Kent Academic Repository

Corfield, David (2017) *Expressing 'The Structure of' in Homotopy Type Theory.* Synthese . ISSN 0039-7857.

Downloaded from

https://kar.kent.ac.uk/63390/ The University of Kent's Academic Repository KAR

The version of record is available from

https://doi.org/10.1007/s11229-017-1569-7

This document version

Author's Accepted Manuscript

DOI for this version

Licence for this version

UNSPECIFIED

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in ***Title of Journal*** , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Kent Academic Repository

Full text document (pdf)

Citation for published version

Corfield, David (2017) Expressing 'The Structure of' in Homotopy Type Theory. Synthese . ISSN 0039-7857. (In press)

DOI

https://doi.org/10.1007/s11229-017-1569-7

Link to record in KAR

http://kar.kent.ac.uk/63390/

Document Version

Author's Accepted Manuscript

Copyright & reuse
Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research
The version in the Kent Academic Repository may differ from the final published version.
Users are advised to check **http://kar.kent.ac.uk** for the status of the paper. **Users should always cite the published version of record.**

Enquiries
For any further enquiries regarding the licence status of this document, please contact:
researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at **http://kar.kent.ac.uk/contact.html**

KAR Kent Academic Repository

University of Kent

Expressing ‘The Structure of’ in Homotopy Type Theory

David Corfield, Department of Philosophy, University of Kent

Abstract

As a new foundational language for mathematics with its very different idea as to the status of logic, we should expect homotopy type theory to shed new light on some of the problems of philosophy which have been treated by logic. In this article, definite description, and in particular its employment within mathematics, is formulated within the type theory.

Homotopy type theory has been proposed as an inherently structuralist foundational language for mathematics. Using the new formulation of definite descriptions, opportunities to express ‘the structure of’ within homotopy type theory are explored, and it is shown there is little or no need for this expression.

1 Introduction

Homotopy type theory has recently been proposed as a new foundational language for mathematics (UFP 2014). This system possesses many novel features compared to the traditional set theoretic foundations, not least the way it integrates logic within itself. Set theory is standardly formulated as a first-order axiomatic theory whose domain ranges over sets and which is equipped with a binary membership relation. Homotopy type theory (henceforth HoTT), on the other hand, is given largely by type formation rules, and rules for the introduction and elimination of terms. Logic, in the shape of a form of propositional and predicate logic, comes built into the type theory, arising from the application of its rules to a certain class of types. Were HoTT ever to take the place of the formal languages currently employed by analytic philosophers, it would involve a significant change of outlook. HoTT possesses the resources to derive, as matters simply arising from its conception of identity, results about homotopy groups of spheres and about group representations, topics normally seen as pieces of advanced mathematics. Were its ‘modal’ and ‘linear’ variants to become accepted too, this extended logic would also have the resources to speak about Noether’s theorem, relating symmetries to conservation laws, and also about geometric quantization (Schreiber 2014, 2016).

Whether we take it as an extended logic, or consider it to entail that logic is to be subsumed within mathematics, an early consideration as to how to engage philosophically with HoTT will be to revisit what thinkers took to be promising applications of previous forms of logic. One of the first applications by Russell of his ‘new logic’ (Russell 1905) was the analysis of definite descriptions. It seems fitting then to see what the latest ‘new logic’ can bring to this question. We will find that a reasonable way to approach this topic casts light on mathematicians’ usage of ‘the’ in a generalised sense, as when they say ‘the product of two groups’, apparently without there being a unique way to construct such a product.

In the second half of the paper, I shall apply this account of definite description to the phrase ‘the structure of A’, for a general type A. Awodey (2014) claims that, through its so-called ‘Univalence Axiom’, HoTT captures what is essentially right about the structuralist position. Shulman (forthcoming) agrees, describing it as a ‘synthetic theory of structures’, in the sense that nothing can be said about mathematical entities defined within it except structurally. We should not expect then to need to, or even to be able to, construct something substantially different from A when contemplating its structure. This is indeed what we find.

It is important to note that naming conventions have not been definitively settled yet. Some people have looked to distinguish *Univalent Foundations* from HoTT, but for the purposes of this article, I shall be working on the understanding that HoTT is a variety of intensional Martin-Löf dependent type theory with higher inductive types and satisfying the Univalence Axiom. I shall be referring to all of these ingredients in the course of this article as we need them.[1]

2 Definite descriptions

2.1 A sketch of dependent type theory

Attempting to make philosophical use of HoTT to a general audience presents an immediate problem in that familiarity with the system cannot be assumed. Were one looking to introduce first-order logic and set theory to a similarly uninitiated audience, it would be reasonable then to illustrate the former and naïve aspects of the latter in terms couched in everyday language, rather then heading straight into pure formal set theory. I shall adopt a similar stategy here, noting that for the most part the discussion of even the underlying dependent type theory has been in terms of its providing a formal language for mathematics. It is worth observing, however, that there are several treatments of natural language in terms of dependent type theory, one of the best and most thoroughgoing of which is Ranta (1994), many of whose ideas are still relevant, I believe, when passing to HoTT. Even so, for dependent type theory or indeed HoTT to play a central role in philosophy of language or metaphysics, further interpretative work would be necessary.

For example, at the core of a type theory, naturally enough, we find ‘types’. In the mathematical case, these are constructed by certain formation rules. What must be established is (i) what it is to be an element of a type, A, and (ii) what it is for two elements of A to be equal.[2] We find then a commitment to the principle that any named element or variable will always appear as belonging to a specified type. I can never ask then whether some term belongs to a proposed type, but will employ expressions such as $a : A$, where A is a type that has already been given. Either I am declaring some entity to belong to a type, or I am deriving this from rules which force the type ascription.

The distinction to be found in philosophy of language as to the different identity criteria employed when speaking of an airline serving n people in a year and yet carrying m passengers translates to a difference between the types

[1] This is in line with the ‘HoTT Book’ (UFP 2014), which provides the most detailed account of the system in a reasonably accessible way. See also Shulman (2017) for an alternative, excellent introduction.

[2] I shall use the expressions ‘element’ and ‘term’ as near synonyms, the latter being appropriate when syntactical considerations are in greater focus.

Person and *Passenger*. The difference between the identities involved in 'I just saw the same car that bumped into yours yesterday' and 'I drive the same car as you' again points to different types being under consideration. How best to consider types as sorts or kinds in the world is perhaps still to be established.

We also need the idea of one type depending on another type. So in mathematics we may have a collection of types indexed by the natural numbers, such as the type of $n \times n$ matrices over a given field. In natural language we might consider the type of players playing for a team t, as t varies over some type of teams. We express this as

$$t : Team \vdash Players(t) : Type.$$

These dependent types are sets, in a sense to be defined later, but we can have examples where they are propositions, such as

$$t : Team \vdash Plays\ in\ UK(t).$$

Quantification then takes place in these dependent type situations, where we find that domains of variation are the indexing types. This relies on the type formation of *dependent sum* and *dependent product*. For the dependent type $B(x)$ depending on $x : A$,

- The dependent sum (sometimes known as dependent pair) $\sum_{(x:A)} B(x)$ has as elements pairs (a, b), where $a : A$ and $b : B(a)$. In the case of the teams above, this amounts to pairs of (team, player in that team) or of (team, proof that it plays in the UK).

- The dependent product, $\prod_{(x:A)} B(x)$, has as elements maps, f, defined on A, such that $f(a) : B(a)$. In the case of teams above, this amounts to a choice of one player from each team or a proof that each team plays in the UK.

Quantification is associated to the second example in each case where the dependent type is a proposition. The dependent sum being inhabited amounts to the existence of a team that plays in the UK, and the dependent product being inhabited amounts to all teams playing in the UK.

The difference with an untyped setting is very apparent when we look to express something with multiple quantifiers, such as 'Everyone sometimes finds themselves somewhere they don't want to be'. In type theory there will be dependency here separately on types of people, times and places, and not variation over some universal domain, requiring conditions that specify that some entities in the domain be people, times or places.

Göran Sundholm's resolution of the puzzle of the farmer sentence is perhaps illuminating here (Sundholm 1986):

- If a farmer owns a donkey, then he beats it.

The problem here is that we expect there to be a compositional account of the meaning of this sentence, in particular, one where the final 'it' appears in the representation. At first glance it appears that an existential quantifier is involved because of the indefinite article, and yet a beginner's attempt to use one is ill-formed, the final y being unbound:

$$\forall x(Farmer(x) \& \exists y(Donkey(y) \& Owns(x,y)) \rightarrow Beats(x,y)).$$

The alternative in standard first-order logic is to rephrase the sentence as something like: 'All farmers beat any donkey that they own', and then to render it formally as

$$\forall x(Farmer(x) \rightarrow \forall y(Donkey(y) \& Owns(x,y) \rightarrow Beats(x,y))).$$

But now we have radically transformed the original sentence, and the 'it' does not seem apparent.

Sundholm showed how we could have our cake and eat it, a compositional and faithful account of the sentence using the resources of dependent type theory:

$$\prod(z : (\sum(x : Farmer)\sum(y : Donkey)Owns(x,y)))Beats(p(z), p(q(z))).$$

Elements of the dependent sum of donkeys owned by farmers are pairs formed of a farmer and then a pair formed of a donkey and a warrant that the donkey is owned by that farmer. From such an element, z, we project to the first component of the pair, $p(z)$, to extract the farmer and then project to the first component of the second component, $p(q(z))$, for the donkey, so at to be able to express the beating of one by the other. It is this last term that is being referred to in natural language as 'it'. An element of the whole dependent product, which will establish the truth of the proposition, will provide a proof of the relevant beating for any such z. Now let us turn to definite description.

2.2 Definite description in natural language

We use 'the' in a number of somewhat related ways:

- The Prime Minister of the United Kingdom is right-handed.
- The Romans invaded Britain in 43 AD.
- The platypus is a nocturnal creature.

For the purposes of this note, I shall be considering its use in definite descriptions, as in the first of the examples above, where 'the' is followed by a singular noun, perhaps restricted in some way, since this seems to be the case with both Russell's 'the present King of France' and 'the structure of A'. I consider this to be the case also in standard mathematical statements such as

- The cyclic group of order 6 has an element of order 3.

Where this last case may give the appearance of employing the kind of general 'the' used in 'the platypus' above,[3] we shall see that it is correctly taken as a case of specifying by restriction one item from a collection.

[3] I thank an anonymous referee for making this claim, even though I consider it false.

There are a range of subtleties to the use of 'the' within the singular terms of natural language, see for example Vendler (1967a, Ch. 2) who dwells on implicit restrictions from beyond the context of a sentence. For instance, it may be the case that an entity has been described in a previous sentence, in which case to produce the full expression we may have to insert a redundancy:

- Yesterday, I bought a car. The car is green.
- Yesterday, I bought a car. The car I bought yesterday is green.

Since our primary goal is to study mathematical usage, these subtleties of ellipsis need not detain us.

Now two related forms of such a use of 'the' present themselves:

1. 'The A', where A is a type of a certain kind. For example, *the donkey owned by John* as an element of the type *Donkey owned by John.*
2. 'The $f(a)$', where $a : A$ and $f : B^A$, for some types A and B (B^A being the type of functions from A to B). For example, *the mother of Julius Caesar* as an element of the type *Woman*, 'mother of' having already been formed as belonging to the type of functions from Person to Woman, and 'Julius Caesar' as belonging to Person.

The former is properly formed when there is a unique individual of the type A. This may come about by forming a singleton type from an existing type, such as 'Donkey owned by John' from the type 'Donkey'. As we have seen, such types may be formed within the type theory using *dependent sums.* In the case of 'the donkey owned by John', if indeed it is a singleton, then there will be one pair formed of an element of the main type, here a donkey, along with a warrant that the specified condition holds, here a proof that it is owned by John.

In case (2), the fact that f is a function forces the existence and uniqueness of $f(a)$, such as in the expression 'the colour of my front door', where 'colour of' is a map from some type of (monochrome) objects to the type of colours. A simple extension would allow B to depend on A, so $f(a) : B(a)$, as in *the captain of team*(a): *Player*(a), for some team $a : Team.$ In either case, we might form a singleton subtype, such as 'Colour which is the colour of my front door', and in this way, we can reduce to (1), 'the A' for some type A.

2.3 Intensional type theory and definite description for types which are not sets

Where I have been considering singleton types as though they are sets with one element, types in HoTT, as a variety of *intensional* type theory, need not be sets, but may be 'mere propositions' or may be 'higher groupoids' (UFP 2014, Chap. 3). These distinctions concern what is called the *truncatedness* or *homotopy level* of the type, a concept which relies on an important feature of HoTT, namely its identity types. As we have seen, variables and terms in a dependent type theory always appear with their associated types. We never ask of a term the type to which it belongs since its type is always explicitly declared. When we have a type already formed and two elements of that type, there is a type formation rule that allow us to form a new type of identities between these elements. So $A : Type,\ a, b : A \vdash Id_A(a, b) : Type$. No restrictions are placed on such identity types, in the sense that it is not required that two elements, p and q, of $Id_A(a, b)$ be identical. Indeed, we can form a further type, $Id_{Id_A(a,b)}(p, q)$, and then iterate this process.

Informally, *mere propositions* are taken to be types for which any two terms are equal. There may be no such terms, in which case the proposition is false, but if there is a term, in which case it is true, then there is only one term. Similarly a type is a *set* if its corresponding identity types are mere propositions, so that an answer to whether two terms are the same is just 'yes' or 'no'. A type is a *groupoid* if its identity types are sets, a *2-groupoid* if its identity types are groupoids, and so on.[4] Note that this hierarchy of levels is cumulative in the sense that a type which is a mere proposition (-1-type) is also a set (a 0-type), a groupoid (1-type), and so on. Indeed any m-type is also an n-type for $n \geq m$.

On the face of it then, given two sets, A and B, we might imagine that the type of products of A and B would form a groupoid rather than a set, there being a set of ways that two constructed products are isomorphic. To count as such a product, any such type must be equipped with projections to A and to B, which satisfy certain conditions. It would appear then that both the obvious product composed of ordered pairs, $A \times B$, and the type of reverse pairs $B \times A$ with the projection from second place to A and from first place to B would represent a product of A and B. Using the convention of naming a type by the kind of its elements, we have $Product(A, B)$ as the type whose elements are sets which behave as a product of A and B should. The identity type $Id_{Product(A,B)}(A \times B, B \times A)$ might then appear to contain a non-singleton *set* of elements in which case $Product(A, B)$ is not a set.

However, it is well known that mathematicians will say '*the* product of two sets'. Category theory has explained how to think of this case as one where, when a construction has been defined by a *universal property*, it does not matter which representative one takes as product. This is because there is a *canonical* isomorphism between any two representatives as given by the universal construction. Indeed, for A and B objects in a category $\mathcal{C}$, 'the' product of A and B is defined as an object, P, with arrows (projections), $p_1 : P \to A$ and $p_2 : P \to B$, such that for any object Q of $\mathcal{C}$ equipped with maps, f to A and g to B, there is a unique arrow $t : Q \to P$, such that $f = p_1 \circ t$ and $g = p_2 \circ t$. An early exercise in category theory has one demonstrate that given two such products, P_1 and P_2, there are unique arrows in each direction between them, which when composed in each order yield identity maps. This establishes that P_1 and P_2 are isomorphic, and canonically so as the isomorphism derives from specified unique arrows.

Category theory and type theory work hand-in-hand here. The universal nature of the product construction applied to all homotopy types as described by the former is perfectly captured in the combination of the four rules of type formation, term introduction, term elimination and computation (UFP 2014, § 1.5). Due to the conditions of the definition of 'Product', there is in fact only a

[4] The expression n-groupoid comes from algebraic topology. They arise, for example, when considering a topological space, a collection of points in the space, paths between pairs of points, paths between paths with the same endpoints, and so on. But they may also be treated 'algebraically' as corresponding to a special kind of n-category, where the morphisms, morphisms between morphisms, etc., are all invertible up to some weak equivalence (see Corfield 2003, Chaps. 9 and 10).

single element of the identity type $Id_{Product(A,B)}(A \times B, B \times A)$, namely, the map which reverses the order of the pairs. We could say that the type of products of two sets is a groupoid in which objects (namely, sets behaving as a product) are related coherently by unique morphisms, or in other words that the groupoid of products is equivalent to the trivial groupoid. Up to equivalence as a homotopy type (UFP 2014, §2.4), such a groupoid is equivalent to a singleton set. Taking our cue from topology, we name this property 'contractibility', in the same way as a contractible space is equivalent to a space composed of a single point. A type which is contractible sits at the lowest level of the hierarchy, at level -2. That a contractible type be describable as a singleton set or as a trivial groupoid reflects the convention stated above that the level hierarchy is cumulative.

Naturally, there is a construction in HoTT which defines what it is for a type to possess this property (UFP 2014, § 3.11):

$$X : Type \vdash isContr(X) \equiv \sum_{(x:\, X)} \prod_{(y:\, X)} Id_X(x, y) : Type.$$

To find an element of this dependent sum requires us to produce an element, say a, of the type X, and then an element of the subsequent dependent product. To specify such an element, we are looking for a coherent[5] collection of identities between a and each element of the type. In the case of the type $Product(A, B)$ for two sets A and B, the type containing any type that acts as such a product, we have a representative $A \times B$, and for any other representative, a canonical isomorphism, such as the switching map from $A \times B$ to $B \times A$.

One level down, consider when X is a set in the HoTT sense. Then we find that X is contractible precisely if we can find an element, and every element is equal to this one. In other words, as expected, a contractible set is a singleton. On the other hand, when X itself is a 'mere proposition', then contractibility amounts to X being inhabited, and so being true.

My proposal then is that we should only form the term 'the X' for a given type X once we have established that $isContr(X)$ is inhabited. Of course, in the context of an assumption that X is contractible, we should be able to form 'the X' as a term depending on the type $isContr(X)$, but until we have constructed an element of $isContr(X)$, we cannot form 'the X' in an assumption-free way. In the case of the type *Present King of France*, without the possibility of establishing unique existence since it lacks any element, Russell's term 'the present King of France' should not be introduced in a non-hypothetical way, in which case it is not available to be used to construct the proposition 'The present King of France is bald.' Rather than conjoining presuppositions into the full expression of a proposition, Martin-Löf-style type theories, such as HoTT, form a proposition, as any type, within a *context*, constructed with a valid dependency structure.[6] Conditions must be in place for constructions to be permissible.[7]

[5] There is a subtlety here that interested readers can read about in (UFP 2014, Remark 3.11.2)

[6] Contexts appear on the left hand side of the symbol '$\vdash$'. $\Gamma \vdash a : A$ expresses the judgment that a belongs to A under the assumptions in Γ. Here, A and a typically depend on variables appearing in Γ. See UFP 2014, Appendix, for details.

[7] The type theorist might say that a judgment of unique existence is presupposed. Without this presupposition, questions concerning the possession of properties simply don't arise. This finds an early echo in Collingwood's *An Essay on Metaphysics*: "To say that a question 'does

Now we can describe formally the rule of what we might call *the introduction*:

$$X : Type, (x, p) : isContr(X) \vdash the(X, x, p) \equiv x : X,$$

In natural language, once we have a type, A, and have established the existence of a unique member, a, of A, we say merely 'the A' rather than tagging this term with the information a and the proof that it is unique. In, say, 1780, when Louis XVI could provide the first component of an element of $isContr(Present\ King\ of\ France)$, then we could introduce the term 'the present King of France' in a non-hypothetical way as equal to Louis XVI. In cases in mathematics where A is not necessarily a set, we may remain sensitive to the mode of construction of the a appearing in the element acting as a warrant for contractibility, although we are usually less sensitive to p, the proof of uniqueness. Syntactically there is a difference between 'the (A, a, p)' and 'the (A, a', p'), although there is a canonical identity between these terms in A. Since $isContr(X)$ itself is a mere proposition (UFP 2014, Lemma 3.11.4), any two of its elements are equal.

We should specify that this rule is intended for those types which are named as concepts whose instances are its elements. HoTT also permits the construction of so-called 'higher inductive types' (UFP 2014, Chap. 6) which allows for the construction of types in which identities behave like the path spaces of topologically (or better homotopically) intricate spaces. A practice has begun of naming some of these higher inductive types after the spatial properties of the type as a whole. Hence in some discussions we find $Circle$, $Interval$, 2-*Sphere*, etc. The type $Circle$, for example, is defined so as to have a single element, $base$, and a named element in the identity type, $loop : Id_{Circle}(base, base)$. It behaves as a homotopy theorist would expect a circle to behave, for example, in terms of the type of mappings from $Circle$ to itself being equivalent to the integers.[8] Perhaps we could say then that $Circle$ is being used as a shorthand for 'the type which behaves like a circle', and as such formed by 'the introduction' from the type of types that behave like a circle.

Returning to the 'the introduction' rule above, now we see that contractibility makes sense of our application of 'the' to an apparent groupoid such as 'the product of two types A and B'. One might think that there could be many ways to produce such a product, but the universal property defining what it means to be a product ensures that any candidate is uniquely isomorphic to any other. It is perhaps illuminating then to consider on this reading that, strictly speaking, for a type which is a groupoid in which every pair of elements is isomorphic, but not canonically so, we should not apply 'the' to the type. We see this in the mathematical construction of algebraically closing a field, where there is a reluctance to say '*the* algebraic closure of field F' although all such closures are isomorphic, since they are not uniquely so (Henriques 2010).[9]

not arise' is the ordinary English way of saying that it involves a presupposition which is not in fact made." (1940, p. 26). In his criticism of Russell's account, Peter Strawson (1950) also speaks of questions concerning the King of France not arising (ibid., p. 330), and elsewhere of the existence and distinguishability of something answering to a definite description as "presupposed and not asserted in an utterance containing such an expression" (1964, p. 85).

[8] In homotopy theory, treating spaces up to continuous deformation, one might define the circle first as a subset of the real plane. The *fundamental group* of the circle will then select a point and look to distinguish between classes of equivalent (i.e., continuously deformable) paths. In HoTT, *Circle* is defined as a type equivalent to this fundamental group.

[9] Conrad advises for two algebraic closures of a field "...*always* keep track of the choice of

The case of 'the cyclic group of order 6' is also relevant here. We will see later how types of structured types are defined in HoTT. Let us take it then that we have a type of groups, *Group*. Now we are to pick out a subtype of this, $CyclicGroupOrder6$, to which end two ways of using dependent sum present themselves. One of these ways is to say that we have a group equipped with a specified single generator of order 6. The second is to say that we have a group and a guarantee of the existence of some (unspecified) single generator of order 6. For either of these two types, all of its elements are isomorphic. However in the second case, an element, that is, a cyclic group of order 6 but with no specified generator, has a nontrivial automorphism, as we can see from the map sending 1 to 5 in the type formed by $\{0, 1, 2, 3, 4, 5\}$ under addition modulo 6. One should therefore show the same wariness about the employment of 'the' in that case as with 'the algebraic closure of field F'. However, if a generator is specified as a part of the structure, then all is well.

Similarly we could form $Group6$, the type of groups of order 6. This type is not contractible since there are two connected components corresponding to the two non-isomorphic groups, so we are not allowed to form 'the group of order 6'. This should indicate to us again that we are not dealing with a general form of 'the' as in 'the platypus'. There are 'lawlike' things we could correctly assert about all groups with 6 elements, and yet we don't say, for example, 'the group with 6 elements has an element of order 3'. We don't because there are *two* groups of order 6.

What of contractible, and so true, propositions, taking these in the HoTT proof-irrevelant sense? Well here while we do not prefix 'the' to a proposition such as 'it is raining', we certainly do say 'the fact that it is raining'. If we wish to retain our *the introduction* rule here, we might write the type as 'Fact that such and such'.[10] Then we would have 'Fact that it is raining' as a type, and, if it is inhabited, we would designate its element by the term 'the fact that it is raining', which appears to convey well that there is no multiplicity involved, for instance, of warrant for the assertion of the proposition. If 'fact' is considered to be employable only in the case of true propositions, then 'state of affairs' might provide an alternative.

Before moving on, let us clarify what may present itself as a problem under a naïve reading of identity statements relating two definite descriptions, as in the famous case of 'The evening star is the morning star'. While we cannot identify terms belonging to different types, it is reasonable to consider these types as produced by dependent sum,

$$\sum_{(x:Star)} Shine\ brightly\ in\ morning(x).$$

Then while 'the morning star' corresponds to a star (or at least a celestial body), a warrant that it shines brightly in the morning, along with a guarantee of its uniqueness in this respect, a projection on the first component lands us in the type *Star*. The evening star is treated similarly, and now the two celestial

isomorphism. In particular, always speak of *an* algebraic closure rather than *the* algebraic closure" (Conrad *n.d.*).

[10] Choosing a case from natural language as here, we encounter the considerations which gave rise to Vendler's advice in his (1967b) and elsewhere to distinguish between facts and propositions. HoTT having been devised by logicians and mathematicians, the mathematical use of 'proposition' as theorem has been employed.

bodies can be compared under the identity criteria for stars (as understood at the time). To put this in natural language, we could say that the star which shines brightly in the morning is the same star as the star which shines brightly in the evening.

In such cases of types being defined by *relative* clauses, we could then propose a second rule called '*relative* the introduction':

$$x : A, B(x) : Type, ((a, b), p) : isContr(\sum_{(x:A)} B(x)) \vdash the\ A\ which\ is\ B((a, b), p) \equiv a : A.$$

This makes better sense of how when we say, for instance, 'The cat sitting in the basket is Siamese', we mean the property 'Siamese' to apply to elements in the Type *Cat* rather than to those in *Cat sitting in the basket* as a dependent sum.

2.4 'The' for dependent types

As we have seen, as a development of Martin-Löf type theory, HoTT makes great play of dependent types. It is worth considering, then, how 'the' might be introduced for such types. Let us begin with the most familiar case where the type depended upon is a set:

$$A : Set, x : A \vdash B(x) : Type$$

In this case we can form $x : A \vdash isContr(B(x)) : Prop$, and it may be that we can construct $x : A \vdash (b(x), p(x)) : isContr(B(x))$, which establishes that for any element of A, the dependent type is contractible. Then we can form a dependent 'the',

$$x : A \vdash the(B(x), b(x), p(x)) \equiv b(x) : B(x),$$

expressed as 'the $B(x)$'. For example, we might have a set of apartments numbered by a set of numbers, N. Then a type depending on N could be $Resident(n)$ for n in N. We may subsequently learn that each apartment is single-occupied, so that $Resident(n)$ is a singleton set for every n. Then along with $the(Resident(a))$, or in more familiar terms 'the resident of a', for a specific a in N, we also have $the(Resident(n)) : Resident(n)$ where n is a variable. we can also use so-called λ-abstraction from the $b(x)$ above, that is, take it as an expression of $(the\ resident)(n)$, where $the\ resident : \prod_{(n:N)} Resident(n)$ is a function of N picking out the unique resident at each abode.

This makes sense of the use of 'the' for a function picking out an element from non-singleton sets, such as 'the captain of team x'. We may think of this as $x : Team \vdash the\ captain(x) : Captain(x)$, where $Captain(x)$ is the type of captains of team x, a subtype of $Player(x)$, or again we could take the 'relative' option, and see '$the\ captain(x)$' as 'the player who is the captain(x)'. Then λ-abstraction again produces $the\ captain : \prod_{(x:Team)} Captain(x)$, or more naturally in the relative sense $the\ captain : \prod_{(x:Team)} Player(x)$.

As in the previous section, the above makes sense for general types as in the case of the product of two types

$$X, Y : Type \vdash the\ product(X, Y) : Product(X, Y).$$

Again we may consider *the product* as an element of a dependent product, $\prod_{(X,Y:Type)} Product(X,Y)$.

A further interesting case comes from allowing dependency on types which are not sets but rather pointed, connected groupoids or ‘delooped groups’. In effect, this is a way of providing a context in which a given group G acts on everything concerned. Corresponding to G there is a type, denoted $\mathbf{B}G$, with one element, $*$, and $Id_{\mathbf{B}G}(*,*) \cong G$. Then, a dependent type V,

$$* : \mathbf{B}G \vdash V(*) : Type,$$

is a type which is equipped with an action by G. The single element of $\mathbf{B}G$ is sent to a type, V, in $Type$, and the elements of $Id_{\mathbf{B}G}(*,*)$ are sent to automorphisms of V, respecting the composition of elements of G. Now standard rules of the type theory provide a way for HoTT to represent constructions in the field of group representation theory, but we can also characterise simple situations in which there is inherent ambiguity.

Imagine that we are in communication with one another and are looking sideways on into a space in which there are three identical balls in a row, but that we are unable to signal to each other which end of the row is which. We can describe ourselves as working in the context $\mathbf{B}S_2$, where S_2 is the two element group. This group acts simply on the row, with the non-identity element reversing the order. Now I could look to use the expression ‘the ball on the left’ to pick out one ball, but of course you cannot know which of the two ends I mean. On the other hand, I can say ‘the ball in the middle’ and successfully convey an intended ball. Left and right are not invariant under the group action, whereas being in the middle is invariant in this way. ‘Middle ball’ is a type in this context and indeed is contractible, allowing me to form ‘the middle ball’.

Another way to describe this situation is via the dependent product construction, $\prod_{(*:\mathbf{B}S_2)} Ball(*)$. Recall that elements of a dependent product are functions from elements of the type depended upon to the dependent type. In the case of group actions, there is a single element, $*$, in $\mathbf{B}G$, so a function in the dependent product picks out an element of V, but one which has all of the group elements of G leaving it invariant. In our case here, the only invariant element is the middle ball. On the other hand, we might want to speak of being positioned in *the* middle or at *the* end. We can do this by forming the dependent sum, $\sum_{(*:\mathbf{B}S_2)} Ball(*)$. This collects the orbits of that action, in other words a type formed of the elements of V, but where whenever a group element g acts on v to give v', there is an element equating v and v'. Imagine here the three balls in a row and, ignoring the trivial arrows, an arrow in each direction between the end balls, and a looping arrow at the middle ball. A little more work is necessary to render it equivalent to a set with two elements, ‘middle’ and ‘end’, namely, truncation to the set of its connected components.

Similarly, playing noughts and crosses (tic-tac-toe), I may say ‘I like to start in a corner’, but it would be reasonable also to say ‘I like to start in *the* corner’. Again, the best way to view that latter expression is as a term in (the truncation of) the dependent sum, since my initial play is invariant under the symmetries of the grid[11]. Of course, once I’ve broken the symmetry by playing in a corner, you can’t just say that you would respond to my play in the corner by a play in ‘the

[11] The dihedral group of order 8

side square', since there's a difference between an adjacent and a non-adjacent such square.

Working in such 'equivariant' contexts, that is, in the presence of a group of symmetries, even though we may not be able to label unambiguously elements of a type, the type may still be a (dependent) set, in the HoTT sense, and as such have a cardinality. So I can say 'there are three balls' in my original situation above. The similar case of two identical balls in an otherwise empty space has generated a considerable debate in metaphysics around the issue of how two things can be non-identical if there are no properties to tell them apart (Black 1952). Some have argued that despite there being no way to pick one out through communicable properties, one ball does differ from the other in the sense that it is identical to that ball itself, while the other ball isn't. If I am looking on, I can distinguish 'this ball' from 'that ball', and think "this ball is this ball and that ball is not." However, the other party in this debate sees such a property as illegitimate for purposes of identification.

We now see this philosophical disagreement being illuminated by the type theoretic understanding of there being a set of cardinality 2 in the context $\mathbf{B}S_2$. I can hear my interlocutor saying there's one ball and there's another, and agree with this claim. Their mention of 'one ball' is a term in the context of the symmetries, $* : \mathbf{B}S_2 \vdash b(*) : \mathit{Ball}(*)$, it is not a 'one' in the absolute or empty context, where nothing appears to the left of '$\vdash$'. When they then mention a 'different one', I can understand them too, this difference being invariant under the symmetry group. On the other hand, in the empty context where I have bound the free variable, the only types available are the dependent sum $\sum_{(*:\mathbf{B}S_2)} \mathit{Ball}(*)$, a set of cardinality 1, and the dependent product $\prod_{(*:\mathbf{B}S_2)} \mathit{Ball}(*)$ which is empty. To sum up, there are ways in this framework to say: there are two balls present, there is one kind of thing, but nothing is distinguishable.[12]

3 The Structure of A

Philosophers of mathematics have long discussed what is meant by the expression 'the structure of A' for a given mathematical entity A. Famously it is possible to give different constructions within set theory of sets which may be taken to represent the natural numbers (Benacerraf 1965). The structure common to these constructions is then understood by many structuralists to be what the natural numbers are, individual numbers being places in the structure. It is also thought by some of these structuralists that to isolate the structure of any construction there needs to be a way to abstract it from whatever it is that 'carries' it, and conditions should be given for when two such abstracted structures are the same (Shapiro 1997, Resnik 1997).

Now HoTT, it is claimed (Awodey 2014), via its so-called 'Univalence Axiom', captures what is essentially right about the structuralist position. By use

[12] Along similar lines, but in a more intricate setting, HoTT provides an excellent way to understand general covariance in physics (nLab). It is striking how such constructions are written into the very machinery of the type theory. Metaphysicians have also looked to treat indiscernible quantum particles, e.g., Lowe (1989). To do this matter full justice along the lines of the present discussion one would need to consider linear types, corresponding to group representations. Then a 'linear' (in the sense of linear logic) version of HoTT (Schreiber 2014) should be the right framework to extend the treatment I am giving here.

of an ‘abstraction principle’, Awodey defines a notion of structure through the isomorphism of types:

$$str(A) = str(B) \Leftrightarrow A \cong B. \tag{DS}$$

This definition is merely suggestive. Awodey is speaking informally here, which may give rise to possible misunderstandings. A more fundamental concept than *isomorphism* in HoTT is *equivalence*, which can be formulated in the language of HoTT in terms of maps with suitable properties (UFP 2014, §2.4, Chap. 4).[13] Equivalence provides the right identity criterion generally, as the Univalence Axiom pronounces. Also, the biconditional ($\Leftrightarrow$) is not part of the syntax of HoTT. It could perhaps appear in HoTT as a notational variant of equivalence in the case of two mere propositions, and yet $A \cong B$ as a type is a variant of the type of equivalences, $A \simeq B$, and in general not a mere proposition.

Awodey concludes, after a discussion of the Univalence Axiom (UA), which says of two types that if they are equivalent as defined in HoTT, then they may be considered equal as elements of the type of small types:

> ...observe that, as an informal consequence of (UA), together with the very definition of “structure” (DS), we have that two mathematical objects are identical if and only if they have the same structure:
>
> $$str(A) = str(B) \Leftrightarrow A = B.$$
>
> In other words, mathematical objects simply *are* structures. Could there be a stronger formulation of structuralism? (Awodey 2014, p. 12)

In other words, taking HoTT as our foundation, all constructions are already fully structural.

This conclusion seems to me to be correct, but here I shall adopt a different argument strategy by examining whether HoTT itself can tell us a little more about such locutions as ‘the structure of A’, ‘A and B share the same structure’ and ‘places in the structure’. Rather than invoking the Fregean notion of an abstraction principle, as Awodey does, I shall propose what appear to be the only plausible definitions within HoTT itself of the relevant terms.

I shall not engage here in a close reading of the wide array of existing structuralist positions. The main point of this section is to show that working in HoTT the kinds of concern that date back to Benacerraf largely dissolve. With our new found ability to express uniqueness up to canonical equivalence by definite description, the motivation to seek some single entity commonly related to two structurally equivalent entities is removed. Our reconstruction of ‘the structure of A’ essentially requires our generalized ‘the’, as applying not only to sets but to any types.

Two plausible options present themselves according to different naming conventions for types that we saw in the previous section. Recall that we have (1) $Circle$ as the type which behaves like the circle, (2) $Natural\ number$ or $\mathbb{N}$, the type of natural numbers. An element of (1) is not a circle, where an element of (2) *is* a natural number. In the case of ‘the structure of A’, then, we may mean:

[13] “In general, we will only use the word *isomorphism* (and similar words such as *bijection*, and the associated notation ($A \cong B$) in the special case when the types A and B “behave like sets”.” (UFP 2014, p. 78)

1. The type which behaves like the structure of A.

2. The unique element up to equivalence of a type $Structure\ of\ A$.

Option (1) needs further unpacking. Perhaps we might recast it as 'the type which behaves structurally like A'. But then this seems to be no different from 'A' itself. If so, 'the structure of' is the identity map on types, and completely redundant. This does tally with Awodey's solution, where since $str(A) :\equiv A$, we would have $str(A) = str(B)$ as definitionally equal to $A = B$, and so, trivially, equivalent:

$$str(A) = str(B) \simeq A = B.$$

Then we never need utter 'the structure of' again.

Let's now pursue option (2). To be in a position to define 'the structure of', we will first consider the expression 'structure of' as applied to a type in the system. Together with our analysis of definite description in the previous section, we will then be able to interpret 'the structure of A'. Finally we consider 'places in' such a structure, and extend these analyses to structured types.

So, if we agree with the analysis of section 2, then to be able to say 'the structure of A' by *the introduction* we must already have (a) formed a type 'Structure of A', and (b) established that it is contractible. Now one plausible candidate for 'Structure of A' is the type

$$Structure(A) :\equiv \sum_{(X:\mathcal{U})} Equiv(A, X),$$

where $\mathcal{U}$ is the type universe of small types, sometimes written $Type$. This is an eminently reasonable choice since elements of this type are types equipped with an equivalence with A, we might say 'types-structured-as-A'. What is required now is to establish the contractibility of this type of such types. Intuitively this should be clear as contraction can take place to A, as it were, along the given equivalences. But, of course, a proof in HoTT requires use of its technical apparatus which I will briefly sketch.

Straight off we have an element of that type to hand, namely (A, Id_A). A is structured as A as witnessed by its identity map. Then to establish $isContr(Structure(A))$ we also need for every $B : \mathcal{U}$ and $f : Equiv(A, B)$ a canonical way to identify (A, Id_A) and (B, f). What such an identity amounts to in the case of a dependent sum is a path in the base type, here that is one in $\mathcal{U}$ between A and B, and a path over this one in the total space of equivalences to A. For the former we use the path that the univalence equivalence makes correspond to f. The effect of transporting $Id_A : Equiv(A, A)$ in the total space will then be $f \circ (Id_A) = f : Equiv(A, B)$. Let us call this process of identification p.[14]

Without any obvious non-equivalent alternatives for 'Structure of A', let us pursue this choice by forming the term 'the structure of A'. In its full glory it is

$$the(Structure(A), (A, id_A), p) \equiv (A, id_A) : Structure(A).$$

[14] We could also work with an equivalent type: $Structure(A) \equiv \sum_{X:\mathcal{U}}(A = X)$. Then Lemma 3.11.8 of (UFP 2014) gives us contractibility.

Dropping p, we find that 'the structure of A' is (A, id_A). Notice that the component id_A is playing a role here. We should note that, having constructed the type A, were we to construct an element $g : Equiv(A, A)$ which is not equal to id_A, then we could equally use (A, g) to witness the contractibility of $Structure(A)$. This would require a modification to p, but to the extent that this component is not mentioned, we might equally well say that (A, g) is 'the structure of A', or indeed any $(B, f) : Structure(A)$. An element of the type is an entity, B, structured as A as witnessed by an equivalence f. Any such element has trivial identity type with any other, $Id_{Structure(A)}((B_1, f_1), (B_2, f_2))$. This is very much like the case we described earlier of the product of types, where we needed not just a type but also extra information such as its projections. If I do not include the extra information, here the way, f, that some type B is equivalent to A, then there need not be only one way that B shows itself to be structured as A.

Now what does it mean to say that A and B have the same structure? Well one might expect that it means to indicate an identity between two elements 'the structure of A' and 'the structure of B'. As in the case of the morning star and the evening star, naively read they are elements of different types and so not to be directly compared, but like that example we can project to the first component, that is the type that the dependent types are depending upon, here the universe $\mathcal{U}$. Then the identity of the elements amounts to an identity between types A and B in $\mathcal{U}$, or in other words equivalence between the types.

Alternatively, we might define a $\mathcal{U}$-dependent type 'X has the same structure as A' $\equiv Equiv(A, X)$. Then consider by λ-abstraction the term $\lambda X.Equiv(A, X)$, which in words we might say designates 'has the same structure as A'. Now, 'has the same structure as B' is an element of the same type, and we can ask for their identity type. This can easily be shown to be equivalent to $Equiv(A, B)$.

I mentioned another approach to definite description as the result of applying a function, as in 'the captain of team t'. Here we might think there is a function from the type of types, $\mathcal{U}$, to some type of structures, $Structure$, fibred above it. The evident choice for $Structure$ is

$$\sum_{(X:\mathcal{U})} Structure(X) \equiv \sum_{(X,Y:\mathcal{U})} Equiv(X, Y),$$

in which case we have a similar solution to the one above in that 'the structure of' is found to be a function in $\prod_{(X:\mathcal{U})} Structure(X)$, which sends A to $\langle A, (A, id_A)\rangle$.[15]

3.1 Places in a Structure

Some structuralist philosophers of mathematics have referred to 'places' or 'positions' in a structure (Resnik 1997, Shapiro 1997), for instance, to refer to particular natural numbers in the structure that is the natural numbers. This is to indicate elements in what results from a process which abstracts away from different presentations of the 'same structure'. Let us see what it is possible to express within HoTT.

[15] Since the fibres are contractible, $Structure$ is equivalent to $\mathcal{U}$ (UFP 2014, Lemma 3.11.9).

Well, ‘places in the structure of A’ suggests that we form a type which depends on $Structure(A)$. There doesn’t appear to be much choice here other than

$$(X,f) : Structure(A) \vdash PlacesIn(X,f) \equiv X : Type.$$

It would be very natural then to form the dependent product to allow the collection of coherent choices of element of A along with their corresponding elements in each type structured as A, according to the specified equivalence:

$$\prod_{((X,f):Structure(A))} PlacesIn(X,f).$$

We might pronounce this ‘Places in A-structured types’. This type can easily be shown to be equivalent to A, since a choice $a : A$ determines an element $f(a) : B$ for each type-structured-as-A, (B,f), and a choice of place in A-structured types delivers an element of A when applied to (A, id_A).

3.2 Types Equipped with Structure

Of course, we don’t just talk about plain types, but also about monoids, groups, vector spaces, etc. Consider one of the simplest cases, the semigroup structure. This merely requires that there be an associative binary multiplication on the type. Following definition 2.14.1 of UFP (2014),

$$SemigroupStr(A) :\equiv \sum_{(m:A\to A\to A)} \prod_{(x,y,z:A)} m(x,m(y,z)) = m(m(x,y),z).$$

Now a semigroup is a type together with such a structure:

$$Semigroup :\equiv \sum_{(A:\mathcal{U})} SemigroupStr(A).$$

Then for a particular $(A,m,a) : Semigroup$, where a is a proof of the associativity of m, we can define

$$Str(A,m,a) :\equiv \sum_{((X,y,z):Semigroup)} f : Equiv_{Semigroup}((A,m,a),(X,y,z)),$$

where $Equiv_{Semigroup}$ requires of an element that it is an equivalence between underlying types and that it transports the semigroup structure correctly. Once again this results in a contractible type as witnessed by $(A,m,a,id_{(A,m,a)})$, which element we may then call ‘the structure of the semigroup A’. Places in (A,m,a)-structured semigroups will again amount to A.

3.3 The complex numbers

We can put together the constructions treated above to handle the case of the complex numbers. The issue at stake here is whether some forms of structuralism are forced to identify the two square roots of -1 in the field of complex numbers, i and $-i$, given that the non-trivial field automorphism, conjugation, maps them to each other, that we cannot distinguish them in terms of the real-valued polynomials they satisfy, and so on.[16]

How to introduce $\mathbb{C}$ in type theory? Of course, there will be many ways to do so, but we can distinguish two styles of definition, as 'particular' and 'abstract' types.[17] For example, we can form a particular type of complex numbers from the bottom up as ordered pairs of reals, with specified addition and multiplication, etc. These reals in turn will have a particular structure depending on how they have been defined (see Chap. 11 of UFP 2014). On the other hand, as an abstract type we can construct a type to which $\mathbb{C}$ as a whole belongs, for instance, the type, $\mathcal{A}$, of algebraically closed fields of characteristic zero and cardinality of the continuum. A concrete construction of a particular type $\mathbb{C}$ then becomes a proof that $\mathcal{A}$ is inhabited.

These specifications carry different information, the difference being very much like that between the two ways of specifying the cyclic group of order 6 in §2.3. In the case of the particular type, any $z : \mathbb{C}$ may be decomposed into its real and imaginary parts. Now $\langle 0, 1\rangle$ and $\langle 0, -1\rangle$ are two different elements, both of which square to -1. On the other hand, in the case where we assume $\mathbb{C} : \mathcal{A}$, we don't have the means to individuate the two square roots, and yet the subtype of elements squaring to -1 is of cardinality 2. There are two such places in the structure. $\mathcal{A}$ is equivalent to $\mathbf{B}Aut(\mathbb{C}) \equiv \mathbf{B}S_2$ (recall this notation from §2.4) with a non-trivial structured auto-equivalence which exchanges the two square roots. We might consider the strict condition for the use of 'the' in 'the complex numbers' to require us to break the symmetry by specifying one root as i, although in practice mathematicians will often say '$\mathbb{C}$ is the algebraically closed field of characteristic zero and cardinality the continuum'. In any case, the situation is very much like the cyclic group of order 6 or the two identical balls, and presents no difficulty to a type theoretic viewpoint.

4 Conclusion

One conclusion to draw from this note is that from the perspective of HoTT, little is gained by explicit use of the word 'structure' in the sense of 'the structure of A.' Types and structured types in HoTT just *are* structures that do not need to be abstracted from an underlying set-like entity. HoTT simply is a "synthetic theory of structures" (Shulman, *forthcoming*). The proper treatment of structure comes along for free and need not be explicitly mentioned. In this sense then HoTT should be viewed very favourably by stucturalists.

If this counts as a result which closes off a certain line of enquiry, on the way to it we have seen something more positive.

1. The analysis of the word *the* in terms of its introduction rule shows that HoTT has something to teach us about the classic philosophical topic of definite descriptions. We have seen that it provides a rationale for mathematicians' use of a generalized 'the' in situations where it appears that they might be referring to more than one entity.

[16] See, e.g., Keränen (2001, 2006) and other contributions to MacBride (2006), and Nodelman and Zalta (2014).

[17] I am indebted to Mike Shulman's discussion 'From Set Theory to Type Theory' (2013).

2. More generally, we were able to make useful sense of several issues concerning type and identity. HoTT promises to be an important tool for philosophers of language and metaphysicians.

3. Our analysis of ‘the’ employed a principle that may prove of lasting importance:

    > (*Treat all types evenly*) Any time we have a construction which traditionally has been taken to apply only to sets or only to propositions, then since in HoTT these form just a certain kind of type, we should look to see whether the construction makes sense for all types.

    Further examples are not hard to find. If we generally take modal operators, such as ‘it is necessarily the case that...’, to apply only to propositions, we should look to see whether there is anything to prevent a more general construction applying to all types.

It is surely intriguing that a newly proposed foundations for mathematics displays the potential to speak to issues within philosophy in general, and is not confined to the domain of philosophy of mathematics.

Acknowledgements: I would like to thank Mike Shulman and Urs Schreiber for their very helpful advice, and three anonymous referees for their suggestions.

5 Bibliography

Awodey, S. 2014. ‘Structuralism, Invariance, and Univalence’, Philosophia Mathematica 22(1): 1-11, (pdf)

Benacerraf, P. 1965. ‘What Numbers Could Not Be’, in Benacerraf, P. & Putnam, H. (eds.), 1983. *Philosophy of Mathematics: Selected Readings*, Cambridge: Cambridge University Press, 2nd edition, 272–294.

Black, M. 1952. ‘The Identity of Indiscernibles’, *Mind* 61(242): 153–164.

Collingwood, R. 1940. *An Essay on Metaphysics*, Oxford: Clarendon.

Conrad, B. *n.d.* ‘Math 121. Uniqueness of algebraic closure’, handout

Corfield, D. 2003. *Towards a philosophy of real mathematics*, Cambridge: Cambridge University Press.

Henriques, A. 2010. Comment on ‘What’s a groupoid? What’s a good example of a groupoid?’ (http://mathoverflow.net/a/1161)

Keränen, J. 2001. ‘The Identity Problem for Realist Structuralism’. *Philosophia Mathematica* 9, pp. 308–30.

Keränen, J. 2006. ‘The Identity Problem for Realist Structuralism II: A Reply to Shapiro’. In MacBride (ed.) 2006, pp. 146–63.

Lowe, E.J. 1989. ‘Impredicative identity criteria and Davidson’s criterion of event identity’, Analysis 49(4): pp. 178-181.

MacBride, F. (ed.) 2006. *Identity and Modality.* Oxford: Clarendon Press.

nLab, 'general covariance', `https://ncatlab.org/nlab/show/general+covariance`

Nodelman, U. and Zalta, E. 2014. 'Foundations for Mathematical Structuralism', *Mind* 123/489: 39–78.

Ranta, R. 1994. *Type-theoretic grammar*, Oxford: Clarendon Press.

Resnik, M., 1997. *Mathematics as a Science of Patterns*, Oxford: Clarendon Press.

Russell, B., 1905. 'On Denoting', *Mind* 14: 479–493.

Schreiber, U. 2014. 'Quantization via linear homotopy types', arXiv:1402.7041

Schreiber, U. 2016. 'Higher prequantum geometry', arXiv:1601.05956

Shapiro, S. 1997. *Philosophy of Mathematics: Structure and Ontology*, Oxford: Oxford University Press.

Shulman, M. 2013. 'From set theory to type theory', online article, `https://golem.ph.utexas.edu/category/2013/01/from_set_theory_to_type_theory.html`

Shulman, M. 2017. 'Homotopy type theory: the logic of space', arXiv:1703.03007

Shulman, M. *forthcoming.* 'Homotopy Type Theory: A synthetic approach to higher equalities', in E. Landry (ed.) *Categories for the working philosopher*, Oxford University Press, (pdf).

Strawson, P.F., 1950. 'On Referring', *Mind* 59: 320–344.

Strawson, P.F. 1964. 'Identifying Reference and Truth-Values', *Theoria* Vol XXX, reprinted in Strawson P. F. 1971. *Logico-Linguistic Papers*, London: Methuen.

Sundholm, G. 1986. 'Proof theory and meaning'. In D. Gabbay & F. Guenthner (Eds.), *Handbook of Philosophical Logic, Vol. III*, pp. 471–506, Reidel: Kluwer.

UFP (Univalent Foundations Program) 2014, *Homotopy Type Theory: Univalent Foundations of Mathematics.*, `http://homotopytypetheory.org/book/`

Vendler, Z. 1967a. Lingustics in Philosophy. Cornell University Press.

Vendler, Z. 1967b. 'Causal Relations', The Journal of Philosophy 64(21), pp. 704-713.