

# The Generalised Algebraic Theory of Contextual Categories

John Cartmell

January 10, 2018

## 1 Introduction

The syntactical notion of *generalised algebraic theory* (GAT) and the equivalent algebraic notion of *contextual category* were introduced in my thesis [1] and published in [2]. Whereas generally the objects of categories can be taken to represent types, the objects of a contextual categories can be taken to represent what I have described as ‘types that vary’ but which subsequently have become known as dependent types.

I remarked in [1] that the theory of contextual categories can be expressed as a generalised algebraic theory by the axiomatisation of identity types on the morphism types of the category (the identity types are required in order to phrase the pullback structure required by the definition of contextual category). The remark is made in order to point out that contextual categories are a generalised algebraic equivalent of GATs and that, in this sense, the machinery of GATs is self-descriptive. The same thing cannot be said of Lawvere algebraic theories nor of many-sorted algebraic theories.

At the time I made that remark I would have much preferred to find a way of theorising contextual categories by means of a GAT that didn’t have additional types (the identity types); their introduction seemed to me not to be in the spirit of algebra. Subsequently Voevodsky in [3] has shown that identity types are not needed; he introduces an operator ‘s’ which canonically maps any morphism to a representative section; he shows that suitably axiomatised this implies the existence of pullbacks. The purpose of this paper is to document the generalised algebraic theory of contextual categories which results from following this approach – the GAT of contextual categories is expressed without introducing types to represent identity of morphisms; its sorts are just those that represent objects and morphisms of the category.

Note that Voevodsky took exception to my use of the name *contextual category* and used the name *C-system* in its place. This is not a material difference; it is a difference of terminology.

To give readability to the theory we introduce a schematic notation for representing chains of dependent variables. As a prelude to presenting the theory we first present two simpler but related theories (previously presented in [1] and [2]): the GAT of categories and the GAT of trees.

## 2 Background Theories and Notations

### 2.1 The GAT of Categories

*Symbol*    *Introductory Rule*

*Ob*        *Ob* is a type

*Hom*        $x, y \in Ob \vdash Hom(x, y)$  is a type

$\circ$           $x, y, z \in Ob, f \in Hom(x, y), g \in Hom(y, z) \vdash \circ(f, g) \in Hom(x, z)$

*id*         $x \in Ob \vdash id(x) \in Hom(x, x)$

*Axioms*

$\circ(id(x), f) = f$ , whenever  $x, y \in Ob, f \in Hom(x, y)$

$\circ(f, id(y)) = f$ , whenever  $x, y \in Ob, f \in Hom(x, y)$

$\circ(\circ(f, g), h) = \circ(f, \circ(g, h))$ , whenever  $w, x, y, z \in Ob, f \in Hom(w, x), g \in Hom(x, y), h \in Hom(y, z)$

## 2.2 Trees of Concepts and the GAT of Trees

A contextual category has a tree of objects and we think of this tree as a tree of dependencies among concepts.

Formally, a tree is any partially ordered set  $(S, <)$  such that for each  $t \in S$ , the set  $\{s \in T : s < t\}$  is well-ordered by the relation  $<$ . We wish to consider just trees  $(S, <)$  such that for each  $t \in S$ , the set  $\{s \in S : s < t\}$  is finite and such that there is a unique root to the tree i.e. a unique least element. We call the elements of set  $S$  the nodes of the tree and for each node the cardinality of the set  $\{s \in S : s < t\}$  is said to be the height of the node  $t$ . We denote by  $S_i, i \geq 0$ , the set of elements of  $S$  of height  $i$ . The set  $S_0$  is a singleton set containing the root of the tree.

Such trees as these we can equivalently describe as models of the generalised algebraic theory given below table 1 in which the nodes of height  $n + 1$  are represented as of a sort  $S_{n+1}$  that is dependent on the sort of nodes of height  $n$ .

If  $A$  and  $B$  are nodes of a tree  $(S, <)$  then we shall write  $A \triangleleft B$  to mean that  $A < B$  in  $S$  and that there does not exist  $x$  such that  $A < x < B$ . For every node  $B$  of tree  $S$  other than the root node there exists a unique node  $A$  such that  $A \triangleleft B$ .

Table 1: The Generalised Algebraic Theory of Trees

<i>Symbol</i>	<i>Introductory Rule</i>
$S_0$	$S_0$ is a type
$S_1$	$x_0 \in S_0 \vdash S_1(x_0)$ is a type
$S_2$	$x_0 \in S_0, x_1 \in S_1(x_0) \vdash S_2(x_0, x_1)$ is a type
$\vdots$	
$S_n$	$x_0 \in S_0, x_1 \in S_1(x_0), \dots, x_{n-1} \in S_{n-1}(x_0, x_1, \dots, x_{n-2}) \vdash S_n(x_0, x_1, \dots, x_{n-1})$ is a type
$\vdots$	
<i>root</i>	<i>root</i> $\in S_0$
<i>Axioms:</i>	
	$x, y \in S_0 \vdash x = y$

## 2.3 Schematic Notation

There is a shorthand that is convenient in the presentation of the GAT of trees and then subsequently in the GAT of contextual categories. We use the shorthand  $x \in S$  for the context  $x_0 \in S_0, x_1 \in S_1(x_0), \dots, x_n \in S_n(x_0, x_1, \dots, x_{n-1})$ .

Using this shorthand, for any  $n \geq 0$  the sort  $S_n$  in the theory of trees is introduced as follows:

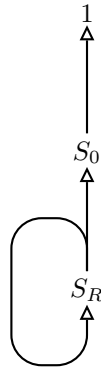
<i>Symbol</i>	<i>Introductory Rule</i>
$S_0$	$S_0$ is a type
$S_{n+1}, n \geq 0$	$x \in S \vdash S_{n+1}(x)$ is a type

## 2.4 An Aside on Recursive Type Definitions

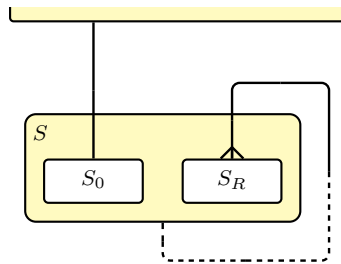
Using the shorthand, we are quite close to having a recursive definition of a single sort  $S_R$ . Such definitions are not possible in generalised algebraic theories but we can imagine a framework in which it is possible to write:

<i>Symbol</i>	<i>Introductory Rule</i>
$S_0$	$S_0$ is a type
$S_R$	$x_0 \in S_0 \vdash S_R(x_0)$ is a type
$S_R$	$x \in S_R \vdash S_R(x)$ is a type

Such a definition could be represented algebraically in a suitably generalised notion of contextual category (a comulti-contextual category?) these dependencies could be represented as follows:



This is not just an idle thought – in data modelling such a tree structure is represented in an entity model diagram in which the injections into the coproduct  $S$  of  $S_0$  and  $S_R$  are represented by containment:

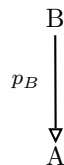


See [www.entitymodelling.org/tutorialone](http://www.entitymodelling.org/tutorialone) for a description of this notation. For an example of the modelling of recursive relationships in the definition of a phrase structure grammar of English see [www.entitymodelling.org/examplesone/englishsentence](http://www.entitymodelling.org/examplesone/englishsentence).

### 3 Definition of Contextual Categories

#### 3.1 The GAT of Tree-Structured Categories

By a tree-structured category we mean (i) a category with a tree-structure defined on its objects such that the tree of objects has a unique root object and (ii) for every  $A \triangleleft B$  in the tree of objects a canonical morphism  $p_B : B \rightarrow A$  this morphism will be distinguished in diagrams by an arrow with a triangular head so:



The theory of tree-structured categories can be presented as a generalised algebraic theory as follows:

<i>Symbol</i>	<i>Introductory Rule</i>
$Ob_0$	$Ob_0$ is a type
$Ob_{n+1}$	$x \in Ob \vdash Ob_{n+1}(x)$ is a type
$Hom_{n,m}$	$x \in Ob, y \in Ob \vdash Hom_{n,m}(x, y)$ is a type
$\circ$	$x \in Ob, y \in Ob, z \in Ob, f \in Hom_{n,m}(x, y), g \in Hom_{m,p}(y, z) \vdash \circ(f, g) \in Hom_{n,p}(x, z)$
$id_n$	$x \in Ob \vdash id_n(x) \in Hom_{n,n}(x, x)$
$p_n$	$x \in Ob \vdash p_n(x) \in Hom_{n,n-1}(x, x_{n-1})$
$1$	$1 \in Ob_0$
$t_n$	$x \in Ob \vdash t_n(x) \in Hom_{n,0}(x, 1)$

#### *Axioms*

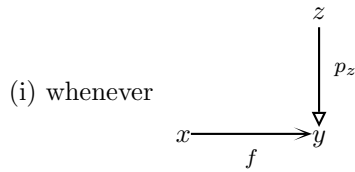
- $\circ(id(x), f) = f$ , whenever  $x \in Ob, y \in Ob, f \in Hom_{n,m}(x, y)$
- $\circ(f, id(y)) = f$ , whenever  $x \in Ob, y \in Ob, f \in Hom_{n,m}(x, y)$
- $\circ(\circ(f, g), h) = \circ(f, \circ(g, h))$ , whenever
  - $w \in Ob, x \in Ob, y \in Ob, z \in Ob, f \in Hom_{l,n}(w, x), g \in Hom_{n,m}(x, y), h \in Hom_{m,p}(y, z)$
- $x = y$ , whenever  $x, y \in Ob_0$
- $f = t_n(x)$ , whenever  $x \in Ob, f \in Hom_{n,0}(x, 1)$

Writing informally, we omit the numeric subscripts and we also write  $p_x$  for  $p(x)$  and  $t_x$  for  $t(x)$ .

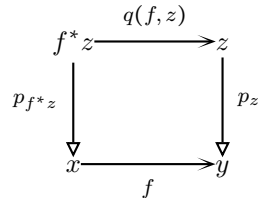
## 3.2 Contextual Categories

### 3.2.1 Original Definition

As defined in [1] and [2], a contextual category is defined to be a tree-structured category  $\mathbf{C}$  with the following additional structure:

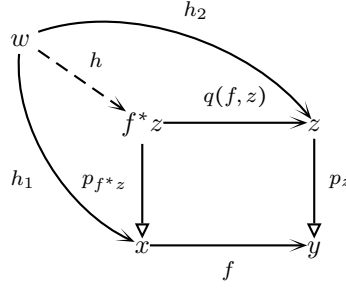


in  $\mathbf{C}$ , an object  $f^*z$  such that  $x \triangleleft f^*z$ , a morphism  $q(f, z) : f^*z \rightarrow z$  such that the diagram:



commutes,

(ii) and such that each such diagram is a pullback diagram, that is: for all objects  $w$  of  $\mathbf{C}$ , and for all morphisms  $h_1 : w \rightarrow x$  and  $h_2 : w \rightarrow z$  (see diagram below) such that  $h_1 \circ f = h_2 \circ p_z$  there exists a unique  $h : w \rightarrow f^*z$  in  $\mathbf{C}$  such that  $h \circ p_{f^*z} = h_1$  and  $h \circ q(f, z) = h_2$ .



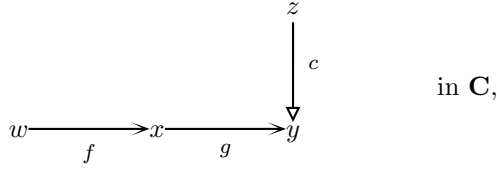
(iii) whenever  $x \triangleleft y$  in  $\mathbf{C}$ ,

$$id_x^* y = y \tag{1}$$

and

$$q(id_x, y) = id_y \tag{2}$$

(iv) whenever



in  $\mathbf{C}$ ,

then

$$(f \circ g)^* z = f^*(g^* z) \tag{3}$$

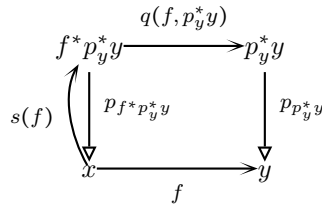
and

$$q(f \circ g, z) = q(f, g^* z) \circ q(g, z) \tag{4}$$

### 3.2.2 Equational Definition following Voevodsky

Following Voevodsky, however, we may replace the pullback condition, (ii), by additional structure (ii') as follows:

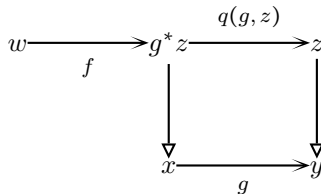
(ii') for all morphisms  $f : x \rightarrow y$  a morphism  $s(f) : x \rightarrow f^* p_y^* y$  as shown here:



such that

$$s(f) \circ p_{f^* p_y^* y} = id_x \tag{5}$$

and such that whenever



in  $\mathbf{C}$  then

$$s(f \circ q(g, z)) = s(f) \tag{6}$$

which is well-typed because

$$\begin{aligned} lhs &= s(f \circ q(g, z)) \in Hom(w, (f \circ q(g, z))^* p_z^* z) \\ rhs &= s(f) \in Hom(w, (f \circ p_{g^* z})^* g^* z) \end{aligned}$$

and

$$Hom(w, (f \circ q(g, z))^* p_z^* z) = Hom(w, (f \circ p_{g^* z})^* g^* z)$$

because

$$\begin{aligned} (f \circ q(g, z))^* p_z^* z &= f^*((q(g, z) \circ p_z)^* z) \\ &= f^*(p_{g^* z} \circ g)^* z \\ &= (f \circ p_{g^* z})^* g^* z \end{aligned}$$

### 3.3 GAT of Contextual Categories

Following Voevosky's insight, the GAT of Contextual Categories is defined to be the GAT of Tree-Structured Categories plus the following additional structure:

*Symbol*    *Introductory Rule*

$$\begin{array}{l} * \quad x \in Ob_n, y \in Ob_m, f \in Hom_{n,m}(x, y), z \in Ob_m(y) \vdash f^* z \in Ob_n(x) \\ q \quad x \in Ob_n, y \in Ob_m, f \in Hom_{n,m}(x, y), z \in Ob_m(y) \vdash q(f, z) \in Hom_{n,m+1}(f^* y_m, y_m) \\ s \quad x \in Ob_n, y \in Ob_m, f \in Hom_{n,m}(x, y) \vdash s(f) \in Hom_{n,n+1}(x, f^* p(y)^* y) \end{array}$$

*Axioms*

$$\begin{aligned} \circ(q(f, z), p(z)) &= \circ(p(f^* z), f), \text{ whenever } x \in Ob_n, y \in Ob_m, f \in Hom_{n,m}(x, y), z \in Ob_m(y) \\ (f \circ g)^* z &= f^*(g^* z), \text{ whenever } w \in Ob_n, x \in Ob_m, y \in Ob_p, f \in Hom_{n,m}(w, x), g \in Hom_{m,p}(x, y) \\ q(f \circ g, z) &= q(f, g^* z) \circ q(g, z), \text{ whenever } w \in Ob_n, x \in Ob_m, y \in Ob_p, f \in Hom_{n,m}(w, x), g \in Hom_{m,p}(x, y) \\ \circ(s(f), p(f^* p(y)^* y)) &= id(x), \text{ whenever } x \in Ob_n, y \in Ob_m, f \in Hom_{n,m}(x, y) \\ s(f \circ q(g, z)) &= s(f), \text{ whenever } w \in Ob_n, x \in Ob_m, y \in Ob_p, g \in Hom_{m,p}(x, y), z \in Ob_{p+1}(y), f \in Hom_{n,m+1}(w, g^* z) \end{aligned}$$

## References

- [1] John Cartmell. *Generalised algebraic theories and contextual categories*. PhD thesis, University of Oxford, 1978.
- [2] John Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32(0):209 – 243, 1986.
- [3] Vladimir Voevodsky. Subsystems and regular quotients of c-systems, 2014, arXiv:1406.7413.