncertainty. It is possible to design a practical piece of equipment that detects atomic transitions or some other event subject to quantum uncertainty to produce a random sequence of numbers in some finite range $[0, k]$, of any desired length. An apparatus of this kind, ERNIE (for Electronic Random Number Indicator Equipment), first employed in 1956, is used to pick monthly winning numbers for UK government premium bonds.

So does ERNIE, by its randomness, give us access to a non-recursive function? Of course not[9]. The free choice sequences generated by ERNIE and his friends are not rule-generated but they are *always finite*. To get something that might be non-recursive we would have to run ERNIE *forever* — and forever is too long to wait. We are mortal as indivduals and as a society; any apparatus we build will eventually stop working, our civilisation will eventually come to an end.

A finite sequence of natural numbers, however long, is always computable. Computable means generable by a rule or method. A rule or method is something that can be used in another time or place to get the same results. For a finite sequence you can record it, and replay it when needed or transmit it to another place to be replayed. Also for any finite sequence, there are an infinite number of algorithms that will generate it and that will remain true as it is extended.

Suppose someone resists this argument and insists that we consider a free choice sequence that runs forever and tries to argue that this has probability 1 of being a non-recursive function in $N \to k$ (appealing to the fallacious cardinality argument). My response is that what we are being asked to consider is not a function. To be given a function, in the general case, we have to be given the function *in its entirety*. Some questions can be answered by looking at points, say $f(0)$, $f(17)$, etc, but to prove something about a function in the general case you

[8] Bishop & Bridges (1985), p9.

[9] It was shown in (De Leeuw et al., 1956) that adding a random number generator to a Turing machine does not enlarge the class of functions it can compute.

need the whole function, and a function on N can only be given, constructively, by a rule or method expressed in finite form.

The proper framework for understanding an indefinitely proceeding free choice sequence is as *codata* rather than as a function on N (see e.g. Turner, 2004). What we have is a *colist* of digits which we can interrogate to get the next digit and another colist. In the case under consideration we are not given any rule governing the sequence of digits.

To conclude, I do not see free choice sequences as having any impact on the plausibility of Church's Thesis (that every computable function is recursive), because if finite they are recursive and if considered as running indefinitely they are not functions within the constructive meaning of the term.

References

Barwise, J., Keisler, H.J., Kunen, K (eds): The Kleene Symposium. North-Holland, Amsterdam (1980)

Beeson, M.J.: Foundations of Constructive Mathematics. Springer Verlag (1985)

Bishop, E.: Foundations of Constructive Analysis. McGraw-Hill (1967), revised and reissued as Bishop & Bridges (1985)

Bishop, E., Bridges, D.: Constructive Analysis. Springer-Verlag (1985)

Bridges, D., Richman, F.: Varieties of Constructive Mathematics. Cambridge University Press (1987)

Brouwer, L.E.J.: On the Unreliability of the logical principles. 1908. New translation with introduction by van Atten, M. & Sundholm, G. (2015) https://www.researchgate.net/publication/283448904

Church, A (1936a): An Unsolvable Problem of Elementary Number Theory. American Journal of Mathematics 58, 345–363 (1936)

Church, A. (1936b): A Note on the Entscheidungsproblem. Journal of Symbolic Logic 1(1), 40–41 (1936)

Coquand, T., Huet, G.: The Calculus of Constructions. Information and Computation 76, 95–120 (1988)

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

De Leeuw, K., Moore, E.F., Shannon, C.E., Shapiro N.: Computability by Probabilistic Machines. In: Automata Studies, eds Shannon, C.E., McCarthy, J., Princeton University Press, pp. 183–212 (1956)

Deutsch, D.: The Fabric of Reality. Allen Lane, The Penguin Press (1997)

Frege, G.: (1879) Begriffsschrift: a formula language, modeled on that of arithmetic, for pure thought. In: van Heijenoort(1967), pp. 1–82

Frege, G.: (1892) On Sense and Reference. In: Geech & Black (1966), pp. 56–78

Friedman, H.: The Consistency of Classical Set Theory Relative to a Set Theory with Intuitionistic Logic. Journal of Symbolic Logic 38(2), 315–319 (June 1973)

Gandy, R.: Church's Thesis and Principles for Mechanisms. In: Barwise, Keisler & Kunen (1980), pp. 123–148

Geech, P., Black, M.: Translations from the Philosophical Writings of Gottlob Frege. Basil Blackwell, Oxford (1966)

Goodman, N.D., Myhill, J.: Choice Implies Excluded Middle. Zeit. Logik und Grundlagen der Math 24, p. 461 (1978)

Greenleaf, N.: Bringing Mathematics Education into the Algorithmic Age. In: LNCS 613, pp. 199–217 (1992) van Heijenoort, J.: From Frege to Gödel — a source book in mathematical logic 1879–1931. Harvard University Press (1967)

Hindley, R.J, Seldin, J.P. (eds): To H.B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism. Academic Press (Sep. 1980)

Hodges, A.: Did Church and Turing have a thesis about machines? In: Olszewski et al. (2006), pp. 242–252

Howard, W.A.: The Formulae as Types Notion of Construction. (original paper manuscript of 1969) In: Hindley & Seldin (1980), pp. 479–490

Martin-Löf, P.: An Intuitionist Theory of Types — Predicative Part. In: Logic Colloquium 1973, eds Rose and Shepherdson, pp. 73–118, North Holland (1975)

Myhill, J.: Constructive Set Theory. Journal of Symbolic Logic 40(3), 347–382 (Sep. 1975)

Norell, U.: Towards a practical programming language based on dependent type theory. PhD Thesis. Chalmers University of Technology (2007)

Olszewski, A., Woleński, J., Janusz R. (eds): Church's Thesis after 70 years. Ontos Verlag, Berlin (2006)

Rieffel, E., Polak, W.: An Introduction to Quantum Computing for Non-Physicists. ACM Computing Surveys 32(3), 300–335, (Sep. 2000)

Russell, B. The Principles of Mathematics. Cambridge University Press (1903)

Stannett, M.: Hypercomputational Models. In: Teuscher (2004), pp. 135–157

Teuscher, C. (ed.): Alan Turing: Life and Legacy of a Great Thinker, Springer-Verlag, Berlin (2004)

Thompson, S.: Are subsets necessary in Martin-Lof type theory? In: Symposium on Constructivity in Computer Science, San Antonio, TX, June 1991, LNCS 613, pp. 46–57 (1992)

Troelstra, A.S. & Dalen, van D.: Constructivism in Mathematics. Studies in Logic and the Foundations of Mathematics 121 & 123, Amsterdam: North-Holland (1988)

Turner, D.A.: Total Functional Programming. Journal of Universal Computer Science 10(7), 751–768 (July 2004)

Univalent Foundations Program: Homotopy Type Theory — Univalent Foundations of Mathematics. Institute for Advanced Study, Princeton (2013). https://homotopytypetheory.org/book/