Imperial College of Science,
Technology and Medicine

(University of London)

Department of Computing

Reasoning in Arithmetic Universes

by

Alan Morrison

Submitted in partial fulfilment
of the requirements for the MSc
Degree in Engineering of the
University of London and for the
Diploma of Imperial College of
Science, Technology and Medicine

September 1996

Abstract

An *Arithmetic Universe* is a category in which coherent logic can be interpreted and in which recursive definitions can be made. The notion of Arithmetic Universe was introduced by André Joyal in a lecture he gave in 1973 in which he presented a categorical approach to proving Gödel's incompleteness theorem. He constructed a minimal category which had the qualities which we would like to ascribe to an Arithmetic Universe. Regrettably, none of his work in this area has ever been published.

Recently, interest in Arithmetic Universes has awakened in the theoretical computer science world as a possible place in which to perform categorical recursion theory and in which specification languages may be interpreted. It has not so far been entirely clear what the minimal set of axioms for defining an Arithmetic Universe should be. In this paper, we give precise definitions of the concepts involved and then re-examine Joyal's construction and give detailed proofs of his results. We then formulate a definition of an Arithmetic Universe as a pre-topos with action variants and we conjecture that this is the minimal set of suitable axioms.

Finally, we present part of a proof of a conjecture of Steve Vickers that Arithmetic Universes have freely generated Lex Theories.

Acknowledgements

I am grateful to Steve Vickers whose idea this thesis was and who has proved himself to be a thoroughly good egg in all sorts of ways. He has been extremely helpful throughout the life of the project and in particular during the final month when the calls upon his time have grown ever larger and more ridiculous. In addition to assistance with specific thesis-related matters he has spent time talking to me about the more general categorical issues underlying the area and has certainly provided me with a great deal of intuition.

I must also record my gratitude to Gavin Wraith for making his notes on this subject available to me and for kindly spending a morning with me explaining them.

All of the commuting diagrams were drawn using Paul Taylor's commuting diagrams package.

Contents

1 Introduction **1**

2 Arithmetic Categories **6**

2.1 Parameterised Recursion 6

2.2 List Arithmetic Categories 12

2.3 Natural Numbers Objects 18

2.4 Inductive Proofs in Arithmetic Categories 20

3 Internal Categories **23**

4 Distributivity **27**

5 Skolem Theories and A.U. Construction **31**

5.1 The Language PRIM 31

5.2 Constructing A.U.'s from Skolem Theories 43

5.2.1 Adding Decidable Subsets to E 43

5.2.2 The Exact Completion of $\check{E}$ 68

6 Defining the Arithmetic Universe **83**

7 Theories in Arithmetic Universes **88**

7.1 Theories 88

7.2 Internal Categories in A.U.'s 96

8 Conclusions **102**

Chapter 1

Introduction

In 1973 André Joyal gave a lecture at a conference in Amiens entitled "Theorem d'Incomplétitude de Gödel et Univers Arithmétique" in which he outlined a categorical approach to the incompleteness results of Gödel. His approach involved the construction of a minimal category in which recursive definitions could be performed which contained a model of itself. He called the resultant category the "Initial Arithmetic Universe", $\mathcal{A}_0$. Regrettably, none of this work was published, although he produced some lecture notes which are still in circulation.

Recursion theory is clearly of interest to Computer Scientists and Joyal's construction produced a category which also admitted the interpretation of observational *coherent* theories. The combination of these two properties therefore seems likely to be very attractive to theoretical computer scientists. For example, a recent proposal by Steve Vickers has suggested that Arithmetic Universes could form the basis for the interpretation of a specification language in which the user would be forced to employ only coherent logic. The related work of Robin Cockett ([Co 90]) , Barry Jay ([Jay 93]) and others has looked at the applications of recursion theory in arbitary distributive categories to problems related to programming semantics. It seems likely that Arithemetic Universes could form the basis of an entirely categorical exposition of recursive function theory.

The lack of published material on the subject has so far acted as a drag upon research work in the area. In particular, there has been some debate about precisely what should be the definition of an Arithmetic Universe.

This thesis attempts to address some of these questions. There is no attempt herein to replicate or to expand upon Joyal's proof of Gödel's result - we are more concerned here with a sufficient framework for other more general applications in recursion theory and specification langauges as mentioned above.

The work of Makkai and Reyes ([MakRey 77]) has shown that pre-toposes are precisely those categories in which coherent logic can be interpreted. It therefore seems that an Arithmetic Universe should be a pre-topos with some additional properties. To get an understanding of precisely what those properties should be, we can turn to Joyal's original construction and see which properties $\mathcal{A}_0$ has. Specifically, $\mathcal{A}_0$ has all free monoids, all list objects and also free category objects over arbitary graph objects. In $\mathcal{A}_0$ we can also lift an arbitary left-action over an internal graph to a unique left-action over the category of paths derived from that graph which agrees with the graph action on the generators - the so-called *action variant*. The starting point for this investigation was the suggestion that these definitions were all equivalent.

To investigate this claim properly, it is neccessary to perform some background work which will allow us to define precisely each of these intuitive concepts in entirely categorical terms. Chapters 2,3 and 4 of this document survey the relevant literature and prove some elementary relations between the above definitions.

Chapter 2 is devoted to the definition of recursion in categories; the definitions have been extracted from Robin Cockett's paper ([Co 90]). He defines *parameterised* recursion, which is going to be of interest to us. He then identifies two flavours of parameterised recursion and associates them with *number-arithmetic* and *list-arithmetic* categories. We demonstrate that list-arithmetic categories have free monoids. We then turn to number arithmetic categories and examine natural numbers objects and the primitive recursive definition of functions in arbitary cartesian categories. Finally we justify the use of familiar inductive techniques in arithmetic categories.

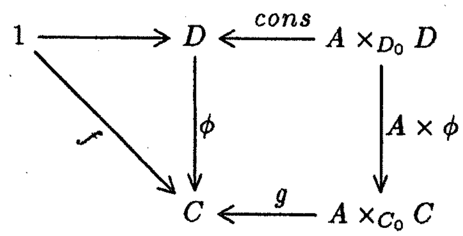
Chapter 3 is concerned with the definitions of internal graphs and of internal categories in a category and is therefore related to the notion of a category as a "universe of mathematical discourse". We define $Cat(\mathcal{C})$, the category of categories in $\mathcal{C}$, and motivate and discuss the definition of internal diagrams in $\mathcal{C}$.

Chapter 4 is a brief review of the relevant parts of Carboni, Lack and Walter's paper ([CLW 93]) about distributive categories. The upshot of this chapter is that any pre-topos is necessarily distributive and hence that Arithmetic Universes are distributive.

In chapter 5 we examine Joyal's construction in some detail. The key notion here is that of a *Skolem Theory* - a cartesian category whose only objects are powers of N, where N is a natural numbers object as defined in chapter 2. We want to define arrows in an arbitary Skolem Theory E using primitive recursion so we start by examining in some detail a language PRIM which appears in Gavin Wraith's notes ([GCW Notes]). It is easy to show that PRIM functions correspond precisely to primitive recursive functions. We define in some detail an interpretation function in E for PRIM functions - in other words, a denotational semantics. We want to know that two functions which are the same when interpreted in **Set** are the same when interpreted in E. In order to express this requirement precisely we define an operational semantics for E in the familiar fashion and we then conjecture without proof a correspondence result which would justify our use of PRIM to define functions throughout the remainder of the paper.

We then examine Joyal's Skolem Theory completion process; this generates a pre-topos with all of the above properties and the completion of the initial Skolem Theory is his category $\mathcal{A}_0$. Most of the material available to me on this topic was rather sketchy and I have therefore performed the majority of the proofs myself. I have however been lucky to have access to Gavin Wraith's unpublished notes on the subject ([GCW Notes]). In particular, Wraith supplied the kernel of the proofs of the existence of free monoids in E and of the existence of finite limits, coproducts and split epi/monic factorisations in $\check{E}$.

In forming the proofs of the existence of free categories over graphs and of action variants, I managed to convince myself of the non-equivalence of the above definitions. When attempting to demonstrate the existence of free categories over graph objects in a category with list objects, one ends up attempting to define functions ϕ using recursive definitions of the following form:



Given the existence only of list objects, we can in general only make definitions of this nature when $C_0 = D_0 = 1$. The definition is possible in the completion of a Skolem Theory only because it can be made using standard list recursion in a category which is lower down in the completion hierarchy and then demonstrated to work in the completed category.

In view of results like this, we postulate a number of non-equivalences between the earlier definitions and we list these with some proved equivalences in chapter 6. We then define an Arithmetic Universe to be a pre-topos which has action variants - we show that this will certainly give us all of the desired properties and conjecture that a weaker definition would not. It has been suggested that an appropriate definition of Arithmetic Universe would be any category which has been obtained from a Skolem Theory by Joyal's completion - we observe that this would preclude the use of **Set** as an A.U. and therefore reject this definition.

Chapter 7 is devoted to a partial proof of a conjecture of Steve Vickers that Arithmetic Universes have freely-generated finitely presented Lex Theories. By this we mean that given freely generated Lex Theories $\mathcal{T}_1$ and $\mathcal{T}_2$ and a Lex functor $\mathcal{T}_1 \longrightarrow \mathcal{T}_2$ the induced pullback functor between models of $\mathcal{T}_2$ and $\mathcal{T}_1$ in a given Arithmetic Universe $\mathcal{A}$ always has a left adjoint. To render this statement precise we need to define Lex Theories; we do this using Sketches. This approach was originally due to Ehresmann, but our approach is due to Barr and Wells ([BW 85]). Using sketches allows us to separate the underlying graph of the theory $\mathcal{T}$ from the limits and commuting diagrams which it includes. Clearly, the graph of $\mathcal{T}$ itself generates a Lex theory $\mathcal{T}'$. We then review some of the material in [John 77] before suggesting an approach to proving the theorem.

Our approach mirrors that of Barr and Wells for the same problem in **Set**. They use the existence of Left Kan extensions of functors into **Set** to show that the theorem is true for models of $\mathcal{T}_1'$ and of $\mathcal{T}_2'$. They then show that the category of models of $\mathcal{T}$ in **Set** is a reflective subcategory of the category of models of $\mathcal{T}'$ in **Set** and hence deduce their result.

We cannot assume the existence of left Kan extensions as our AU $\mathcal{A}$ may not be cocomplete. However, Johnstone ([John 77]) shows that left Kan extensions of *internal* functors will always exist in $\mathcal{A}$. We prove our result by demonstrating that categories freely generated over finite graphs can always be represented in $\mathcal{A}$ and then that functors from these categories to $\mathcal{A}$ correspond precisely to internal diagrams over the representation of these categories. Our proof uses the existence of action variants to allow us to rely upon the *internal* properties of the Arithmetic Universe.

We leave unproved the conjecture that if $\mathcal{T}$ is finitely presented then the category of models of $\mathcal{T}$ in $\mathcal{A}$ is a reflective subcategory of the category of models of $\mathcal{T}'$ in $\mathcal{A}$, although we give some rationale for believing the statement to be true.

Chapter 2

Arithmetic Categories

2.1 Parameterised Recursion

We start with a brief overview of the definitions which relate to monoids in a category.

Definition 2.1 (Cartesian Categories) *A* cartesian category *is one which has all finite products.*

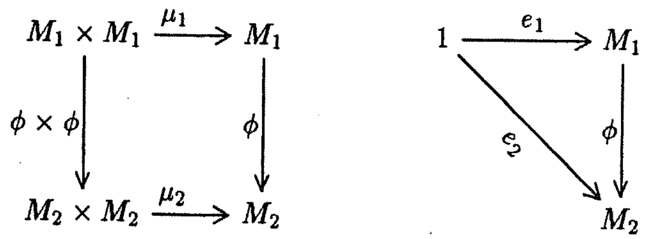
In the following, we will deal with a cartesian category C. The definitions which relate to monoids could equally as well be made in a monoidal category.

Definition 2.2 (Monoid Objects) *If C is a cartesian category and $M \in Ob(C)$ then* M is a monoid in C *if there is a map $e : 1 \longrightarrow M$ and a map $\mu : M \times M \longrightarrow M$ such that the following diagrams commute:*

$$\begin{array}{ccc} M \times M \times M & \xrightarrow{\mu \times id_M} & M \times M \\ {\scriptstyle id_\mu \times \mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ M \times M & \xrightarrow{\mu} & M \end{array} \qquad \begin{array}{ccc} M & \xrightarrow{e \times id_M} & M \times M \\ {\scriptstyle id_M \times e}\downarrow & \searrow^{id_M} & \downarrow{\scriptstyle \mu} \\ M \times M & \xrightarrow{\mu} & M \end{array}$$

where we exclude the obvious isomorphisms.

Definition 2.3 *If M_1 and M_2 are monoids in C then a* monoid homomorphism $\phi : M_1 \longrightarrow M_2$ *in C is a C-arrow which makes the following commute:*



Note that when C is **Set**, the above definitions reduce to the familiar equational ones for a monoid. The following is a direct analogue of the associated definition in **Set**:

Definition 2.4 *If C is a cartesian category and $S \in Ob(C)$ then if it exists, a* free monoid over S *is a monoid object $M(S)$ in C with a C-arrow $\eta : S \longrightarrow M(S)$ such that if P is another monoid object in C with $f : S \longrightarrow P$ a C-arrow then there is a unique monoid homomorphism $\phi : M(S) \longrightarrow P$ with $\phi \circ \eta = f$ in C.*

Lemma 2.5 (Uniqueness of Free Monoids) *Free monoids over an object S of C are unique up to isomorphism.*

Proof Trivial. □

It is easy to prove in the **Set** case that the free monoid over a set S is S^*, the set of finite lists over S with unit the empty list ε and $\mu : S^* \times S^* \longrightarrow S^* : \langle [s_1 \dots s_n], [t_1 \dots t_m] \rangle \mapsto [s_1 \dots s_n t_1 \dots t_m]$ given by concatenation. We will investigate the connection between lists and free monoids in our more general setting.

The following results appear in [Co 90]:

Definition 2.6 *If A, X are C-objects then an* action *of A on X is an arrow $A \times X \longrightarrow X$.*

Definition 2.7 *If C is a cartesian category then $Act_A(C)$ is the category of actions of A in C with morphisms $(A \times X_1 \xrightarrow{x_1} X_1) \longrightarrow (A \times X_2 \xrightarrow{x_2} X_2)$*

given by arrows $f : X_1 \longrightarrow X_2$ which make the following square commute:

$$\begin{array}{ccc} A \times X_1 & \xrightarrow{\;x_1\;} & X_1 \\ {\scriptstyle id_A \times f}\Big\downarrow & & \Big\downarrow{\scriptstyle f} \\ A \times X_2 & \xrightarrow{\;x_2\;} & X_2 \end{array}$$

Definition 2.8 *$U_A : Act_A(\mathcal{C}) \longrightarrow \mathcal{C}$ is the forgetful functor which maps $(x_1 : A \times X_1 \longrightarrow X_1) \mapsto X_1$.*

Definition 2.9 *$\mathcal{C}$ is* recursive *(or $\mathcal{C}$* has enough recursive objects*) if for every $A \in Ob(\mathcal{C})$, U_A has a left adjoint F_A.*

We can express this in terms of the unit of the adjunction; this means that if $X \in Ob(\mathcal{C})$ and $A \in Ob(\mathcal{C})$ then there is an A-action

$$r_1^{A,X} : A \times rec(A,X) \longrightarrow rec(A,X)$$

and a map

$$r_0^{A,X} : X \longrightarrow U_A(r_1^{A,X})$$

with the universal property that if $g : A \times C \longrightarrow C$ is any object of $Act_A(\mathcal{C})$ with $f : X \longrightarrow U_A(h)$ then there is a unique A-action morphism

$$rec(g,f) : U_A(r_1^{A,X}) \longrightarrow U_A(g)$$

such that $rec(g,f) \circ r_0^{A,X} = f$.

In other words, for any $A, X \in Ob(\mathcal{C})$ there is an object $rec(A,X)$ and arrows $r_1^{A,X}$ and $r_0^{A,X}$ such that given any $f : X \longrightarrow C$ and $g : A \times C \longrightarrow C$ there exists a unique $rec(g,f) : rec(A,X) \longrightarrow C$ such that the following commutes:

$$\begin{array}{ccccc} X & \xrightarrow{\;r_0^{A,X}\;} & rec(A,X) & \xleftarrow{\;r_1^{A,X}\;} & A \times rec(A,X) \\ & {\scriptstyle f}\searrow & \Big\downarrow{\scriptstyle rec(g,f)} & & \Big\downarrow{\scriptstyle id_A \times rec(g,f)} \\ & & C & \xleftarrow{\;\;g\;\;} & A \times C \end{array}$$

Cockett observes that there is no reason at first to assume that $rec(A,B)$ should be remotely well-behaved and it may not be; we do not as yet have the ability to perform the *parameterised* recursion from which primitive arithmetic arises. We now state necessary conditions for parameterised recursion to be possible. As usual, if C has exponentiation then parameterisation is automatic, so we will worry about it only when exponentiation is not assumed.

We can obtain parameterised recursion by requiring that recursive objects in slice categories are closely related to those in the underlying category.

Definition 2.10 *Let C be a cartesian category, $Y \in Ob(C)$. The* overlying functor *is defined by* $V_Y : C/Y \longrightarrow C : [X \xrightarrow{F} Y] \mapsto X$. *The functor* I_Y *is defined by* $I_Y : C \longrightarrow C/Y : X \mapsto [X \times Y \xrightarrow{\pi_2} Y]$.

Lemma 2.11 $V_Y \dashv I_Y$

Proof Let $[X_0 \xrightarrow{F} Y] \in Ob(C/Y)$ and suppose that $X_1 \in Ob(C)$.
Define

$$\theta : Hom_C(V_F(Y), X_1) \longrightarrow Hom_{C/Y}(F, I_Y(X_1))$$

by

$$(X_0 \xrightarrow{f} X_1) \mapsto \langle f, F\rangle : (X_0 \xrightarrow{F} Y) \longrightarrow (X_1 \times Y \xrightarrow{\pi_2} Y)$$

Since $\pi_2 \circ \langle f, F\rangle = F$, θ is well-defined.

θ is trivially injective. If $\pi_2 \circ \langle \alpha, \beta\rangle = F$ then $\langle \alpha, \beta\rangle = \theta\alpha$ so θ is surjective, too.

Also, if

$$(X_0' \xrightarrow{F} Y) \quad \text{and} \quad F \xrightarrow{g} F'$$

in C/Y then for any $f : V_Y(F') \longrightarrow X_1$,

$$\begin{aligned}
\mathrm{Hom}_{C/Y}(g, X_1)(\theta(f)) &= \langle f, F'\rangle \circ g \\
&= \langle f \circ g, F' \circ g\rangle \\
&= \langle f \circ g, F\rangle \\
&= \theta(Hom_C(V_Y(g), X_1)(f))
\end{aligned}$$

It follows that the following diagram commutes and hence that θ is natural in F:

$$\begin{array}{ccc} Hom_{\mathcal{C}}(V_Y(F'), X_1) & \xrightarrow{\quad\theta\quad} & Hom_{\mathcal{C}/Y}(F', I_Y(X_1)) \\ {\scriptstyle Hom_{\mathcal{C}}(V_Y(g), X_1)}\Big\downarrow & & \Big\downarrow{\scriptstyle Hom_{\mathcal{C}/Y}} \\ Hom_{\mathcal{C}}(V_Y(F), X_1) & \xrightarrow{\quad\theta\quad} & Hom_{\mathcal{C}/Y}(F, I_Y(X_1)) \end{array}$$

Similarly, if $g : X_1 \longrightarrow X_1'$ then for any $f : V_Y(F) \longrightarrow X_1$,

$$\theta(g \circ f) = \langle g \circ f, F \rangle = (g \times Y) \circ \langle f, F \rangle$$

so θ is natural in X_1 and we have therefore demonstrated the adjunction. □

Lemma 2.12 *If $\mathcal{C}$ is a cartesian category and $G \equiv C \xrightarrow{g} Y$ is an object in $\mathcal{C}/Y$ then $I_Y(A) \times G$ exists in $\mathcal{C}/Y$ and is given by one of the sides in the folowing commuting square:*

$$\begin{array}{ccc} C & \xleftarrow{\quad\pi_2\quad} & A \times C \\ {\scriptstyle g}\Big\downarrow & & \Big\downarrow{\scriptstyle id_A \times g} \\ Y & \xleftarrow{\quad\pi_2\quad} & A \times Y \end{array}$$

Proof The square trivially commutes - we need to show that it is a pullback in $\mathcal{C}$.

If $C \xleftarrow{f_2} X \xrightarrow{f_1} A \times Y$ is a $\mathcal{C}$-diagram with $g \circ f_2 = \pi_2 \circ f_1$ then $\langle \pi_1 \circ f_1, f_2 \rangle : X \longrightarrow A \times C$ satisfies $\pi_2 \circ \langle \pi_1 \circ f_1, f_2 \rangle = f_2$ and

$$(id_A \times g) \circ \langle \pi_1 \circ f_1, f_2 \rangle = \langle \pi_1 \circ f_1, g \circ f_2 \rangle = \langle \pi_1 \circ f_1, \pi_2 \circ f_1 \rangle = f_1$$

Moreover any map $\lambda : X \longrightarrow A \times C$ with $\pi_2 \circ \lambda = f_2$ and $(id_A \times g) \circ \lambda = f_1$ determines a map $X \longrightarrow A \times Y \times C$ which has the universal property of $\langle f_1, f_2 \rangle$. The uniqueness of λ therefore follows from the uniqueness of $\langle f_1, f_2 \rangle$ and it is $\langle \pi_1 \circ f_1, f_2 \rangle$, as desired. □

Lemma 2.13

$$rec(I_Y(A), I_Y(B)) = [rec(\pi_2^{A \times Y}, \pi_2^{B \times Y}) : rec(A, B \times Y) \longrightarrow Y]$$

Now note that there is a unique comparison map c

$$rec(\langle\langle \pi_1, \pi_2 \circ \pi_1\rangle \circ r_1, \pi_2 \circ \pi_2\rangle, r_0 \times id_Y) : rec(A, B \times Y) \longrightarrow rec(A, B) \times Y$$

which is derived from the following recursive data:

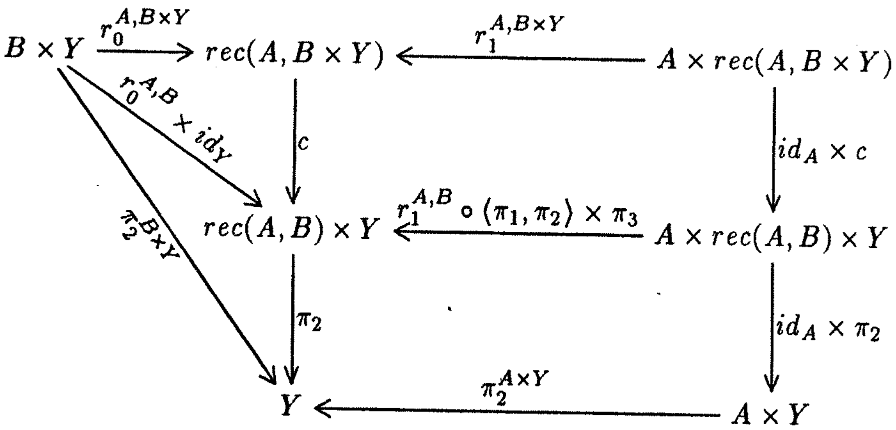
$$B \times Y \xrightarrow{r_0^{A,B} \times id_Y} rec(A,B) \times Y \xleftarrow{r_1^{A,B} \circ \langle \pi_1, \pi_2\rangle \times \pi_3} A \times rec(A,B) \times Y$$

Lemma 2.14 *Requiring that this comparison map is iso is equivalent to requiring that I_Y preserves the recursion.*

Proof We need only show that this is a $\mathcal{C}/Y$ map

$$[rec(\pi_2^{A\times Y}, \pi_2^{B\times Y}) : rec(A, B \times Y) \longrightarrow Y] \xrightarrow{\sim} [\pi_2 : rec(A,B) \times Y \longrightarrow Y]$$

This follows immediately from the commutativity of the following diagram and the uniqueness of $rec(\pi_2^{A\times Y}, \pi_2^{B\times Y})$.



We have $\pi_2 \circ c = rec(\pi_2^{A\times Y}, \pi_2^{B\times Y})$, as desired. □

Definition 2.15 (Parameterised Recursion) *The recursion is* parameterised *in the case where I_Y preserves recursion.*

Definition 2.16 *We define $list(A)$ to be $rec(A, 1)$. We may write A^* for $list(A)$.*

Note: $rec(A,B) \xrightarrow{\sim} rec(A, 1 \times B) \xrightarrow{\sim} rec(A,1) \times B \xrightarrow{\sim} list(A) \times B$ is a series of isomorphisms in a category which has parameterised recursion. We call B the *parameter* of the recursion.

Definition 2.17 *If C is a cartesian category then C is:*

1. arithmetic *if for every B $rec(1, B)$ exists and is parameterised;*

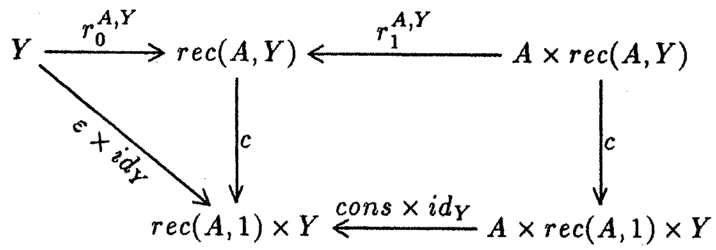
2. list-arithmetic *if for any pair A, B of C-objects, $rec(A, B)$ exists and is parameterised.*

2.2 List Arithmetic Categories

Definition 2.18 *In list-arithmetic categories, we write $r_0^{A,1}$ as $\varepsilon : 1 \longrightarrow A^*$ and $r_1^{A,1}$ as $cons : A \times A^* \longrightarrow A^*$.*

Lemma 2.19 *If C is a list-arithmetic category then we can write $r_0^{A,B}$ and $r_1^{A,B}$ as $\varepsilon \times id_B$ and $cons \times id_B$ respectively.*

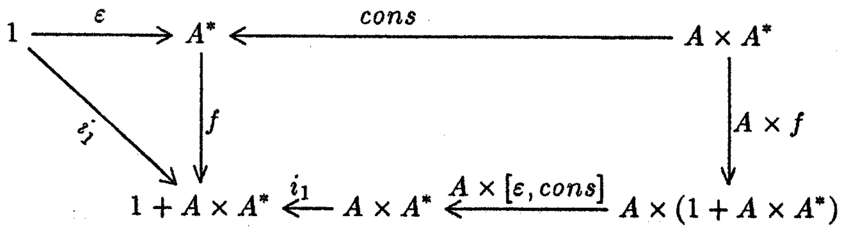
Proof This follows immediately from our requirement that the map c is iso, in the following case:



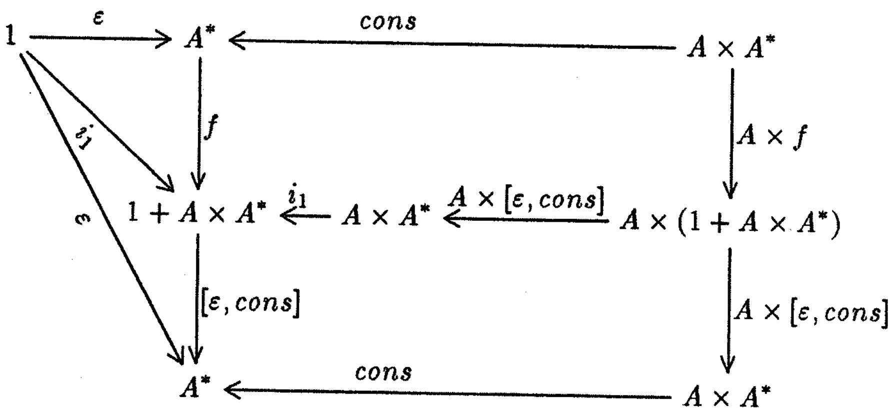
Since c is iso, any diagram $Y \xrightarrow{f_2} X \xleftarrow{f_1} A \times Y$ gives rise to a map $rec(A, 1) \times Y \xrightarrow{\lambda} X$ with the desired universal properties and such a map gives us a map $rec(A, Y) \xrightarrow{\lambda} X$ and must therefore be unique. $\square$

Lemma 2.20 *If C is list arithmetic then $list(A) \cong 1 + A \times list(A)$.*

Proof Let $f : list(A) \longrightarrow 1 + A \times list(A)$ be the unique map which makes the following diagram commute:



Then the following diagram commutes, since $[\varepsilon, cons] \circ i_2 \circ A \times [\varepsilon, cons] = cons \circ A \times [\varepsilon, cons]$.



By the uniqueness requirement, it follows that $[\varepsilon, cons] \circ f = A^*$.

$$\begin{aligned}\text{Also, } f \circ [\varepsilon, cons] &= [i_1, f \circ cons] \\ &= [i_1, i_2 \circ A \times [\varepsilon, cons] \circ f] \\ &= [i_1, i_2 \circ (A \times A^*)] \\ &= [i_1, i_2] \\ &= A^*\end{aligned}$$

So f is iso. □

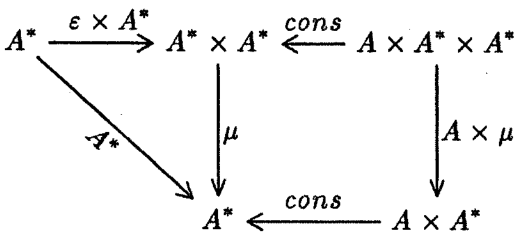
Definition 2.21 *We write the map f in the above Lemma as*

$$pop : list(A) \longrightarrow 1 + A \times list(A)$$

It will be useful in what follows to define some additional functions on lists which will be familiar from functional programming languages:

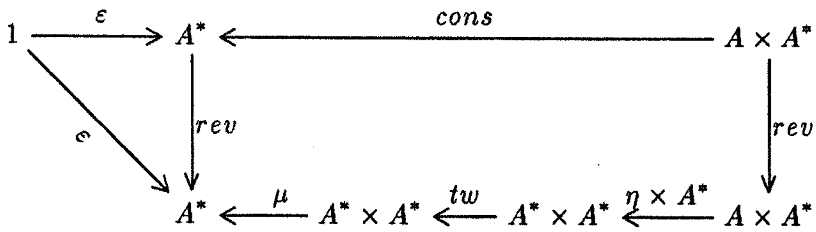
Definition 2.22 *In a list-arithmetic category C, we define:*

$back : A^* \longrightarrow A^* \equiv A^* \xrightarrow{pop} 1 + A \times A^* \xrightarrow{1+\pi_2} 1 + A^* \xrightarrow{[\varepsilon, A^*]} A^*$
$fst : A \times A^* \longrightarrow A \equiv A \times A^* \xrightarrow{\pi_1} A$
$\mu : A^* \times A^* \longrightarrow A^*$ *is given by the following recursion schema:*



$\eta : A \longrightarrow A^* \equiv A \xrightarrow{A\times\varepsilon} A \times A^* \xrightarrow{cons} A^*$

$rev : A^* \longrightarrow A^*$ is given by the following recursion schema:



where tw is the usual isomorphism $A \times B \xrightarrow{\sim} B \times A$.

$front : A^* \longrightarrow A^* \equiv A^* \xrightarrow{rev} A^* \xrightarrow{back} A^* \xrightarrow{rev}$

$last : A \times A^* \longrightarrow A \equiv A \times A^* \xrightarrow{A\times rev} A \times A^* \xrightarrow{A\times pop} A \times (1 + A \times A^*)$

$\xrightarrow{\delta} A + (A \times A \times A^*) \xrightarrow{[A,\pi_2]} A$

where $\delta : A \times (B + C) \longrightarrow (A \times B) + (A \times C)$ is the canonical map of chapter 4; we shall use last only when δ is iso.

Lemma 2.23 *If C is list-arithmetic then for any $A \in Ob(C)$, $list(A)$ is the free monoid over A.*

Proof $\varepsilon : 1 \longrightarrow list(A)$ will be the unit of the monoid.

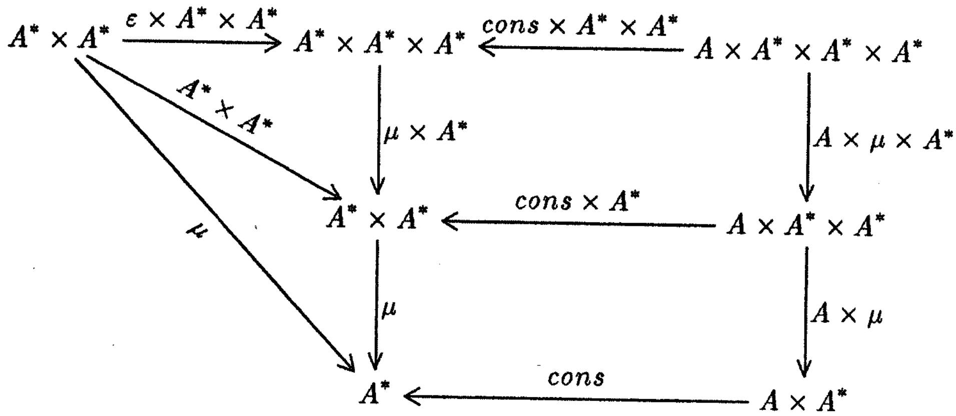
$\mu : list(A) \times list(A) \longrightarrow list(A)$ is the map defined in 2.22.

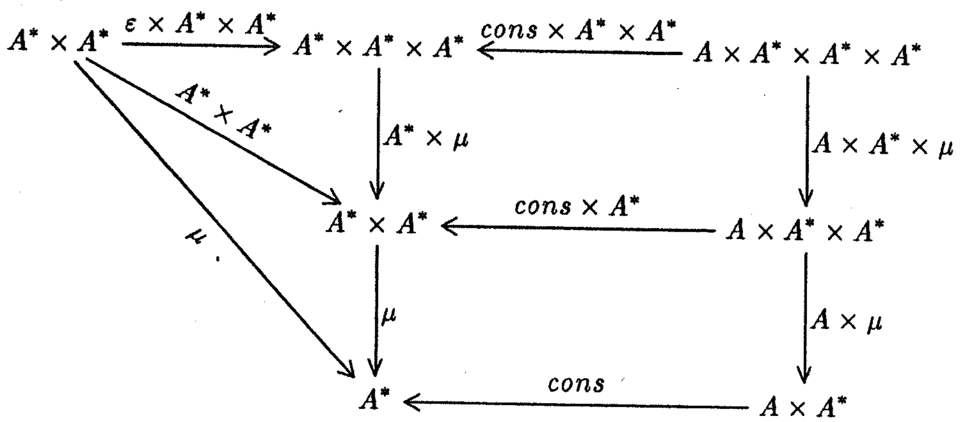
Recall from Lemma 2.19 that:

$r_0^{A,list(A)\times list(A)} \equiv \varepsilon \times list(A) \times list(A)$

$r_1^{A,list(A)\times list(A)} \equiv cons \times list(A) \times list(A)$

Associativity of μ then follows trivially from the commutativity of the following diagrams:

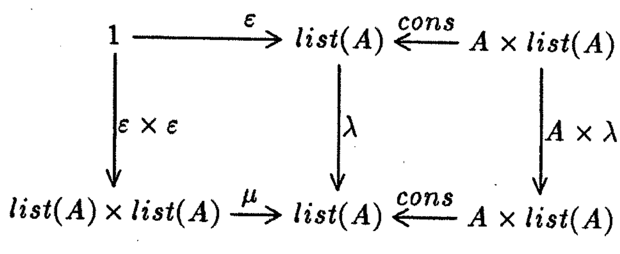




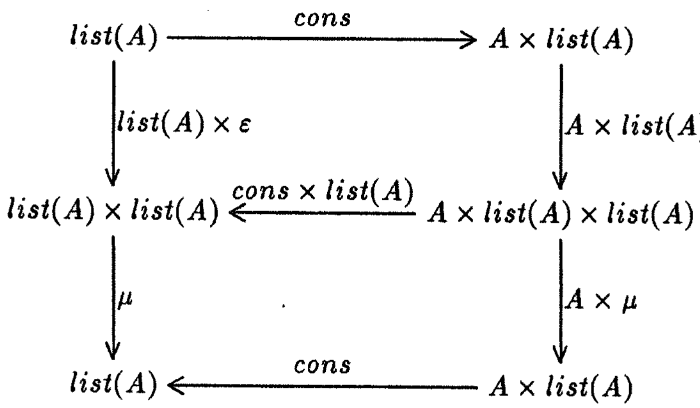
By the universal property of $rec(cons, \mu)$, we must have that $\mu \circ (\mu \times list(A)) = \mu \circ (list(A) \times \mu)$, as desired.

To see that ε is a left inverse for μ, observe that the left hand triangle in the defining diagram for μ gives us $\mu \circ (\varepsilon \times list(A)) = list(A)$.

To show that ε is a right inverse for μ, we employ induction:



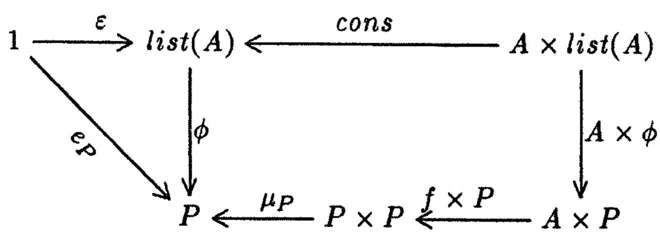
This diagram commutes if we take $\lambda \equiv list(A)$; the right hand side does so trivially, the left hand side by the definition of μ. However, with $\lambda \equiv \mu \circ (list(A) \times \varepsilon)$, the left hand square trivially commutes and so does the right hand one, since the following diagram commutes, the top trivially and the bottom by definition of μ:



Since λ must be unique, we have $\mu \circ (list(A) \times \varepsilon) = list(A)$, so ε is a right inverse for μ.

We have therefore demonstrated that $list(A)$ is a monoid.

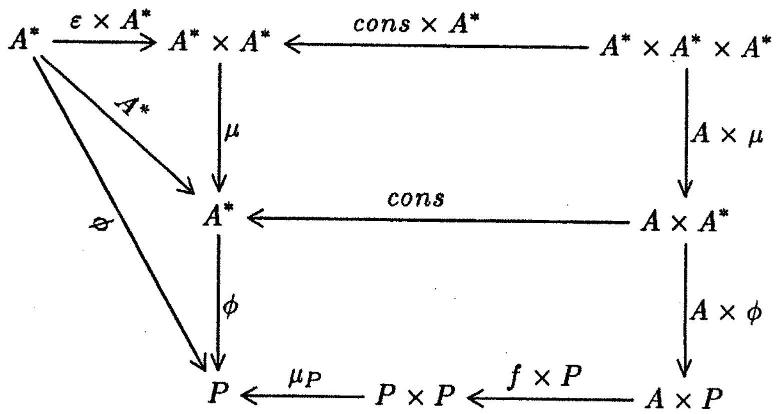
Suppose that (P, e_P, μ_P) is a $\mathcal{C}$-monoid with $f : S \longrightarrow P$ a $\mathcal{C}$-arrow. Define $\phi : list(A) \longrightarrow P$ to be the unique arrow which makes the following diagram commute:

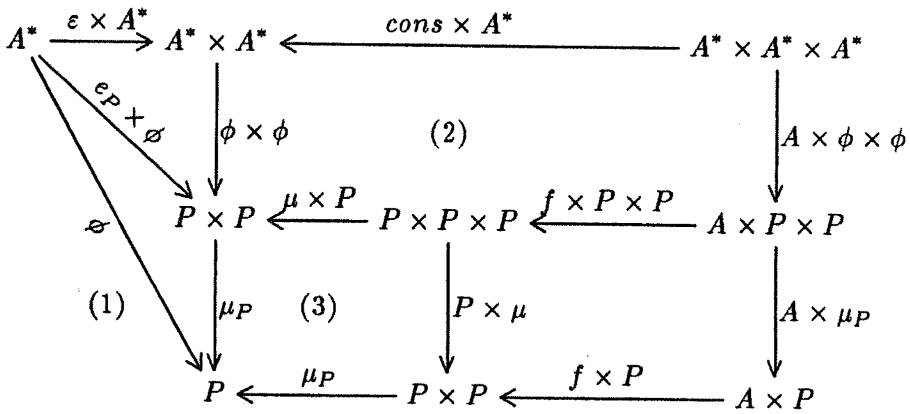


We claim that ϕ is the unique monoid homomorphism which we require to complete the proof.

ϕ trivially preserves units.

To see that ϕ preserves multiplication, observe that the following diagrams commute:





Commutativity of the first diagram is trivial. Commutativity of the second diagram comes from the commutativity of the sub-diagrams - (1) is the left unit for P, (2) is the definition of μ and (3) is the associativity of μ_P.

Since the arrow obtained from the recursion data

$$A^* \xrightarrow{\phi} P \xleftarrow{\mu_P} P \times P \xleftarrow{f \times P A \times P}$$

must be unique, we have that $\mu_P \circ (\phi \times \phi) = \phi \circ \mu$ as desired and so we have shown that ϕ is a monoid homomorphism.

Define $\eta : A \longrightarrow A^*$ to be the composite

$$A \xrightarrow{A \times \varepsilon} A \times A^* \xrightarrow{cons} A^*$$

Then

$$
\begin{aligned}
\phi \circ \eta &= \phi \circ cons \circ (A \times \varepsilon) \\
&= \mu_P \circ (f \times P) \circ (A \times \phi) \circ (A \times \varepsilon) \\
&= \mu_P \circ (f \times (\phi \circ \varepsilon)) \\
&= \mu_P \circ (f \times e_P) \\
&= f
\end{aligned}
$$

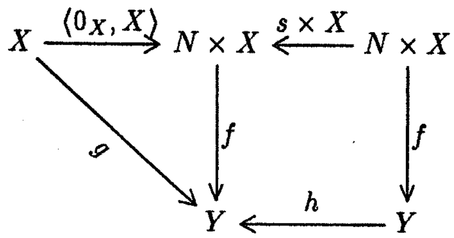
For uniqueness, suppose that $\lambda : A^* \longrightarrow P$ is a monoid homomorphism with $\lambda \circ \eta = f$ and observe that $\mu \circ (\eta \times A^*) = cons$. Then we have:

$$
\begin{aligned}
\lambda \circ cons &= \lambda \circ \mu \circ (\eta \times A^*) \\
&= \mu_P \circ (\lambda \times \lambda) \circ (\eta \times A^*) \\
&= \mu_P \circ (f \times \lambda)
\end{aligned}
$$

So λ makes the defining diagram for ϕ commute and so by the usual uniqueness argument, $\lambda = \phi$, which completes the proof. □

2.3 Natural Numbers Objects

Definition 2.24 *A* Natural Numbers Object (NNO) *in a cartesian category $\mathcal{C}$ is a diagram $1 \xrightarrow{0} N \xleftarrow{s} N$ in $\mathcal{C}$ such that given any diagram $X \xrightarrow{g} Y \xleftarrow{h} Y$ in $\mathcal{C}$ there is a unique $f : N \times X \longrightarrow Y$ which makes the following diagram commute:*



We will use the notion of a Natural Numbers Object extensively. Effectively, a NNO allows us to interpret the induction theorem with parameters. There are a number of available definitions - the original one, due to Lawvere, existed in a Cartesian Closed Category and replaced the object X in the above definition by the terminal object 1 - it is easy to show that in such a category these definitions are equivalent. Our definition is designed for use in weaker categories and the object which we have defined is sometimes referred to as a *parameterised NNO*.

Lemma 2.25 *A cartesian category has a NNO iff it is arithmetic.*

Proof This is trivially the case. N is $rec(1, X)$. □

Definition 2.26 *Given arrows* $X \xrightarrow{g} Y$ *and* $N \times X \times Y \xrightarrow{h} Y$, *we say that* f *and* g *determine a function* $f : N \times X \longrightarrow Y$ by primitive recursion *if there is a unique* $f : N \times X \longrightarrow Y$ *such that* $f \circ \langle 0_X, X \rangle = g$ *and* $h \circ (N \times X, f) = f \circ (s \times X)$.

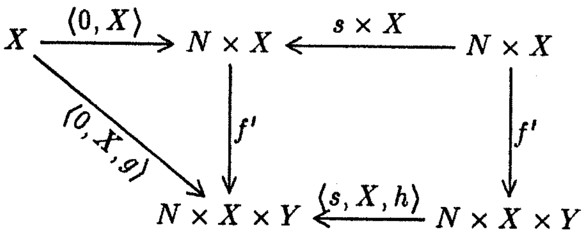
ie. if a definition of the form:

$$f(0, x) = g(x)$$
$$f(s(n), x) = h(n, x, f(n, x))$$

is valid.

Lemma 2.27 *If a cartesian category* C *has a NNO as defined in 2.24 then any two arrows* $X \xrightarrow{g} Y$ *and* $N \times X \times Y \xrightarrow{h} Y$ *determine a function* $f : N \times X \longrightarrow Y$ *by primitive recursion.*

Proof Letf' be the arrow determined by the following recursion schema:

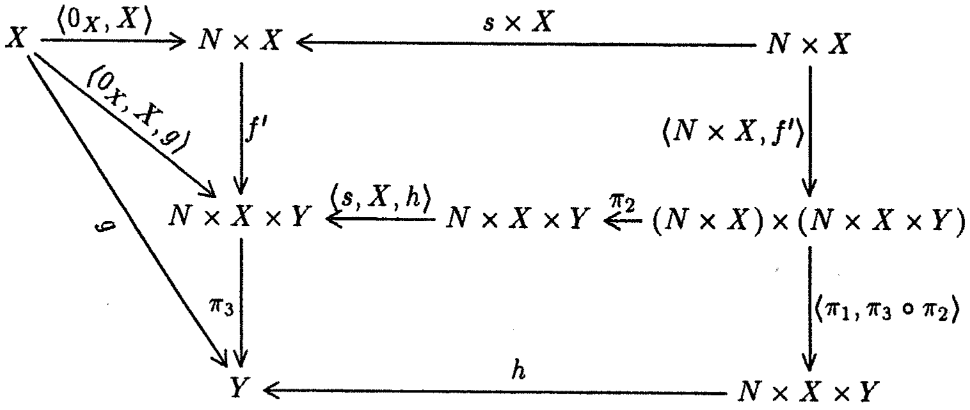


Observe the following:

$\pi_1 \circ f' \circ \langle 0, X \rangle = 0$ and $\pi_1 \circ \langle 0, X \rangle = 0$
$\pi_1 \circ f' \circ (s \times X) = s \circ \pi_1 \circ f'$ and $\pi_1 \circ (s \times X) = s \circ \pi_1$

So by the universal property of recursively defined maps using NNO's, $\pi_1 \circ f' = \pi_1$. Similarly, $\pi_2 \circ f' = \pi_2$.

Now consider the following diagram:



We know that the top right hand rectangle commutes by the definition of f', so we can write

$$
\begin{aligned}
\pi_3 \circ f' \circ (s \times X) &= \pi_3 \circ \langle s, X, h \rangle \\
&= h \circ f' \\
&= h \circ \langle \pi_1 \circ f', \pi_2 \circ f', \pi_3 \circ f' \rangle \\
&= h \circ \langle \pi_1^{N \times X}, \pi_2^{N \times X}, \pi_3 \circ f' \rangle \\
&= h \circ \langle \pi_1^{(N\times X)\times(N\times X\times Y)}, \pi_3^{(N\times X\times Y)} \circ \pi_2^{(N\times X)\times(N\times X\times Y)} \rangle \\
&\quad \circ \langle N \times X, f' \rangle
\end{aligned}
$$

So $\pi_3 \circ f'$ is a solution to the recursive definition. It is easy to see that any solution gives rise to a solution to the NNO recursion used to define f' and $\pi_3 \circ f'$ is therefore the *unique* solution to the recursive equation. □

2.4 Inductive Proofs in Arithmetic Categories

We now justify the use of familiar styles of induction in Arithmetic Categories. In this subsection, let C be a category with pullbacks.

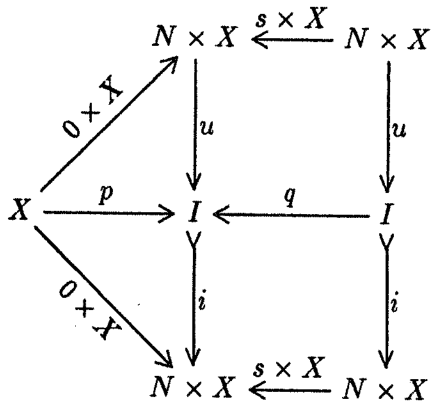
Definition 2.28 *A monic* $i : I \rightarrowtail A$ is closed under $f : A \longrightarrow A$ *if for any* $T \xrightarrow{x} A$, $x \in i \Rightarrow f \circ x \in i$.

Lemma 2.29 $i : I \rightarrowtail A$ *is closed under* $f : A \longrightarrow A$ *iff there is some* $h : I \longrightarrow I$ *with* $f \circ i = i \circ h$.

Proof If $x \in i \Rightarrow f \circ x \in i$ then $f \circ i \in i \Rightarrow \exists h. f \circ i = i \circ h$. If $\exists h. f \circ i = i \circ h$ and $x \in i$ then since $i \in f^{-1}(x)$ we have $f(x) \in i$. □

Lemma 2.30 *Suppose that C has NNO N. Then if $i : I \rightarrowtail N \times X$ has $0 \times X \in i$ and is closed under $s \times X$ then $i \cong id_{N \times X}$.*

Proof There are p, q such that the lower part of the following diagram cimmutes:



Then define n as shown from recursion data p, q. Then $i \circ u$ must be $id_{N \times X}$, by the uniqueness criterion for recursively defined data. u is therefore both monic and split epic and is therefore iso. □

Corollary 2.31 *If C has NNO N and $f, g : N \times X \longrightarrow Y$ have $f(0, x) = g(0, x)$ and $f(n, x) = g(n, x) \Rightarrow f(s(n), x) = g(s(n), x)$ then $f = g$.*

Proof By Lemma 2.30, the equaliser of f and g is an isomorphism. □

We can extend definition 2.28 as follows:

Definition 2.32 *A monic $i : I \rightarrowtail C$ is* closed to *$f : A \times C \longrightarrow C$ if $\langle a, x \rangle \in A \times i \Rightarrow f \circ \langle a, x \rangle \in i$.*

Lemma 2.33 *$i : I \rightarrowtail C$ is closed to f iff $\exists q : A \times I \longrightarrow I$ such that $i \circ q = f \circ (A \times i)$.*

Proof Precisely as for 2.29. □

Lemma 2.34 *Suppose that C has list objects. If $I \overset{i}{\rightarrowtail} A^* \times B$ has $\varepsilon \times B \in I$ and is closed under $cons \times B$ then $i \cong id_{A^*}$.*

Proof Analogous to 2.30. □

Corollary 2.35 *If C has list objects and $f, g : A^* \times B \longrightarrow X$ satisfy*

$$f(\varepsilon, b) = g(\varepsilon, b)$$

and

$$f(\mathbf{a}, b) = g(\mathbf{a}, b) \Rightarrow f(cons(a_1, \mathbf{a}), b) = g(cons(a_1, \mathbf{a}), b)$$

then $f \equiv g$.

Proof Precisely as for 2.31. □

Chapter 3

Internal Categories

In this short chapter, we define the notion of category object in a category. This concept is analogous to that of an internal monoid object of the previous section - we use the equational definition of a category to give its definition in terms of arrows with commuting diagrams and then we define a category object to be any collection of objects and arrows in a suitable category which satisfies our requirements.

All of this formalism is a special case of the more general categorical notion of a theory, which we introduce in chapter 7.

Definition 3.1 *Let C be any cartesian category. An* internal graph *in C is any diagram of the form* $G_1 \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} G_0$.

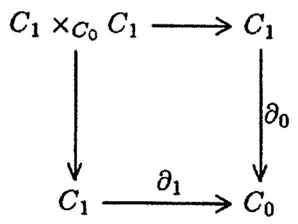
Definition 3.2 *An* internal graph homomorphism *between internal graphs* **G** *and* **H** *is a pair of maps* $\phi_1 : G_1 \longrightarrow H_1$ *and* $\phi_0 : G_0 \longrightarrow H_0$ *with* $\partial_1^H \circ \phi_1 = \phi_0 \circ \partial_1^G$ *and* $\partial_0^H \circ \phi_1 = \phi_0 \circ \partial_0^G$.

Definition 3.3 *Let C be any finitely complete category. An* internal category *in C consists of:*

1. *A pair of objects C_0, C_1 which we will refer to as the* object of objects *and the* object of arrows respectively*;*

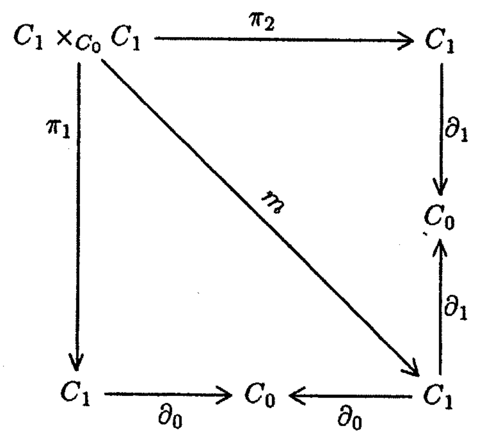
2. *Morphisms* $C_1 \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} C_0$, $C_0 \xrightarrow{e} C_1$ *and* $C_1 \times_{C_0} C_1 \xrightarrow{m} C_1$, *where*

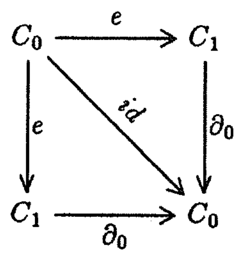
$C_1 \times_{C_0} C_1$ is the following pullback:

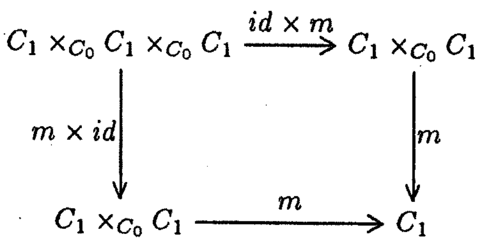


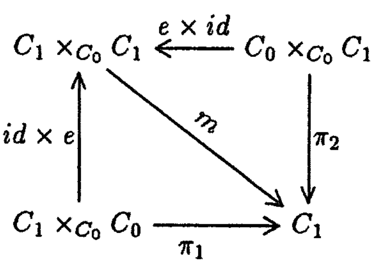
We refer to these as the source, target, unit *and* composition arrows *respectively.*

We require the following diagrams to commute:









Definition 3.4 *An internal functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ between internal categories $\mathbf{C}$ and $\mathbf{D}$ is a pair of morphisms $F_0 : C_0 \longrightarrow D_0$ and $F_1 : C_1 \longrightarrow D_1$ in $\mathcal{C}$ which makes the obvious diagrams with $e, d_0, d_1, comp$ commute.*

Definition 3.5 *If $\mathcal{C}$ is any finitely complete category, $Cat(\mathcal{C})$ is the category of internal categories and internal morphisms in $\mathcal{C}$.*

Definition 3.6 *Let $\mathbf{G}$ be an internal graph in a left exact category $\mathcal{C}$. If it exists, the* free category over $\mathbf{G}$ *is an internal category $\mathbf{C}$ in $\mathcal{C}$ with an* *internal graph homomorphsim* $\eta : \mathbf{G} \longrightarrow \mathbf{C}$ *such that for any internal category* $\mathbf{D}$ *in* $\mathcal{C}$ *for which there is a graph homomorphism* $\phi : \mathbf{G} \longrightarrow \mathbf{D}$, *there is a unique internal functor* $\bar{\phi} : \mathbf{C} \longrightarrow \mathbf{D}$ *with* $\bar{\phi} \circ \eta = \phi$.

Lemma 3.7 *The free internal category over an internal graph* $\mathbf{G}$ *is unique up to isomorphism.*

Proof Trivial. □

In our usual external category theory, functors $F : \mathcal{C} \longrightarrow \mathcal{D}$ between small categories play a different role to functors from small categories into the ambient category **Set**. The following definitions expands this notion to functors from internal categories in a category $\mathcal{C}$ to $\mathcal{C}$. We start by generalising the concept of an external graph morphism from $\mathcal{G}$ to **Set** and then extend this to a functor from a category $\mathcal{C}$ to **Set**.

Definition 3.8 *Let* $\mathbf{G} = \langle G_0, G_1, d_0, d_1 \rangle$ *be an internal graph in a category* $\mathcal{C}$ *which has pullbacks. An* internal diagram on $\mathbf{G}$, *or* left $\mathbf{G}$-action in $\mathcal{C}$, *is an object*

$$\pi : F \longrightarrow G_0$$

of $\mathcal{C}/G_0$, *with an* action

$$\mu : G_1 \times_{G_0} F \longrightarrow F$$

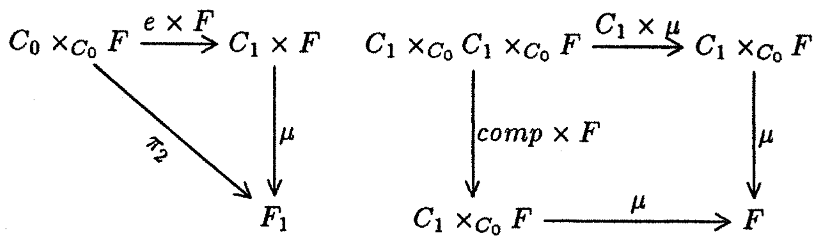
where $G_1 \times_{G_0} F$ *is the pullback of* π *along* d_0, *such that the following diagram commutes:*

$$\begin{array}{ccc} G_1 \times_{G_0} F & \xrightarrow{\mu} & F \\ \downarrow{\scriptstyle \pi_1} & & \downarrow{\scriptstyle \pi} \\ G_1 & \xrightarrow{d_1} & G_0 \end{array}$$

Here the map $\pi : F \longrightarrow G_0$ should be regarded as a G_0-indexed family of $\mathcal{C}$-objects which corresponds to the object function for an external graph morphism $H : \mathbf{G} \longrightarrow \mathbf{Set}$. In **Set**, $H(A)$ can be extracted as $\pi^{-1}(A)$.

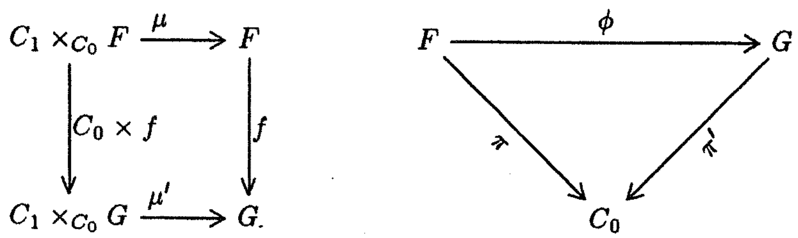
The map $\mu : G_1 \times_{G_0} F \longrightarrow F$ corresponds in **Set** to the operation $\langle f, x \rangle \mapsto f.x$, where $f : A \longrightarrow B$, $x \in H(A)$ and $f.x = (Hf)(x)$. The commutativity of the diagram gives us $f.x \in H(B)$.

Definition 3.9 *Let* $\mathbf{C} = \langle C_0, C_1, d_0, d_1, e, comp\rangle$ *be an internal category in a category* $\mathcal{C}$ *which has pullbacks. An* internal diagram on $\mathbf{C}$, *or* left $\mathbf{G}$-action in $\mathcal{C}$, *is an internal diagram* (F, π, μ) *on the underlying graph of* $\mathbf{C}$ *which makes the following diagrams commute:*



These diagrams express the unit and associativity laws respectively for the internal functor.

Definition 3.10 *A* morphism of internal diagrams over $\mathbf{C}$ *between internal diagrams* $\langle F, \pi, \mu\rangle$ *and* $\langle G, \pi', \mu'\rangle$ *on* $\mathbf{C}$ *in* $\mathcal{A}$ *is an arrow* $f : F \longrightarrow G$ *in* $\mathcal{A}$ *which makes the following diagrams commute:*



Definition 3.11 $IAct(\mathbf{C}, \mathcal{A})$ *is the category of internal diagrams over* $\mathbf{C}$ *in* $\mathcal{A}$ *and morphisms of internal diagrams over* $\mathbf{C}$.

Note that the usual notation for $IAct(\mathbf{C}, \mathcal{A})$ is $\mathcal{A}^{\mathbf{C}}$ - we have avoided this in order to prevent confusion with definition 7.8.

Chapter 4

Distributivity

We will see that the categories in which we are principally interested are distributive. In this section, we briefly overview the relevant definitions and some of the key results. The results appear in great detail in [CLW 93] and require no amplification; we therefore give sketches of only the more interesting proofs.

Definition 4.1 *A* distributive category *is one with finite sums and products in which the canonical map* $\delta : (A \times B) + (A \times C) \longrightarrow A \times (B + C)$ *is an isomorphism.*

The map δ is the following one:

$$(A \times B) + (A \times C) \xrightarrow{\langle [\pi_1^{A,B}, \pi_1 A,C], [\iota_1^{B,C} \circ \pi_2^{A,B}, \iota_2^{B,C} \circ \pi_2^{A,C}] \rangle} A \times (B + C)$$

Some authors use the following notion as their definition for distributivity:

Definition 4.2 *An* extensive category $\mathcal{A}$ *is one with finite sums and products such that the canonical functor* $\mathcal{A}/A \times \mathcal{A}/B \longrightarrow \mathcal{A}/(A + B)$ *is an equivalence.*

Firstly, we state a technical lemma:

Lemma 4.3 *A category $\mathcal{A}$ with finite sums is extensive iff it has pullbacks along coprojections of coproducts and every commutative diagram*

$$\begin{array}{ccccc} A_1 & \xrightarrow{a_1} & A & \xleftarrow{a_2} & A_2 \\ \downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f_2} \\ X_1 & \xrightarrow{x_1} & X_1 + X_2 & \xleftarrow{x_2} & X_2 \end{array}$$

comprises a pair of pullback squares in $\mathcal{A}$ just when the top row is a coproduct diagram in $\mathcal{A}$.

Proof Appears in [CLW 93] in great detail. □

Definition 4.4 *In a category with finite sums and pullbacks along their injections a coproduct diagram $X_1 \xrightarrow{x_1} X_1 + X_2 \xleftarrow{x_2} X_2$ is* universal *if pulling it back along any arrow into $X_1 + X_2$ gives a coproduct diagram.*

It is a trivial consequence of Lemma 4.3 that sums are universal in any extensive category.

Definition 4.5 *In a category with finite sums and pullbacks along their injections, sums are* disjoint *if the pullback of the injections of a binary sum is the initial object and all injections are monic.*

Lemma 4.6 *Sums are disjoint in an extensive category.*

Proof Easy consequence of Lemma 4.3. □

[CLW 93] prove the following converse to Lemmas 4.3 and 4.6:

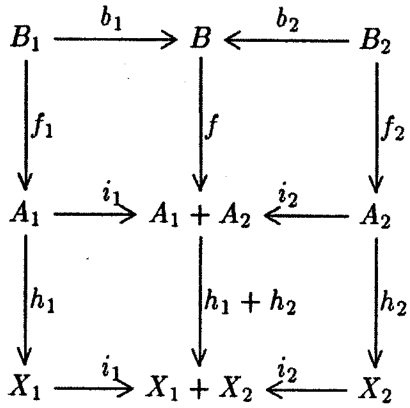
Theorem 4.7 *A category with finite sums and pullbacks along their injections is extensive iff the sums are universal and disjoint.*

Proof "Only If" - see above.

"If" - Universality of sums gives us half of the condition of Lemma 4.3. We want to show that with universal and disjoint sums the following is a pullback:

$$\begin{array}{ccccc} A_1 & \xrightarrow{i_1} & A_1 + A_2 & \xleftarrow{i_2} & A_2 \\ \downarrow{\scriptstyle h_1} & & \downarrow{\scriptstyle h_1 + h_2} & & \downarrow{\scriptstyle h_2} \\ X_1 & \xrightarrow{x_1} & X_1 + X_2 & \xleftarrow{x_2} & X_2 \end{array}$$

Suppose that $X_1 \xleftarrow{g} B \xrightarrow{f} A_1 + A_2$ with $(h_1 + h_2) \circ f = x_1 \circ g$. Pull the above diagram back along f:



Now $x_1 \circ g \circ b_2 = (h_1 + h_2) \circ f \circ b_2 = x_2 \circ h_2 \circ f_2$.

So there is a unique arrow from B_2 to $x_1^{-1}(X_2)$, which is initial since sums are disjoint. It follows that B_2 is initial. Since sums are universal, $B = B_1 + B_2 \cong B_1$, so b_1 is invertible. $f_1 \circ b_1^{-1}$ is the desired map $\langle f, g \rangle : B \longrightarrow A$. Similarly, A_2 is a pullback. □

Lemma 4.8 *In a distributive category, injections are monic.*

Proof [CLW 93] □

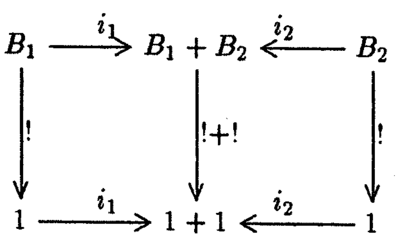
Lemma 4.9 *Initials are strict in a distributive category. (ie any arrow $A \longrightarrow 0$ is invertible).*

Proof [CLW 93] □

Theorem 4.10 *An extensive category with products is distributive.*

Proof Suppose that $A, B_1, B_2 \in Ob(C)$.

We know that the following is a coproduct diagram:



hence by Lemma 4.3 it is a pullback diagram.

In any category, if this is a pullback then so is

$$\begin{array}{ccccc}
A \times B_1 & \xrightarrow{A \times i_1} & A \times (B_1 \times B_2) & \xleftarrow{A \times i_2} & A \times B_2 \\
\Big\downarrow ! & & \Big\downarrow (!+!) \circ \pi_2 & & \Big\downarrow ! \\
1 & \xrightarrow{\quad i_1 \quad} & 1 + 1 & \xleftarrow{\quad i_2 \quad} & 1
\end{array}$$

It follows from Lemma 4.6 that this diagram is a coproduct. This is exactly the requirement of distributivity. □

[CLW 93] note in passing that the converse is not true: $\wp(X)$ for a set X is distributive but not extensive.

Chapter 5

Skolem Theories and A.U. Construction

In this chapter, we define Skolem Theories and exmaine their properties using a language PRIM which appeared in outline in [GCW Notes]. We then examine Joyal's completion process from a Skolem Theory to an Arithmetic Universe in some detail; this process is sketched in [GCW Notes] and an outline of the proofs of the existence of free monoids in E and of finite limits, coproducts and surjective images in $\breve{E}$ appears in that document.

Definition 5.1 *A* Skolem Category *is a cartesian category with a NNO. A* Skolem Theory *is a Skolem Category in which every object is a power of N.*

There is an initial Skolem Theory which is generated by the diagram $1 \xrightarrow{0} N \xleftarrow{s} N$.

Definition 5.2 *A* morphism of Skolem theories *is a Lex functor which preserves recursion.*

Throughout this chapter, we will write Σ for the initial Skolem Theory and E for an arbitary Skolem Theory.

5.1 The Language PRIM

GCW mentions that the arrows of a Skolem Theory E can be related to programs in a rudimentary programming language. We will examine this approach in greater depth.

Definition 5.3 *PRIM is a language which uses a countable collection of registers x,y,z,..., each of which holds a natural number, with statements which have the following intended meanings:*

`SKIP`	*do nothing*
`CLR x`	*set* `x` *to zero*
`INC x`	*increment* `x`
`LOOP x` {*Instructions not referring to* `x` }	*do Instructions* `x` *times*

We define a code block *in PRIM to be C, where:*

`C ::= SKIP | INC x | CLR x | LOOP x {C₁} | C₁;C₂`

We define a PRIM program $P : \mathbb{N}^r \longrightarrow \mathbb{N}^s$ *to be the following, where C is a code block:*

```
DEF P(x)
    C
RETURN(y)
```

The intended meaning here is that the paremeters x_i used to define P will be placed in the first r registers of the PRIM machine and all other registers will be zeroed. The placeholders $x_1, x_2, \ldots, x_r$ in P which refer to them will then refer to the relevant register during execution. After execution, the elements of **y** will be placed in the first s registers of the PRIM machine and all others will be zeroed.

We can use this intuition to define an operational semantics for PRIM:

Definition 5.4 *A* memory state *is a function* $\sigma : \mathbb{N} \longrightarrow \mathbb{N}$ *which gives the values held in each of the registers of the PRIM-machine during execution. We will denote by* Σ *the set of all memory states.*

The common nomenclature for the initial Skolem Theory and the set of all memory states is regrettable but should not in practice cause any confusion. Note that any PRIM program takes only a finitely large list of input parameters and is of finite length so we can assume that all elements of Σ are finite - we will not want to interpret Σ as a Skolem Theory object, so this will suffice for our purposes. Our operational semantics will now be a syntax-directed system of transitions on the memory state of the PRIM machine.

Definition 5.5 *A* configuration *for a PRIM machine is one of the following:*

1. $\langle C, \sigma \rangle$, *where C is an unexecuted code block and $\sigma \in \Sigma$;*
2. σ, *for $\sigma \in \Sigma$, in which case no further execution remains and the program has terminated.*

Definition 5.6 *The transition relation $\leadsto$ is a binary relation on configurations. To define it, we will require some auxilary functions, all of which map Σ to Σ:*

1. $succ_n(\sigma)(m) = \begin{cases} \sigma(m) & if\ n \neq m \\ \sigma(m)+1 & if\ n = m \end{cases}$

2. $pred_n(\sigma)(m) = \begin{cases} \sigma(m) & if\ n \neq m \\ \sigma(m)-1 & if\ n = m\ and\ \sigma(n) > 0 \\ \sigma(m) & if\ n = m\ and\ \sigma(n) = 0 \end{cases}$

3. $zero_n(\sigma)(m) = \begin{cases} \sigma(m) & if\ n \neq m \\ 0 & if\ n = m \end{cases}$

We now define $\leadsto$ to be the smallest relation which satisfies the following:

1. $\langle \mathtt{SKIP},\ \sigma \rangle \leadsto \sigma$
2. $\langle \mathtt{INC\ x},\ \sigma \rangle \leadsto \mathbf{succ_x} \circ \sigma$
3. $\langle \mathtt{CLR\ x},\ \sigma \rangle \leadsto \mathbf{zero_x} \circ \sigma$
4. $\langle \mathtt{LOOP\ z\ C},\ \sigma \rangle \leadsto \begin{cases} \sigma, & if\ \sigma(z) = 0 \\ \langle C; LOOP\ z\ \{C\}; INC\ z, pred_z \circ \sigma \rangle, else \end{cases}$
5. $\langle \mathtt{C_1},\ \sigma \rangle \leadsto \langle \mathtt{C_1'},\ \sigma' \rangle \Rightarrow \langle \mathtt{C_1; c_2},\ \sigma \rangle \leadsto \langle \mathtt{C_1'; C_2},\ \sigma' \rangle$
6. $\langle \mathtt{C_1},\ \sigma \rangle \leadsto \sigma' \Rightarrow \langle \mathtt{C_1; C_2},\ \sigma \rangle \leadsto \langle \mathtt{C_2},\ \sigma' \rangle$

Lemma 5.7 *Given any code block C and memory state σ in PRIM, there is a unique memory state σ' such that $\langle C, \sigma \rangle \leadsto^* \sigma'$, where $\leadsto^*$ is the transitive reflexive closure of $\leadsto$.*

Proof This is a trivial induction which we ommit. □

Definition 5.8 *ρ maps finite parameter lists to memory states. A parameter list $\langle x_1, x_2, \ldots, x_n \rangle$, is mapped under ρ to $[1 \mapsto x_1, \ldots, n \mapsto x_n]$.*

Definition 5.9 *The operator $\mathcal{O}$ maps PRIM programs to memory states. If* P *has code block C and*

$$\langle C, \rho\langle x_1, \ldots, x_n \rangle\rangle \leadsto^* \sigma'$$

then if the RETURN *statement of P is* RETURN(**x**) *then*

$$\mathcal{O}[\![\mathrm{P}]\!]\langle x_1, x_2, \ldots, x_n\rangle = [x \mapsto 0, x \notin \mathbf{x}]\sigma'$$

Lemma 5.10 *PRIM has the following properties:*

1. *Assignment of register* x *to register* y *can be defined and is written* y $\longleftarrow$ x*;*
2. *If P is any PRIM program then we can run it within another PRIM program Q.*

Proof

1. Trivial:

```
CLR y;
LOOP x {INC y}
```

2. Also trivial. Suppose we wish to run $\mathrm{P}(x_1, \ldots, x_r)$ within Q and to place the output in registers $y_1, \ldots, y_s$. Since P is finite, there is a number n which is the maximum register used by P; let $m = \max(n, r, s)$. We can therefore run the following commands within Q to accomplish our goal:

m + 1 $\longleftarrow$ 1
...
m + m $\longleftarrow$ m
1 $\longleftarrow$ $\mathrm{x_1}$
...
r $\longleftarrow$ $\mathrm{x_r}$
All of the commands of $\mathrm{P}(\mathrm{x_1}, \ldots, \mathrm{x_r})$

```
y1 + m <— 1
...
ys + m <— s
1 <— m + 1
...
m <— m + m
CLR m+1
...
CLR m+m
```

□

In view of the above, we will allow commands of the form

$$\texttt{y} \longleftarrow \texttt{P}(\texttt{x}_1, \ldots, \texttt{x}_m)$$

where P is a PRIM program.

Lemma 5.11 *The functions defineable in PRIM are precisely the primitive recursive functions.*

Proof PRIM contains the functions **0**, succ and $p_i^n : (x_1, \ldots, x_n) \mapsto x_i$, trivially. As a consequence of Lemma 5.10, PRIM is closed under substitution. Given $g : \mathbb{N} \longrightarrow \mathbb{N}$ and $h : \mathbb{N}^3 \longrightarrow \mathbb{N}$ as recusive data for a function f, we can calculate f using the following PRIM routine:

```
DEF f(n,x)
    y <— g(x)
    i <— 0
    LOOP n{y <— h(i,x,y)
           i <— s(i)}
RETURN(y)
```

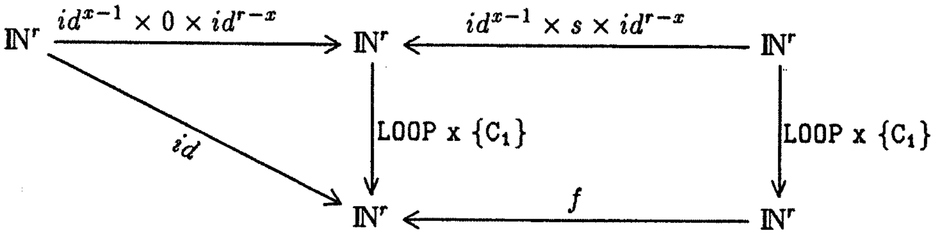
So PRIM can define all primitive recursive functions.

We show that all PRIM functions are recursively definable by induction on the length of the program. For the base case, note that `INC x` and `CLR x` are both primitive recursive functions. Then for the inductive step, assume that programs with $\leq n$ instructions are primitive recursive functions. Then any program P of length $n+1$ must have one of the following formats:

CLR x;C or INC x;C or LOOP x {C};C′

where C and C′ are code blocks. Since INC and CLR are both primitive recursive functions and PRIM is closed under substitution, the first two of these must be primitive recusive functions.

Consider the LOOP statement. Since C is shorter than P it must define a recursively definable function $f : \mathbb{N}^r \longrightarrow \mathbb{N}^r$, where r is at least the number of registers employed in the execution of P. Now $\mathbb{N}^r \xrightarrow{id} \mathbb{N}^r \xleftarrow{f} \mathbb{N}^r$ is recursive data for the LOOP statement:



It follows that P defines a primitive recursive function and this completes the proof. □

Definition 5.12 *Let C be a PRIM code block. Reg(C) is the maximum number of registers used in the execution of C.*

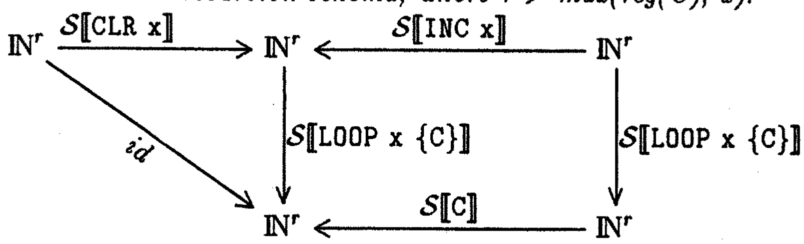
Definition 5.13 *We interpret a PRIM code block C as a Σ-arrow $\mathcal{S}$[[C]] as follows:*

SKIP $\mathbb{N}^r \xrightarrow{id} \mathbb{N}^r$ *for any* $r \in \mathbb{N}$*;*

INC x $\mathbb{N}^r \xrightarrow{id\times\ldots\times id\times s\times id\times\ldots\times id} \mathbb{N}^r$*, for any* $r > x$*,*
where the s here occurs in the xth position;

CLR x $\mathbb{N}^r \xrightarrow{id\times\ldots\times id\times 0o!\times id\times\ldots\times id} \mathbb{N}^r$*, for any* $r > x$*,*
where 0o! *is in the xth position;*

LOOP x {C} $\mathcal{S}$[[LOOP x {C}]] *is obtained from the following recursion schema, where* $r >$ *max(reg(C), x):*



$C_1;C_2$ $\mathcal{S}[[C_2]] \circ \mathcal{S}[[C_1]]$

Definition 5.14 *Given a PRIM program*

```
DEF P(x_1,...,x_r)
    C
RETURN(z)
```

let $n = max(r, reg(c), z)$. *We define* $\mathcal{S}[\![P]\!]$ *to be the following composition:*

$$\mathbb{N}^r \xrightarrow{id\times 0^{(n-r)}} \mathbb{N}^n \xrightarrow{\mathcal{S}[\![c]\!]} \mathbb{N}^n \xrightarrow{\pi_z} \mathbb{N}$$

$\mathcal{S}[\![P]\!]$ *is easily defined for programs which return a vector of natural numbers.*

Conjecture 5.15 $\mathcal{O}[\![P_1]\!] = \mathcal{O}[\![P_2]\!]$ *iff* $\mathcal{S}[\![P_1]\!] = \mathcal{S}[\![P_2]\!]$.

Example

Consider the following programs:

```
DEF P_1(x)
        INC z;
        INC z;
        LOOP z {INC x};
        CLR z;
RETURN(x)
```

```
DEF P_2(X)
        INC x;
        INC x;
RETURN(x)
```

We calculate $P_1(\alpha)$ and $P_2(\alpha)$ using the operational semantics:

$$\begin{aligned}
P_1(\alpha) &\rightsquigarrow \langle \texttt{INC z; INC z; LOOP z \{INC x\}; CLR z}, [x \mapsto \alpha]\rangle \\
&\rightsquigarrow \langle \texttt{INC z; LOOP z \{INC x\}; CLR z}, succ_z \circ [x \mapsto \alpha]\rangle \\
&\rightsquigarrow \langle \texttt{LOOP z \{INC x\}; CLR z}, succ_z \circ succ_z \circ [x \mapsto \alpha]\rangle \\
&\rightsquigarrow \langle \texttt{INC x; LOOP z \{INC x\}; INC z; CLR z}, pred_z \circ succ_z \circ succ_z \circ [x \mapsto \alpha]\rangle \\
&\rightsquigarrow \langle \texttt{LOOP z \{INC x\}; INC z; CLR z}, succ_x \circ succ_z \circ [x \mapsto \alpha]\rangle \\
&\rightsquigarrow \langle \texttt{INC x; LOOP z \{INC x\}; INC z; INC z; CLR z}, pred_z \circ succ_z \circ [x \mapsto \alpha + 1]\rangle
\end{aligned}$$

$$\rightsquigarrow \langle \texttt{LOOP z \{INC x\}; INC z; INC z; CLR z}, [\texttt{x} \mapsto \alpha + 2]\rangle$$
$$\rightsquigarrow [\texttt{x} \mapsto \alpha + 2]$$

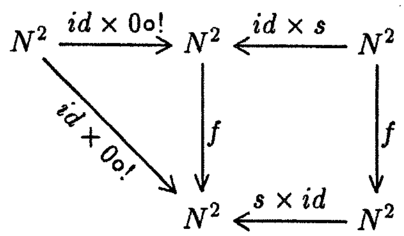
$$\mathrm{P}_2(\alpha) \rightsquigarrow \langle \texttt{INC x; INC x}, [\texttt{x} \mapsto \alpha]\rangle$$
$$\rightsquigarrow \langle \texttt{INC x}, \texttt{succ}_\texttt{x} \circ [\texttt{x} \mapsto \alpha]\rangle$$
$$\texttt{succ}_\texttt{x} \circ \texttt{succ}_\texttt{x} \circ [\texttt{x} \mapsto \alpha]$$
$$= [\texttt{x} \mapsto \alpha + 2]$$

So $\mathcal{O}[\![\mathrm{P}_1]\!] = \mathcal{O}[\![\mathrm{P}_2]\!]$.

$\mathcal{S}[\![\mathrm{P}_1]\!]$ is the following composite:

$$N \xrightarrow{id\times 0} N^2 \xrightarrow{id\times s} N^2 \xrightarrow{id\times s} N^2 \xrightarrow{f} N^2 \xrightarrow{\pi_1} N$$

where f is the unique arrow which makes the following commute:



$\mathcal{S}[\![\mathrm{P}_2]\!]$ is the composite $N \xrightarrow{s} N \xrightarrow{s} N$.

Now

$$\begin{aligned}
\mathcal{S}[\![\mathrm{P}_1]\!] &= \pi_1 \circ f \circ (id \times s) \circ (id \times s) \circ (id \times 0\circ!) \\
&= \pi_1 \circ (s \times id) \circ f \circ (id \times s) \circ (id \times 0\circ!) \\
&= \pi_1 \circ (s \times id) \circ (s \times id) \circ f \circ (id \times 0\circ!) \\
&= \pi_1 \circ (s \times id) \circ (s \times id) \circ (id \times 0\circ!) \\
&= \pi_1 \circ ((s \circ s) \times (0\circ!)) \\
&= s \circ s \\
&= \mathcal{S}[\![\mathrm{P}_2]\!]
\end{aligned}$$

Definition 5.16 *We write* $\mathrm{P}_1 \sim \mathrm{P}_2$ *if* $\mathcal{O}[\![\mathrm{P}_1]\!] = \mathcal{O}[\![\mathrm{P}_2]\!]$. $\sim$ *is clearly an equivalence relation.*

Conjecture 5.17 *Every* Σ*-arrow is the image under* $\mathcal{S}[\![\_]\!]$ *of an equivalnence class of* $\sim$.

It follows that arrows in PRIM and Σ-functions do not bijectively correspond. A PRIM program P is a *routine* for calculating the function represented in Σ by the arrow $\mathcal{S}[\![\mathrm{P}]\!]$ and the bijection is between routines with the same effect on Σ-arrows. We are effectively saying that we cannot prove in Σ that routines are different, so that this is not a useful framework for contemplating efficiency issues.

Lemma 5.18 $Hom_{\Sigma}(1, N) \cong N$

Proof Immediate from the construction of Σ. □

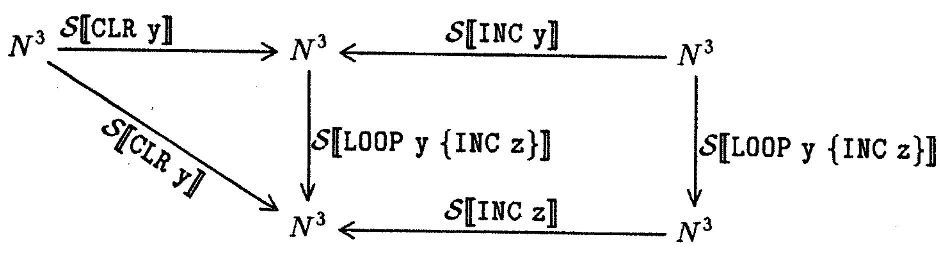
Defining simple arithmetic functions in PRIM is easy. Some simple examples which will be useful to us follow below. Note that in view of Lemma 5.10, we can call previously defined functions in our PRIM programs.

```
DEF add(x,y)
    z <— x;
    LOOP y { INC z };
RETURN(z)
```

We will write $x + y$ for add(x,y). In Σ, $\mathcal{S}[\![\mathrm{add}]\!]$ is the following arrow:

$$N^2 \xrightarrow{id\times 0} N^3 \xrightarrow{\mathcal{S}[\![\mathtt{z}\leftarrow\mathtt{x}]\!]} N^3 \xrightarrow{\mathcal{S}[\![\mathtt{LOOP\ y\ \{INC\ z\}}]\!]} N^3 \xrightarrow{\pi_z} N$$

where $\mathcal{S}[\![$`LOOP y {INC z}`$]\!]$ is defined recursivel;y as follows:



so add(x,y) is $1 \xrightarrow{\langle x,y\rangle} N \times N \xrightarrow{\mathcal{S}[\![\mathrm{add}]\!]} N$.

We can therefore define multiplication in PRIM:

```
DEF mult(x,y);
    CLR z;
    LOOP x {z <— z+y};
RETURN(z)
```

We will write xy for mult(x, y).

So $\mathcal{S}[\![\texttt{mult}]\!]$ is the following Σ-arrow:

$$N^2 \xrightarrow{id\times 0} N^3 \xrightarrow{\mathcal{S}[\![\texttt{CLR x}]\!]} N^3 \xrightarrow{\mathcal{S}[\![\texttt{LOOP x \{z}\leftarrow\texttt{z+y\}}]\!]} N^3 \xrightarrow{\pi_z} N$$

where we obtain $\mathcal{S}[\![$LOOP x {z ⟵ z + y}$]\!]$ recursively as before.

We can continue to define primitive arithmetic functions in this way:

```
DEF exp(x,y);
    CLR z;
    INC z;
    LOOP x { z ⟵ z.y};
RETURN(z)
```

We will write x^y for exp(x,y).

```
DEF pred(x);
    CLR y;
    CLR z;
    LOOP x { y ⟵ z;
             INC z };
RETURN(z)
```

```
DEF diff(x,y)
    z ⟵ x;
    LOOP y { z ⟵ pred(z)};
RETURN(z)
```

We will write $x \dot{-} y$ for diff(x,y).

```
DEF zero(n)
RETURN( ⟵ s(0) ∸ n)
```

```
DEF even(n)
    z ⟵ 1
    LOOP n {z ⟵ zero(z)}
RETURN(z)
```

```
DEF odd(n)
RETURN( zero(even(n)))
```

```
DEF eq(n,m)
RETURN( ⟵   zero ( (n ∸ m) + ( m ∸ n)))
```

```
DEF case(e,x,y)
RETURN( ⟵   zero(zero(e)).x + zero(e).y)
```

We can write case(e,x,y) as IF e THEN x ELSE y.

```
DEF gt(x,y)
RETURN( ⟵   zero (y ∸ x))
```

We write $x > y$ for gt(x,y).

```
DEF get(x,y)
RETURN( ⟵   x + s(0) > y)
```

We write $x \geq y$ for gte(x,y).

We know that each of these functions has an interpretation in Σ. If Σ' is any other Skolem category then there is a unique map of Skolem categories $F : \Sigma \longrightarrow \Sigma'$ and as F must preserve recursion, it is obvious how our PRIM programs will be interpreted in arbitary Skolem Categories.

Definition 5.19 *Given arrows* $X \xrightarrow{Q_1} N_C$, $X \xrightarrow{Q_2} N_C$ *in a Skolem category C and a PRIM function OP(x,y), we define* $\mathtt{OP}(\mathtt{Q}_1, \mathtt{Q}_2)$ *to be*

$$X \xrightarrow{\langle Q_1, Q_2 \rangle} N_C^2 \xrightarrow{F(S[\![\mathtt{OP}]\!])} N_C$$

This definition has obvious extensions to more input parameters and to N_C^k outputs.

Definition 5.20 *A* predicate *on an object X in a Skolem category is a map* $X \xrightarrow{Q} N$ *such that* $Q.Q = Q$*. We write* $P(X)$ *for the set of predicates on* X.

Lemma 5.21 *$P(X)$ has a Boolean Algebra structure with the following definitions:*

$(NOT\ Q)(x) = zero(Q(x))$
$(P\ AND\ Q)(x) = P(x).Q(x)$
$(P\ OR\ Q)(x) = zero(zero(P(x) + Q(x)))$

Proof Define

$$\top \equiv X \xrightarrow{!} 1 \xrightarrow{0} N \xrightarrow{s} N$$
$$\bot \equiv X \xrightarrow{!} 1 \xrightarrow{0} N$$

Then we can use our conjecture concerning interpretations to justify the obvious use of PRIM programs which yields all of the required equations. □

It follows that any Boolean combination of primitive recursive predicates is a primitive recursive predicate.

In passing we recall:

Definition 5.22 *If P is a predicate then the function* $\mu z < y.P$ *is the* bounded minimisation function, *given by:*

$$g(\mathbf{x}, y) = \mu z < y.P(\mathbf{x}, z) = \begin{cases} \text{the least } z < y \text{ with } P(\mathbf{x}, z) & \text{if such a z exists} \\ y & \text{otherwise} \end{cases}$$

We call $\mu z < y$ *the* bounded minimisation operator.

Lemma 5.23 *We can compute bounded minimisation operators in PRIM and hence interpret them in an arbitary Skolem theory.*

Proof If $N \times X \xrightarrow{Q} N$ is a predicate, define a function $N \times X \xrightarrow{f} N$ as follows:

```
DEF f(n,x)
    CLR y
    CLR m
    LOOP n { IF Q(m,x) AND y AND NOT eq(Q(0,x),1)
                        THEN z ⟵ m
             ELSE z ⟵ ; y
             y ⟵ z;
             INC m; }
RETURN(y)
```

Then $f(n,x) = \mu_{K<N}Q(K,x)$. □

Corollary 5.24 *$\mu z < y$ is primitive recursive.*

5.2 Constructing A.U.'s from Skolem Theories

5.2.1 Adding Decidable Subsets to E

Theorem 5.25 (Pairing Functions) *In any Skolem Category there is an isomorphism $N \times N \xrightarrow{\tau} N$. Any such τ is called a* pairing function.

Proof $\tau(x,y) = (2x+1)2^y - 1$ does the trick - this is a basic result from the theory of computability. We will refer to τ as *pair*. □

Definition 5.26 *We define the map $N \xrightarrow{\langle pr_1, pr_2 \rangle} N \times N$ to be the inverse of pair.*

Note that it does not matter *which* pairing function we choose to define $\langle pr_1, pr_2 \rangle$ so long as we stick to a single choice. Note that *pair* and *pr* are primitive recursive and can therefore be represented by maps in a Skolem Category.

Recall ([Mac 71, p.168]) that we can always construct the free monoid over an object A in $\mathcal{C}$ if $\mathcal{C}$ has denumerable coproducts. Clearly, E need not satisfy this condition. However, we shall see that the presence of recursive definitions gives us enough structure to perform the required constructions in a Skolem Theory.

Lemma 5.27 *Define $\mu : N \times N$ by $\langle n, m\rangle \mapsto 2^{e_n}m + n$, where e_n is the smallest integer for which $2^{e_n} > n$. Then $\langle N, \mu, 0\rangle$ is a monoid object in E.*

Proof Trivial. □

We can learn something significant about E by examining what is going on here in **Set**. If we express every integer in binary format then the multiplication operator applied to $\langle m, n\rangle$ simply returns

(Binary Representation of n) ++(Binary Representation of m)

In other words, we are simply building lists. Note that the multiplication operator has reversed the order of m, n. This allows us to adopt some suggestive notation for the following Lemma:

Definition 5.28 *Define $\eta : N \longrightarrow N$ in E by $n \mapsto 2^n$.*

Lemma 5.29 *In E, any n in N can be uniquely expressed as $\mu(\eta(n_h), n_t)$.*

Proof n_h will be the number of zeros on the right of the expansion of n as a binary number. Note that in non-initial Skolem Theories we may not have an initial digit - the expansion could be infinite, say - and this motivated the change in order of digits in the statement of Lemma 5.27.

On input n, the following PRIM routine returns on input n the first number $\geq n/2$:

```
DEF HALF(n)
    LOOP n {
         IF (NOT DONE) AND (2*C ≥ n) THEN {
            Z ⟵ C;
            DONE ⟵ 1;
         }
         INC C;
    }
RETURN (z)
```

Then the following routines determine n_h and n_t:

```
DEF HD(n)
    IF n ≠ 0 THEN {
        Z ⟵ n;
        LOOP n {
            IF EVEN(Z) THEN{
                INC C;
                Z ⟵ HALF(Z);
            }
        }
    }
RETURN(C)
```

```
DEF TL(n)
    H ⟵ HD(n);
    LOOP n{
        IF n = μ(η(H),T) THEN DONE ⟵ 1;
        IF NOT DONE THEN INC T;
    }
RETURN(T)
```

The following routine therefore extracts n_h and n_t for a given n:

```
DEF GET-VECT(n)
    H ⟵ HD(n);
    T ⟵ TL(n);
    Z ⟵ PAIR(H,T);
RETURN(Z)
```

This completes the proof. □

We now apply the above to prove the following:

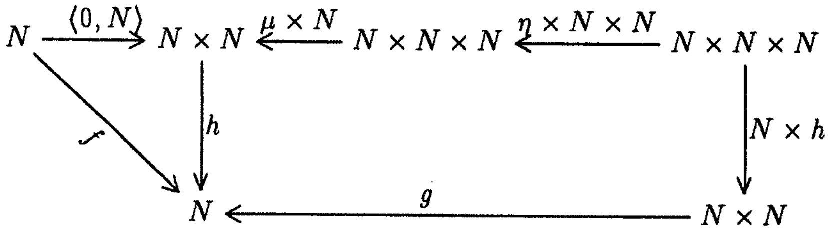
Lemma 5.30 *N is the list object over N in E.*

Proof

Define $cons : N \times N \longrightarrow N : \langle n, m\rangle \mapsto \mu(\eta(n), m)$

$\epsilon : N \longrightarrow N : n \mapsto 0$

Suppose that $N \xrightarrow{f} N \xleftarrow{g} N \times N$. We claim that there is a unique $N \times N \xrightarrow{h} N$ such that the following commutes:



ie. that h is well defined by the following recursion schema:

$h(0, n) = f(n)$

$h(\mu(\eta(m), ms), n) = g(m, h(ms, n))$

Intuitively, we would expect this to work since h is defined at n in terms of its value at number $ms < n$ and because we can perform recursion. We actually define h recursively by retaining *all* earlier values of h and then accessing them as required:

```
DEF h(n,m)
    z=PAIR(f(m),0)
    i <-- 1
    LOOP n {
         H <-- HD(n);
         T <-- TL(n);
         z1 <-- z;
         j <-- 1;
         LOOP (i-T) {
              z1 <-- pr1(z1;
              INC j;
         }
         z <-- pair(g(H,z1),z);
         INC i;
```

```
    }
RETURN(pr_1(z))
```

h clearly satisfies the recursion schema, from a straightforward examination of the PRIM program. Now suppose that h' also satisfies the recursion equations. We want to use induction to show that $h' \equiv h$. However, we cannot yet use Lemma 2.35 as we have not proved yet that N *is* a list object and so we must induct on the size of n and employ Lemma 2.31.

Define $k(0,n) = h(0,n)$

$$k'(0,n) = h'(0,n)$$
$$k(s(m),n) = pair(h(s(m),n), k(m,n))$$
$$k'(s(m),n) = pair(h'(s(m),n), k'(m,n))$$

Then $k(0,n) = k'(0,n) = f(n)$.

Suppose that the two functions are equal at m.

$k(s(m),n) = pair(g(s(m)_h, h(s(m)_t, n)), k(m,n))$.

But $h(s(m)_t, n) = \texttt{LOOP}(n - s(m)_t)\{pr_1(k(m,n))\}$

$= \texttt{LOOP}(n - s(m)_t)\{pr_1(k'(m,n))\}$, by IH

$= h'(s(m)_t, n)$

So $k(s(m),n) = k'(s(m),n)$ and hence by Lemma 2.31, $k \equiv k'$.

Taking pr_1 of both sides, we get $h \equiv h'$, as required. □

This result therefore allows us to prove equality of functions in E by list induction as in Lemma 2.35:

If $f(0,x) = g(0,x)$

and $f(n,x) = g(n,x) \Rightarrow f(\mu(\eta(n_1),n),x) = g(\mu(\eta(n_1),n),x)$

then $f \equiv g$

In passing, let us briefly consider the following PRIM routine:

```
DEF GET-FULL-VECT(n)
    M-LEN = μ_{k<n}(2^k > n)    /* Max length of the rep.  of n */
```

```
    J ⟵ n
    J ⟵ ST(n)
    V-LEN ⟵ 1
    IF (pr₂(J) = 0) THEN (DONE ⟵ 1)
    LOOP M-LEN {
         IF (NOT DONE) THEN {
            J ⟵ PAIR(PAIR(pr₁(J),HD(pr₂(J))), TL(pr₂(J)));
            INC V-LEN;
            IF pr₂(J)=0 THEN { DONE ⟵ 1;
                               V-LEN ⟵ V-LEN ∸ 1;
                               J ⟵ pr₁(J)
            }
         } }
    RV ⟵ PAIR(J, V-LEN);
RETURN(RV)
```

In **Set** this would give us

$$pair(n_1, pair(n_2, \ldots, pair(n_{r-1}, n_r)\ldots))$$

where $n = \mu(\eta(n_1), \ldots, \eta(n_r))$ is a finite length representation for n in **Set**. The routine converges in the sence of [Jay 93] - in other words, there is a computable upper bound $\mathtt{M-LEN}$ in N to the number of iterations required to produce a fully expanded vector. The point here is that $\mathtt{M-LEN}$ exists *internally* - in specific cases, the completed representation could be infinite.

Lemma 5.31 *E has free monoids.*

Proof This is a trivial consequence of 5.30 and 2.23. □

We now proceed towards the definition of an Arithmetic Universe. In view of the above pairing result, we henceforth define arrows in a Skolem Theory using only the objects 1 and N. Recall the following definition in **Set**:

Definition 5.32 *Suppose that* $\mathbb{N} \xrightarrow{Q} \mathbb{N}$ *is a predicate in the Skolem Theory of primitive recursive functions,* Σ_{PRIM}. *Then* Q *determines a set*

$\{n \in \mathbb{N} | Q(n) = 1\}$, *which we will call the* extension *of* Q. *We call a subset of* $\mathbb{N}$ decidable *if it is the extension of a predicate* Q.

In other words, $X \subseteq \mathbb{N}$ is decidable iff its characteristic function is primitive recursive. We want to extend our universe to include decidable subsets of $\mathbb{N}$ and primitive recursive functions between them. In a general Skolem Theory we need not have an easy notion of global elements as in Σ_{PRIM}, so instead we use the correspondence between predicates and decidable subsets, and we take the arrows to be suitably defined equivalence classes:

Definition 5.33 *Suppose that* E *is a Skolem Category. Define the category* $\check{E}$ *as follows:*

Objects *Predicates* $N \xrightarrow{A} N$ *of* E*;*

Arrows *Suppose that* A,B *are* $\check{E}$*-objects. Define* $Arr_{A,B} = \{N \xrightarrow{f} N | A \leq B \circ f\}$. *Define a relation* $\sim$ *on* $Arr_{A,B}$ *by* $f_1 \sim f_2$ *if* $A \leq eq(f_1, f_2)$. *It is clear that* $\sim$ *is an equivalence relation on* $Arr_{A,B}$. *The arrows* $A \longrightarrow B$ *in* $\check{E}$ *are* $\sim$*-equivalence classes.*

So $Arr_{A,B}$ consists precisely of thise arrows which map A into B. Each equivalence class of $\sim$ consists of arrows which agree on A - in Σ_{PRIM} these would be those arrows whose restrictions to A agreed -ie. the primitive recursive functions $A \longrightarrow B$.

Lemma 5.34 *Composition of arrows in* $\check{E}$ *is well-defined.*

Proof Suppose that A,B,C are $\check{E}$-objects with $[f_1] = [f_2] : A \longrightarrow B$ and $[g_1] = [g_2] : B \longrightarrow C$ are $\check{E}$-arrows. We need to show that $g_1 \circ f_1] : A \longrightarrow C$, which is trivial, since $A \leq B \circ f_1 \leq C \circ g_1 \circ f_1$ and that $[g_1 \circ f_1] = [g_2 \circ f_2]$, which follows from the identities $eq(g_1 \circ f_1, g_2 \circ f_2) \geq eq(g_1, g_2) \circ eq(f_1, f_2) \geq B \circ A \geq A$. □

In the case where E is Σ_{PRIM}, we clearly have Σ_{PRIM} as a full subcategory of $\check{\Sigma}_{PRIM}$ - $\check{\Sigma}_{PRIM}$ consists of Σ_{PRIM} augmented with decidable subsets of $\mathbb{N}$ and the primitive recursive functions on those sets. We will establish this in the general case:

Definition 5.35 *We define* $TRUE$ *to be the map*

$$N \xrightarrow{!} 1 \xrightarrow{0} N \xrightarrow{s} N$$

Note that in Σ_{PRIM}, $TRUE = \chi_{\mathbb{N}}$.

Lemma 5.36 *If E is any Skolem Theory then E can be embedded as a full subcategory of $\check{E}$.*

Proof We identify N with $TRUE$. Define $\zeta \equiv N \xrightarrow{!} 1 \xrightarrow{0} N$, define $\check{1}$ to be $eq(\_, \zeta) : N \longrightarrow N$. (ie $N \xrightarrow{\_\times\zeta} N \times N \xrightarrow{S[\circ q]} N$).

Then if $N \xrightarrow{P} N$ is an $\check{E}$-object we have $P \leq eq(\zeta, \zeta)$, so $[\zeta] : P \longrightarrow \check{1}$ in $\check{E}$.

If $[\zeta'] : P \longrightarrow \check{1}$ then $P \leq eq(\zeta, \zeta')$, by definition of $Arr_{A,B}$ above. However this is the same as saying that $[\zeta] = [\zeta']$, so $[\zeta] : P \longrightarrow \check{!}$ is unique and $\check{1}$ is the terminal object in $\check{E}$.

We can embedd E into $\check{E}$ using F, defined on objects by $F(1) = \check{1}$ and $F(N) = TRUE \stackrel{def}{=} \check{N}$ and on arrows by $F(f) = [f]$.

We know that for any predicate Q, $Q \leq \check{N} \circ f$, where $N \xrightarrow{f} N$ is any arrow; in particular, $\check{N} \leq \check{N} \circ f$, so that $\check{N} \xrightarrow{[f]} \check{N}$. In general, we clearly have $X \xrightarrow{f} Y$ in $E \Rightarrow \check{X} \xrightarrow{[f]} \check{Y}$ in $\check{E}$.

$F(f \circ g) = [f] \circ [g]$ by the previous lemma and F trivially respects identity maps, so F *is* a functor.

Any arrow $[f]$ in $\check{E}$ is the image of f under F, so F is full. □

Lemma 5.37 $\zeta \equiv N \xrightarrow{!} 1 \xrightarrow{0} N$ *is initial in* $\check{E}$.

Proof This is easy: for any predicate Q, $\zeta \leq Q$ so any $f : N \longrightarrow N$ maps $\zeta \longrightarrow P$, where P is a predicate; moreover, if $f_1, f_2 : N \longrightarrow N$ are any arrows we have $\zeta \leq eq(f_1, f_2)$, so that $[f_1] = [f_2]$ on $Arr_{A,B}$, which gives us uniqueness. □

Regarding A and B as the extensions which they represent in **Set**, we have $(n, m \in A) \Rightarrow (f(n) = f(m) \Rightarrow n = m)$ iff f is mono. The following Lemma generalises this statement:

Lemma 5.38 $A \xrightarrow{[f]} B$ *in* $\check{E}$ *is monic iff* $A(n).A(m).eq(f(n), f(m)) \leq eq(n, m)$.

Proof 'Only If' is easy: $[f]$ monic $\Rightarrow eq(f(n), f(m)) \leq eq(n, m)$, which yields the result.

'If' is also fairly strightforward: Suppose that

$$A(n).A(m).eq(f(n), f(m)) \leq eq(n, m)$$

If $f \circ n = f \circ m$ then $A(n).A(m).eq(f(n), f(m)) \leq eq(n,m)$. If $f \circ n = f \circ m$ then $A(n).A(n) \leq eq(n,m)$. Now n, m are $\breve{E}$-arrows, so $C \leq A \circ n$ and $C \leq A \circ m$. ie $C^2 \equiv C \leq A(n).A(m) \leq eq(n,m)$. If C is identically zero then $C \equiv \zeta$, so C is initial by lemma 5.37 and $n = m$. If C is not identically zero then $eq(n,m) = s(0)$ so $n = m$. □

Corollary 5.39 *Every $\breve{E}$-object is a subobject of $\breve{N}$.*

Proof If $N \xrightarrow{A} N$ then since $A \leq \breve{N} \circ id$, we have $[id_N] : A \longrightarrow \breve{N}$.

Since $A(N).A(M) \leq s(0)$, $A(n)A(m) \leq eq(n,m)$, so $[id_N]$ is monic. □

This is directly analogous to the statement in **Set** that every predicate deterines a subset of $\mathbb{N}$.

Lemma 5.40 *$\breve{E}$ has finite products.*

Proof We have shown that $eq(\_, \zeta)$ is the terminal object in $\breve{E}$. For products over non-empty diagrams, given predicates A,B in $\breve{E}$, we will use $(A \check{\times} B)$ to denote the $\breve{E}$-object $mult \circ (A \times B) \circ \langle pr_1, pr_2 \rangle$, where $\langle pr_1, pr_2 \rangle : N \longrightarrow N^2$ is the inverse of the pairing function $pair : N^2 \longrightarrow N$. In terms of generalised elements, $(A \check{\times} B)(n) = A(pr_1(n)).B(pr_2(n))$.

Now $A \circ pr_1 \geq mult \circ (A \times B) \circ \langle pr_1, pr_2 \rangle$, so $[pr_1] : A \check{\times} B \longrightarrow A$ in $\breve{E}$. We claim that $A \xleftarrow{[pr_1]} A \times B \xrightarrow{[pr_2]} B$ is a product diagram in $\breve{E}$. Suppose that $B \xleftarrow{f_2} C \xrightarrow{f_1} A$ and write $\langle f_1 \check{,} f_2 \rangle$ for the equivalence class of $N \xrightarrow{\Delta} N \times N \xrightarrow{f_1 \times f_2} N \times N \xrightarrow{pair} N$, where Δ is the usual diagonal map. Then since $C \leq B \circ f_2$ and $C \leq A \circ f_1$, we have that $C \leq A \check{\times} B \circ \langle f_1 \check{,} f_2 \rangle = mult \circ (A \times B) \circ (f_1 \times f_2) \circ \Delta$, so $\langle f_1 \check{,} f_2 \rangle$ is a well defined arrow $C \longrightarrow A \check{\times} B$ in $\breve{E}$, $pr_1 \circ \langle f_1 \check{,} f_2 \rangle = f_1$ and $pr_2 \circ \langle f_1 \check{,} f_2 \rangle = f_2$. If $[g] : C \longrightarrow A \check{\times} B$ with $[pr_2] \circ [g] = [f_2]$ and $[pr_1] \circ [g] = [f_1]$ then since $[pr_1]$ and $[pr_2]$ are jointly monic, $[g] = \langle f_1 \check{,} f_2 \rangle$. □

Lemma 5.41 *$\breve{E}$ has equalisers.*

Proof If $A \underset{[v]}{\overset{[u]}{\rightrightarrows}} B$ is an $\breve{E}$-diagram then $C(n) = A(n).eq(u(n), v(n))$ with $C \xrightarrow{[id_N]} A \underset{[v]}{\overset{[u]}{\rightrightarrows}} B$ is an equaliser diagram, since $C \leq A \circ id = A$ and

$C \leq eq \circ \langle u, v \rangle$, so $[u] \circ [id_N] = [v] \circ [id_N]$; if $D \xrightarrow{[f]} A$ with $D \leq eq(u \circ f, v \circ f)$ then

$$\begin{aligned} D = D^2 &\leq mult \circ \langle D, eq \circ \langle u \circ f, v \circ f \rangle \rangle \\ &\leq mult \circ \langle A \circ f, eq \circ \langle u \circ f, v \circ f \rangle \rangle \\ &= C \circ f \end{aligned}$$

so $D \xrightarrow{[f]} C$ and clearly, $D \xrightarrow{[f]} A = D \xrightarrow{[f]} C \xrightarrow{[id_N]} A$.

If $D \xrightarrow{[g]} C$ with $id_N \circ [g] = [f]$ then $[g] = [f]$, which gives us uniqueness.

□

This result is really very obvious: in **Set**, C is the characteristic function of $\{x \in A | u(x) = v(x)\}$.

Lemma 5.42 *$\check{E}$ has finite coproducts.*

Proof We have already seen in Lemma 5.37 that $\check{E}$ has initial object $\zeta \equiv N \xrightarrow{!} 1 \xrightarrow{0} N$. For non-empty co-products, suppose $A, B \in Ob(\check{E})$.

Define $(A \dot{+} B)(n) = \begin{cases} A(n/2) & \text{if } n \text{ is even} \\ B((n-1)/2) & \text{if } n \text{ is oddd} \end{cases}$

Define $i_1 : N \longrightarrow N : n \mapsto 2n$
$i_2 : N \longrightarrow N : n \mapsto 2n + 1$

Then it is easy to show that

$$A \xrightarrow{[i_1]} A \dot{+} B \xleftarrow{[i_2]} B$$

is a coproduct diagram. □

In general, $\check{E}$ need not have coequalisers.

Definition 5.43 *A* split epic *in a category C is an arrow $f : A \longrightarrow B$ which has a right inverse.*

Clearly, split epics are epic. In **Set**, the Axiom of Choice is equivalent to the statement that epics split.

Lemma 5.44 *Every arrow $A \xrightarrow{[f]} B$ in $\check{E}$ has a factorisation into a split epic followed by a monic.*

Proof Define a predicate I on generalised elements by

$$I(n) = A(n).eq(1, \Sigma_{k=0}^{n} A(k).eq(f(k), f(n)))$$

In other words, I(n) iff A(n) AND (n is the first integer in the extension of A whose value under f is $f(n)$).

Define $N \xrightarrow{p} N$ in terms of generalised elements by

$$p(n) = \mu_{k<n}(A(k).eq(f(k), f(n)))$$

Then

$$\begin{aligned}(I \circ p)(n) &= A(\mu_{k<n}(A(k).eq(f(k), f(n)))).eq(1, \Sigma_{k=0}^{n} A(k).eq(f(k), f(p(n))))\\ &= A(\mu_{k<n}(A(k).eq(f(k), f(n))))\\ &= \begin{cases} A(n) & \text{there is no } k < n \text{ with A(k) and } f(k) = f(n) \\ 1 & \text{otherwise} \end{cases}\end{aligned}$$

So $[p] : A \longrightarrow 1$. Moreover, $I \leq A \leq B \circ f$, so $[f] : I \longrightarrow B$. We therefore have that $A \xrightarrow{[p]} I \xrightarrow{[f]} B$.

It is trivially true that $eq(f \circ p, f) \geq A$, so $[f] = [f] \circ [p]$.

(Note that $I \xrightarrow{[f]} B$ is a different map from $A \xrightarrow{[f]} B$; in the former, $[f]$ is a class of maps which only have to agree on the extension on I, which is a subset of the extension of A).

We have that $I \leq eq(p \circ id_N, id_N)$, so $[p] \circ [id_N] = [id_N]$, and

$$A \xrightarrow{[p]} I \xrightarrow{[f]} B$$

is a split epic/monic factorisation of $[f]$, where $A \xrightarrow{[p]} I$ has a right inverse $I \xrightarrow{[id_N]} A \xrightarrow{[p]} I$. □

Recall that the usual proof in **Set** of this result involves taking I to be $f(A)$ and then using the Axiom of Choice to find a right inverse for f. In our proof, I is a subset which consists for every $x \in f(A)$ of precisely one element of $f^{-1}(\{x\})$ (the least one); we can calculate I without recourse to the Axiom of Choice because f has a countable well-ordered domain. We will employ this approach again in Theorem 5.74.

Definition 5.45 *An arrow $A \xrightarrow{q} Q$ in a category C is* surjective *if the smallest subobject of Q through which q factors is id_Q.*

Definition 5.46 *A* surjective image *of* $A \xrightarrow{f} B$ *is an image factorisation* $A \xrightarrow{q} Q \rightarrowtail^{i} B$ *with* q *surjective.*

Lemma 5.47 $A \xrightarrow{f} B$ *has surjective image* $A \longrightarrow A' \xhookrightarrow{i} B$ *iff* i *is the smallest subobject of* B *through which* f *factors.*

Proof 'Only if'. If $A \xrightarrow{q} Q \xhookrightarrow{i} B$ is a surjective image of f then if f factors through $Q' \xhookrightarrow{i'} B$, q must factor through $i \cap i'$, so $i \cap i' = id_Q$.

'If'. If i is the smallest subobject of B through which f factors then q factors through $Q' \xhookrightarrow{q'} Q$, f factors through $i \circ q' \subseteq i$ so $q' = id_Q$. □

Corollary 5.48 *Surjective images are equivalent up to a commuting isomorphism.*

Definition 5.49 *A catgory* C *has image factorisation if every* $A \xrightarrow{f} B$ *in* C *has a surjective image.*

Lemma 5.50 $A \xrightarrow{[p]} I \xrightarrow{[f]} B$ *is the image factorisation of* $A \xrightarrow{[f]} B$.

Proof Note that $I(n).I(m).eq(f(n), f(m)) \leq eq(n, m)$; this is trivially true if $eq(f(n), f(m)) = 0$ and if $f(n) = f(m)$ and $m \neq n$ then at least one of $I(n)$ and $I(m)$ is zero, so $I \xrightarrow{[f]} B$ is monic.

If $A \xrightarrow{[f]} B$ factors through $C \xhookrightarrow{[g]} B$ via $A \xrightarrow{[h]} C$ then $I \xrightarrow{[f]} B$ factors through $C \xhookrightarrow{[g]} B$ via $[h] \circ [id_N]$ so $[f] : I \longrightarrow B$ is the minimal subobject through which $[f] : A \longrightarrow B$ factors. □

Definition 5.51 *A surjection* $A \xrightarrow{q} Q$ *is* stable *if for every* $s : Q' \longrightarrow Q$, *the pullback of* q *along* s *is surjective.*

Note that since pullbacks of monics are monic, we could equally as well have defined this as preservation of image factorisation by pullback.

Lemma 5.52 *If* $A \xrightarrow{[f]} \alpha$ *is a surjection in* $\check{e}$ *then if* $C(n_0) = TRUE$ *there is a map* $N \xrightarrow{n} N$ *such that* $A(n).eq(f(n), n_0) = TRUE$.

Proof Suppose $C(n_0) = TRUE$ and $A(\_).eq(f(\_), n_0) \equiv FALSE$.

Define a predicate C_0 by $C_0(n) = C(n).not(eq(n, n_0))$. Then

$$\begin{aligned} C_0 \circ f)(n) &= C(f(n)).not(eq(f(n), n_0)) \\ &\geq C(f(n).A(n) \\ &\geq A(n).A(n) \\ &= A(n) \end{aligned}$$

So $A \xrightarrow{[f]} C_0$ is an $\breve{E}$-arrow and $A \xrightarrow{[f]} C$ factors through C_0 as $A \xrightarrow{[f]} C_0 \xrightarrow{[id_N]} C$, so $C \xrightarrow{[id_N]} C$ should factor through $C_0 \xrightarrow{[id_N]} C$, since $A \xrightarrow{[f]} C$ is surjective. This is a contradiction, so our result follows. □

Lemma 5.53 *In $\breve{E}$, any surjection is a split epic.*

Proof Any surjection s has a factorisation $m \circ e$, where m is a split epic and m is monic. By the definition of surjectivity, m is iso. □

Lemma 5.54 *Image factorisation in $\breve{E}$ is stable under pullback.*

Proof If e is a split epic with right inverse p then $e \circ p \circ g = g = g \circ id_B$ so there is a map $\langle id, p \circ g\rangle : B \longrightarrow g^{-1}(A)$ with $g^{-1}(e) \circ \langle id, p \circ g\rangle = id_B$. □

Definition 5.55 *A* regular category *is one which has finite limits and image factorisation which is stable under pullback.*

Lemma 5.56 *Coproducts in $\breve{E}$ are stable under pullback.*

Proof Suppose that the following is a pullback diagram:

$$\begin{array}{ccccc} A & \xrightarrow{[i_1]} & A+B & \xleftarrow{[i_2]} & B \\ \uparrow{\scriptstyle [pr_2]} & & \uparrow{\scriptstyle [g]} & & \uparrow{\scriptstyle [pr_2]} \\ X_A & \xrightarrow{[pr_1]} & X & \xleftarrow{[pr_1]} & X_B \end{array}$$

Then

$$X_A(n) = A(pr_2(n)).X(pr_1(n)).eq(g(pr_1(n)), 2.pr_2(n))$$
$$X_B(n) = B(pr_2(n)).X(pr_1(n)).eq(g(pr_1(n)), 2.pr_2(n) + 1)$$

Now suppose that $X_A \xrightarrow{[f_1]} C \xleftarrow{[f_2]} X_B$.

Define $n' = \begin{cases} pair(n, g(n)/2) & g(n) \text{ even} \\ pair(n, (g(n)-1)/2) & g(n) \text{ odd} \end{cases}$

Define $f : N \longrightarrow N : n \mapsto \begin{cases} f_1(n') & g(n) \text{ even} \\ f_2(n') & g(n) \text{ odd} \end{cases}$

By assumption,

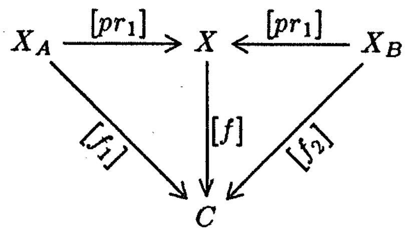
$$\begin{aligned} X(n) &\leq ((A \dot{+} B) \circ g)(n) \\ &= \begin{cases} A(g(n)/2) & g(n) \text{ even} \\ B((g(n)-1)/2) & g(n) \text{ odd} \end{cases} \end{aligned}$$

$$\begin{aligned} \text{So } X(n) &\leq \begin{cases} X_A(n') & g(n) \text{ even} \\ X_B(n') & g(n) \text{ odd} \end{cases} \\ &\leq \begin{cases} (C \circ f_1)(n') & g(n) \text{ even} \\ (C \circ f_2)(n') & g(n) \text{ odd} \end{cases} \\ &\leq (C \circ f)(n) \end{aligned}$$

So $X \xrightarrow{[f]} C$ in $\check{E}$.

$$\begin{aligned} X_A(n) = 1 &\Rightarrow g(pr_1(n)) \text{ is even} \\ &\Rightarrow (f \circ pr_1)(n) = f_1(pair(pr_1(n), g(pr_1(n))/2)) \\ &\qquad\qquad\quad = f_1(pair(pr_1(n), pr_2(n))) \end{aligned}$$

It follows that $X_A \leq eq(f \circ pr_1, f_2)$, so the following diagram commutes:



For uniqueness, observe that if $[h]$ also makes this diagram commute then

$$X(n) = 1 \Rightarrow \begin{cases} X_A(n') = 1 & g(n) \text{ even} \\ X_N(n') = 1 & g(n) \text{ odd} \end{cases}$$

$$\Rightarrow \begin{cases} (h \circ pr_1)(n') = f_1 & g(n) \text{ even} \\ (h \circ pr_2)(n') = f_1 & g(n) \text{ odd} \end{cases}$$

So $X \leq eq(h, f)$. $\square$

Lemma 5.57 *Coproducts in $\breve{E}$ are disjoint.*

Proof Suppose the following is a pullback diagram:

$$\begin{array}{ccc} A & \xrightarrow{i_1} & A + B \\ \uparrow{\scriptstyle [pr_2]} & & \uparrow{\scriptstyle [i_2]} \\ C & \xrightarrow{[pr_1]} & B \end{array}$$

Then using the previous lemmas and the elementary properties of pullbacks we have that

$$\begin{aligned} C(n) &= A(pr_2(n)).B(pr_1(n)).eq(2.pr_2(n), 2.pr_1(n) + 1) \\ &= \zeta, \text{ as required} \end{aligned}$$

$\square$

Theorem 5.58 *$\breve{E}$ has list objects.*

Proof Suppose that $N \xrightarrow{A} N$ is an $\breve{E}$-object. Recall that $(N, \mu, 0)$ as defined in Lemma 5.30 is the free monoid over N in E.

By the freeness of $(N, \mu, 0)$, there is a unique arrow $\bar{A} : N \longrightarrow N$, which is a monoid homomorphism from $(N, \mu, 0)$ to $(N, \cdot, 0)$ with $\bar{A} \circ \eta = A$.

We prove that $\bar{A}$ is a predicate, using the induction principle of 2.35 applied to $\bar{A}.\bar{A}$ and the constant function 1.

1. $\bar{A}(0) = s(0)$, since $\bar{A}$ preserves units and so $\bar{A}(0).\bar{A}(0) = 1$.

2. If $\bar{A}(n).\bar{A}(n) = 1$ then

$$\begin{aligned} \bar{A}(\mu(\eta(n_1), n)).\bar{A}(\mu(\eta(n_1), n)) &= A(n_1).\bar{A}(n).A(n_1).\bar{A}(n) \\ &= 1 \text{ by the IH and since } A \text{ is a predicate.} \end{aligned}$$

So $\bar{A}.\bar{A} = 1$.

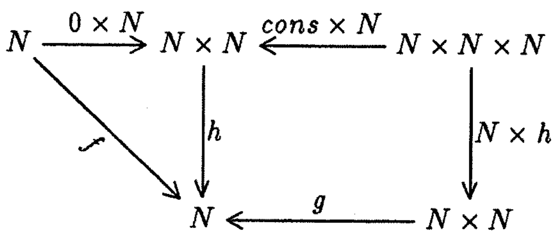
Define $\bar{\mu} : N \longrightarrow N \equiv N \xrightarrow{pair} N \times N \xrightarrow{\mu} N$.

$$\begin{aligned}\text{Then } (\bar{A}\check{\times}\bar{A})(n) &= \bar{A}(pr_1(n)).\bar{A}(pr_2(n)) \\ &= \bar{A}(\mu(pr_1(n), pr_2(n))) \\ &= (\bar{A} \circ \mu)(n)\end{aligned}$$

So $\bar{A}\check{\times}\bar{A} \xrightarrow{[\bar{\mu}]} \bar{A}$ in $\check{E}$.

Since $\bar{A}(0) = 1$, $TRUE \xrightarrow{[0]} \bar{A}$ in $\check{E}$.

Now suppose that $B \xrightarrow{[f]} C \xleftarrow{[g]} A \times C$. By Lemma 5.30, since N is the list object over N in E there is a unique map h which makes the following commute:



We use list induction again to show that $[h] : \bar{A} \times B \longrightarrow C$. ie that $\bar{A}(n_1).B(n_2) = 1 \Rightarrow C(h(pair(n_1, n_2)))$. We perform the induction on n_1.

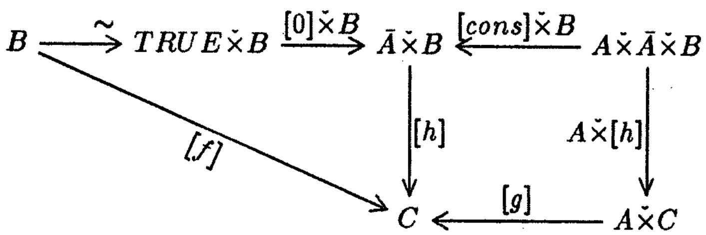
Base Case: If $n_1 = 0$ then $\bar{A}(n_1).B(n_2) = 1 \Rightarrow B(n_2) = 1 \Rightarrow C(f(n)) = 1$, since $B \xrightarrow{[f]} C$ and so by definition of h, $C(h(pair(0, n))) = 1$.

Inductive Step: Suppose that $\bar{A}(\mu(\eta(n_h), n_t), n) = 1 = B(n)$.

$$\begin{aligned}\text{Then } C(h(pair(\mu(\eta(n_h), n_t), n))) &= C(g(n_h, h(pair(n_t, n)))) \\ &= 1\end{aligned}$$

Since by the IH, $\bar{A}(n_t) = 1 = B(n) \Rightarrow C(h(pair(n_t, n)))$.

So the following diagram commutes in $\check{E}$:



For uniqueness, suppose that $[k] : \bar{A}\check{\times}B \longrightarrow C$ also makes the diagram commute. We need to show that $\bar{A}\check{\times}B \leq eq(h, k)$.

ie that $\bar{A}(n_1).B(n_2) = 1 \Rightarrow h(pair(n_1, n_2)) = k(pair(n_1, n_2))$.

Once again, we proceed by list induction on n_1.

Base Case: $n_1 = 0$. Then by definition, $\bar{A}(0).B(n_2) = 1 \Rightarrow B(n_2) = 1$. So

$$h(pair(0, n_2)) = f(n_2) = k(pair(0, n_2))$$

by the commutativity of the left triangle.

Inductive Step: Assume

$$\bar{A}(n_1).B(n_2) = 1 \Rightarrow h(pair(n_1, n_2)) = k(pair(n_1, n_2))$$

and suppose that $\bar{A}(\mu(\eta(n), n_1)).B(n_2) = 1$. Then $A(n).\bar{A}(n_1).B(n_2) = 1$ and we can deduce that $\bar{A}(n_1).B(n_2) = 1$.

$$\begin{aligned} \text{Then } h(pair(\mu(\eta(n), n_1), n_2)) &= g(pair(n, h(pair(n_1, n_2)))) \\ &= g(pair(n, k(pair(n_1, n_2)))) \\ &= k(pair(\mu(\eta(n), n_1), n_2)) \end{aligned}$$

So $\bar{A}\check{\times}B \leq eq(h, k)$, as desired. □

Corollary 5.59 *$\check{E}$ has free monoids.*

Proof This follows immediately from Lemma 2.23 and Theorem 5.58. $(\bar{A}, \bar{\mu}, [0])$ is the free monoid over A, with $[\eta] : A \longrightarrow \bar{A}$ as in definition 5.28. □

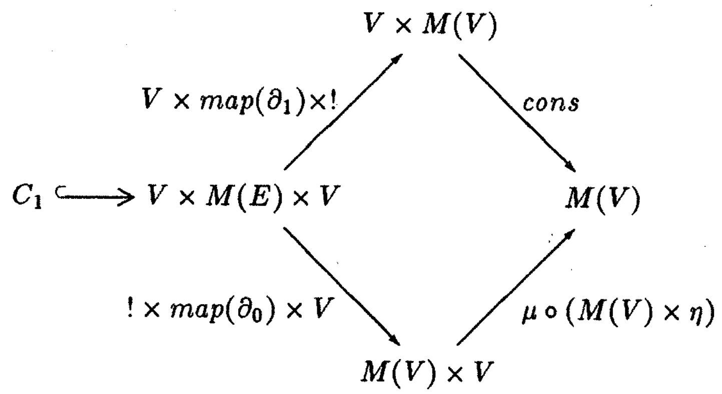
Theorem 5.60 *There are free category objects over graph objects in $\check{E}$.*

Proof Suppose that $\mathbf{G} \equiv E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V$ is a graph object in $\check{E}$.

Let $M(E)$ be the free monoid over E, which is the list object over E.

Define $map(\partial_1)$ to be the unique arrow $M(E) \longrightarrow M(V)$ which comes from the recursion data $1 \xrightarrow{\epsilon_V} M(V) \xleftarrow{cons} V \times M(V) \xleftarrow{\partial_1 \times M(V)} E \times M(V)$. Define $map(\partial_0)$ analogously.

Define C_1 to be the following equaliser:



So in **Set**,

$$C_1 = \{\langle v_1, [e_1, \ldots, e_n], v_2\rangle | v_1 = \partial_0 e_1, v_2 = \partial_1 e_n, \partial_1 e_i = \partial_0 e_{i+1}\} \cup \Delta_V$$

Define $d_0 \equiv C_1 \hookrightarrow V \times M(E) \times V \xrightarrow{\pi_1} V$

$d_1 \equiv C_1 \hookrightarrow V \times M(E) \times V \xrightarrow{\pi_3} V$

Define $C_1 \times_{C_0} C_1 \xrightarrow{m} V \times M(E) \times V$, where the pullback is of d_0 along d_1, by

$$\begin{aligned} m \;\equiv\; & C_1 \times_{C_0} C_1 \hookrightarrow (V \times M(E) \times V)^2 \xrightarrow{\langle \pi_1, \pi_2\rangle \times \langle \pi_2, \pi_3\rangle} \\ & V \times M(E) \times M(E) \times V \xrightarrow{V \times \mu \times V} V \times M(E) \times V \end{aligned}$$

We want to show that m equalises the above diagram and hence that it induces a map $C_1 \times_{C_0} C_1 \longrightarrow C_1$.

This can be done painfully via a huge commuting diagram, or alternatively using generalised elements. For $\langle\langle v_1, w_1, v_2\rangle, \langle v_2, w_2, v_3\rangle\rangle \in_T C_1 \times_{C_0} C_1$,

$$\begin{aligned} cons(v_1, map(\partial_1)(\mu(w_1, w_2))) &= cons(v_1, \mu(map(\partial_1)w_1, map(\partial_1)w_2)) \\ &= \mu(cons(v_1, map(\partial_1)w_1), map(\partial_1)w_2) \\ &= \mu(\mu(map(\partial_0)w_1, \eta(v_2)), map(\partial_1)w_2) \\ &= \mu(map(\partial_0)w_1, \mu(\eta(v_2), map(\partial_1)w_2)) \\ &= \mu(map(\partial_0)w_1, \mu(map(\partial_0)w_2, \eta(v_3))) \\ &= \mu(map(\partial_0)(\mu(w_1, w_2)), \eta(v_3)), \text{ as required.} \end{aligned}$$

It is similarly easy to show that m is associative and that $e \equiv V \xrightarrow{\Delta} V \times V$ is the identity for m.

We claim that $\mathbf{C} \equiv \langle C_1, V, d_0, d_1, m, e \rangle$ is the free category object over $\mathbf{G}$ in $\breve{E}$, with injection of generators $in : \mathbf{G} \longrightarrow \mathbf{C}$ the graph homomorphism with components

$$in_0 : V \longrightarrow V \equiv id_V$$

$$in_1 : E \longrightarrow C_1 \equiv \langle \partial_0, E, \partial_1 \rangle$$

Now suppose that $\mathbf{D} \equiv \langle D_1, D_0, src, tar, m_D, e_D \rangle$ is a category object in $\breve{E}$ with $\phi : \mathbf{G} \longrightarrow \mathbf{D}$ a graph homomorphism.

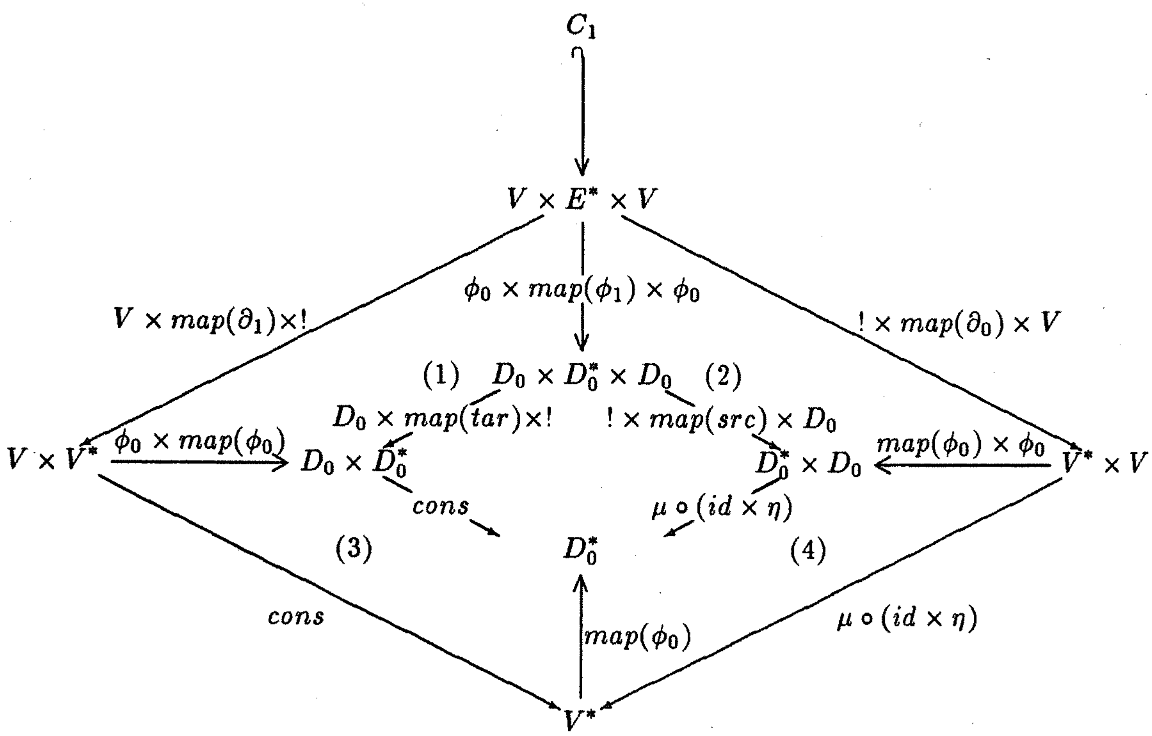
Recall that $\phi_0 \circ \partial_1 = tar \circ \phi_1$ and $\phi_0 \circ \partial_0 = src \circ \phi_1$

We therefore have:

$$map(\phi_0) \circ map(\partial_1) = map(tar) \circ map(\phi_1)$$

$$map(\phi_0) \circ map(\partial_0) = map(src) \circ map(\phi_1)$$

This gives us commutativity of (1) and (2) in the following diagram:



Commutativity of (3) and (4) is trivial. Since the left hand outside path from C_1 to $D_0^* \equiv$ the right hand outside path from C_1 to D_0^*, it follows from the commutativity of (1), (2), (3), (4) that the inside paths from C_1 to D_0^* are the same and hence that

$$C_1 \hookrightarrow V \times E^* \times V \xrightarrow{\phi_0 \times map(\partial_1) \times \phi_0} D_0 \times D_1^* \times D_0$$

induces a map $C_1 \overset{k}{\hookrightarrow} D_a$, where $D_a \hookrightarrow D_0 \times D_1^* \times D_0$ equalises $cons \circ (D_0 \times map(tar) \times !)$ and $\mu \circ (id \times \eta) \circ (! \times map(src) \times D_0)$.

It is fairly clear by now that we want to define an arrow $D_a \longrightarrow D_1$ by returning the composition of all of the maps in an element $\langle o_1, \mathrm{e}, o_2 \rangle$ and by returning $e_D(o_1)$ for an element $\langle o_1, [], o_1 \rangle$. It is not in general clear how we can define the arrow $D_a \longrightarrow D_1$, as our recursive definitions are over arbitary lists and recursive definitions of the following form are *not* permitted:

$$\begin{array}{ccccc}
1 & \xrightarrow{\varepsilon} & D_x & \xleftarrow{cons} & D_1 \times_{D_0} D_a \\
& \searrow^{f} & \Big\downarrow \lambda & & \Big\downarrow D_1 \times \lambda \\
& & C & \xleftarrow{g} & D_1 \times_{D_0} D_a
\end{array}$$

since we are then effectively checking at every application of $cons$ that the element to which we are applying it is in $D_1 \times_{D_0} D_a$ and nothing in chapter 2 entitled us to do this.

In $\check{E}$, we can fall back upon the definition of arrows within E. Recall that $m_D : D_1 \times_{D_0} D_1 \longrightarrow D_1$ in $\check{E}$ is represented by at least one arrow $m_D : N \longrightarrow N$ in E.

We can define an arrow $\lambda : N \longrightarrow N$ in E via the following recursion:

$$\begin{array}{ccccccc}
N & \xrightarrow{0} & N & \xleftarrow{\quad cons \quad} & & & N \times N \\
& \searrow^{0} & \Big\downarrow \lambda & & & & \Big\downarrow N \times \lambda \\
& & N & \xleftarrow{m_D} & N & \xleftarrow{pair} & N \times N
\end{array}$$

It will be convenient to write $triple(n_1, n_2, n_3)$ for $pair(pair(n_1, n_2), n_3)$ and $(tr_1(n), tr_2(n), tr_3(n))$ for the unique triple (n_1, n_2, n_3) such that $n = triple(n_1, n_2, n_3)$.

Now define $f : N \longrightarrow N$ by $n \mapsto triple(tr_1(n), \lambda(tr_2(n)), tr_3(n))$
$g : N \longrightarrow N$ by $n \mapsto (if\ tr_2(n) = 0\ then\ e(tr_1(n))\ else\ tr_2(n))$

Since these functions are PRIM-definable they are both well-defined as arrows in E.

Define $h : N \longrightarrow N \equiv g \circ f$. We claim that $[h] : D_a \longrightarrow D_1$ in $\check{E}$. As usual, our proof uses list induction, as in Lemma 2.35. In this case, for $n = triple(n_1, n_2, n_3)$, we perform our induction on n_2.

Base Case: $n_2 = 0$. Then $D_a(triple(n_1, n_2, n_3)) = 1 \Rightarrow D_0(n_1) = 1 \Rightarrow D_1(h(triple(n_1, n_2, n_3)))$, so we have $D_1(h(triple(n_1, n_2, n_3))) = 1$.

Inductive Step: Assume that

$$D_a(triple(n_1, n_2, n_3)) = 1 \Rightarrow D_1(h(triple(n_1, n_2, n_3))) = 1$$

Suppose that $D_a(triple(n_1, \mu(\eta(n), n_2), n_3)) = 1$. Then if $n_2 = 0$ we must have $D_1(n) = 1$, $D_1(\lambda(\mu(\eta(n), n_2))) = D_1(m_D(pair(n, 0))) = D_1(n) = 1$. If $n_2 \neq 0$ then $D_1(\lambda(n_2)) = 1$, $D_1(\lambda(\mu(\eta(n)), n_2)) = D_1(m_D(n, \lambda n_2)) = 1$, by the definition of arrows in $\check{E}$.

It follows that $[h] : D_a \longrightarrow D$ in $\check{E}$ and hence that $[h \circ k] : C_1 \longrightarrow D$ in $\check{E}$.

Define $\bar{\phi_1} \equiv h \circ k$. We claim that $\bar{\phi} \equiv \langle \bar{\phi_0}, \bar{\phi_1} \rangle$ is an internal functor from **C** to **D**.

$$\begin{aligned}
\bar{\phi_1} \circ e &= V \xrightarrow{\Delta} V \times V \xrightarrow{\phi_0 \phi_0} D_0 \times D_0 \xrightarrow{D_0 \times \epsilon \times D_0} D_0 \times D_1^* \times D_0 \xrightarrow{h} D_1 \\
&= V \xrightarrow{\Delta} V \times V \xrightarrow{\phi_0 \phi_0} D_0 \times D_0 \xrightarrow{e_D \circ \pi_1} D_1 \\
&= V \xrightarrow{\phi_0} D_0 \xrightarrow{e_D} D_1
\end{aligned}$$

So $\bar{\phi}$ respects identities.

To show commutativity of

$$\begin{array}{ccc}
C_1 \times_{C_0} C_1 & \xrightarrow{\bar{\phi_1} \times \bar{\phi_1}} & D_1 \times_{D_0} D_1 \\
\Big\downarrow m & & \Big\downarrow m_D \\
C_1 & \xrightarrow[\bar{\phi_1}]{} & D_1
\end{array}$$

We need to show that if $C_1(r) = 1 = C_1(s)$ and $d_1(r) = d_0(s)$ then $m_D(\bar{\phi_1}(r), \bar{\phi_1}(s)) = \bar{\phi_1}(m(r, s))$.

Write r as $triple(n_1, n_2, n_3)$ and proceed by list induction on r_2.

Base Case: $r_2 = 0$. If $C_1(r) = 1 = C_1(s)$ with $d_1(r) = d_0(s)$ then $r_1 = r_3 = s$.

$m_D(\bar{\phi}_1(r), \bar{\phi}_1(s)) = m_D(e_D(r_3), \bar{\phi}_1(s)) = \bar{\phi}_1(s) = \bar{\phi}_1(m(r,s))$.

Inductive Step: Suppose that $C_1(triple(r_1, \mu(\eta(r), r_2), r_3)) = 1 = C_1(s)$ and that $d_1(triple(r_1, \mu(\eta(r), r_2), r_3)) = d_0(s)$.

Then by definition of C_1, $C_1(triple(r_1, r_2, r_3)) = 1$ and by definition of d_1, $d_1(triple(r_1, \mu(\eta(r), r_2), r_3)) = d_0(s)$, so we can use the IH to deduce that

$$m_D(\bar{\phi}_1(triple(r_1, r_2, r_3)), \bar{\phi}_1(s)) = \bar{\phi}_1(m(triple(r_1, r_2, r_3), s))$$

Then

$$\begin{aligned}
&m_D(\bar{\phi}_1(triple(r_1, \mu(\eta(r), r_2), r_3)), \bar{\phi}_1(s)) \\
&\quad = m_D(\lambda \circ map(\phi_1) \circ \mu(\eta(r), r_2), \bar{\phi}_1(s)) \\
&\quad = m_D(\lambda \circ \mu(\eta(\phi_1(r)), map(\phi_1) r_2), \bar{\phi}_1(s)) \\
&\quad = m_D(m_D(\phi_1(r), \lambda \circ map(\phi_1) r_2), \bar{\phi}_1(s)) \quad \text{(Def of } \lambda) \\
&\quad = m_D(\phi_1(r), \bar{\phi}_1(m(triple(r_1, r_2, r_3), s))) \quad \text{(Associativity and IH)} \\
&\quad = \bar{\phi}_1(m(triple(r_1, \mu(\eta(r, r_2)), r_3), s)) \quad \text{by definition, as required}
\end{aligned}$$

So $\bar{\phi}$ *is* a functor.

For commutativity:

$$\begin{aligned}
(\bar{\phi}_1 \circ in_1)(n) &= k(\phi_0(src(n)), \phi_1(n), \phi_0(tar(n))) \\
&= \phi_1(n)
\end{aligned}$$

So $p\bar{h}i \circ in = \phi$.

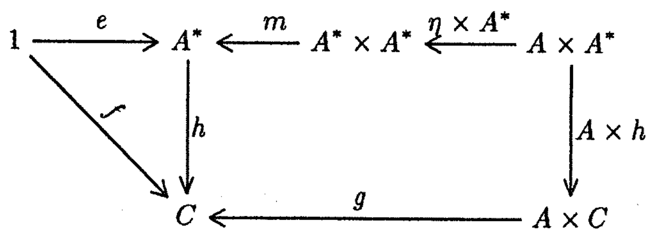
Uniqueness follows via another easy induction. □

It is worth remarking again upon why the proof succeeded. All of the reasoning could have been performed without alteration in any category with finite left limits and list objects, up to the point where we needed to define an arrow $D_a \longrightarrow D$. To do this, we needed to fold m_D through D_a; in general, this is *not* possible, for the reasons mentioned in the course of the proof. We can however get away with this in $\check{E}$ because we can use the fact that m_D is defined on *products* in E and not on pullbacks - this allows us to demonstrate inductively in $\check{E}$ that it has the desired properties.

Definition 5.61 *Let C be a category which has free category objects over graph objects and let $\mathbf{G}$ be any internal graph object in C with associated free internal category $\mathbf{C}$ and let $\eta : \mathbf{G} \longrightarrow \mathbf{C}$ be the unit of the corresponding adjunction. If any internal diagram $\langle F, \pi, \gamma \rangle$ on $\mathbf{G}$ lifts to a unique internal diagram $\langle F, \pi, \gamma' \rangle$ on $\mathbf{C}$ with $\gamma' \circ (\eta \times_{G_0} F) = \gamma$ then we say that C* has action variants.

Lemma 5.62 *Any finitely complete category which has action variants has list objects.*

Proof Let $\mathcal{E}$ be any finitely complete category and let $A^* \overset{!}{\underset{!}{\rightrightarrows}} 1$ be the free category over the graph object $A \overset{!}{\underset{!}{\rightrightarrows}} 1$ in $\mathcal{E}$. Then A^* has a multiplication operator m and a unit e, as above. Consider the recursion data $1 \xrightarrow{f} C \xleftarrow{g} A \times C$. To show that A^* is a list object over A, we need to show the existence of a unique h such that the following diagram commutes:



Now observe that $\langle C, !, g \rangle$ is a diagram over the graph $A \overset{!}{\underset{!}{\rightrightarrows}} 1$ and therefore by assumption must lift to a unique diagram $\langle C, !, g' \rangle$ over $A^* \overset{!}{\underset{!}{\rightrightarrows}} 1$, with $g' \circ (\eta \times C) = g$. Define h to be

$$h \equiv A^* \xrightarrow{\sim} A^* \times 1 \xrightarrow{A^* \times f} A^* \times C \xrightarrow{g'} C$$

Then we claim that h is the desired map.

Firstly, $h \circ e = g' \circ \langle e, f \rangle = f$ by the definition of an action.

Also, $h \circ \mu(\eta \times A^*) = g' \circ (A^* \times f) \circ m \circ (\eta \times A^*)$
$= g' \circ \langle m \circ (\eta \times A^*), f \circ ! \rangle$
$= g' \circ \langle \eta, g' \circ \langle A^*, f \rangle \rangle$, by definition of internal diagram
$= g \circ g' \circ \langle A^*, f \rangle$, by the properties of action variants

$= g \circ (A \times h)$, as required

For uniqueness, note that any h which makes the above diagram commute is the action variant of the following action:

$$\langle C, !, A \times C \xrightarrow{\pi_1} A \xrightarrow{A \times f} A \times C \xrightarrow{g} C \rangle$$

and must necessarily be unique. □

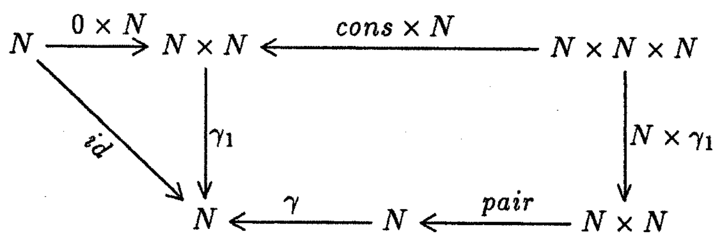
Theorem 5.63 *$\check{E}$ has action variants.*

Proof Let $\mathbf{G} \equiv G_1 \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} G_0$ be an $\check{E}$-graph and suppose that $\langle F, \pi, \gamma \rangle$ is a diagram over $\mathbf{G}$. Let $\mathbf{C} \equiv C \overset{d_1}{\underset{d_0}{\rightrightarrows}} G_0$ be the free category over $\mathbf{G}$, as described in Theorem 5.60. We want to define $\gamma' : C \times_{G_0} F \longrightarrow F$ so that $\pi \circ \gamma' = d_1 \circ \pi_1$. Recall that $C_1 \hookrightarrow V \times E^* \times V$ equalises $cons \circ (V \times map(\partial_1) \times !)$ and $\mu \circ (id \times \eta) \circ (! \times map(\partial_0) \times V)$, so in **Set** we could easily define γ':

$$\langle \langle f_1, \ldots, f_n \rangle, x \rangle \mapsto \gamma(f_1, \gamma(f_2, \ldots, \gamma(f_n, x) \ldots))$$

However, we cannot in general make a recursive definition of γ' along these lines, as we cannot make recursive definitions over pullbacks, precisely as stated in the proof of Theorem 5.60

We again employ the technique of Theorem 5.60 and make the desired definitions in E before proving inductively that they work in $\check{E}$. Recall that $\gamma : N \longrightarrow N$ in E and define $\gamma_1 : N \times N \longrightarrow N$ in E using the following recursion:



So γ_1 will fold up a list, as in **Set** above.

Define $\phi : N \longrightarrow N$ by

$$triple(n_1, n_2, n_3) \mapsto (if\ n_2 = 0\ then\ e_C(n_1)\ else\ n_2)$$

Since ϕ is PRIM-definable, this is a meaningful definition.

Define γ' by $n \mapsto \gamma_1(\phi(pr_1(n), pr_2(n)))$. We claim that $[\gamma'] : C \times_{G_0} F \longrightarrow F$ in $\check{E}$.

We need to show that if $C(m) = F(n) = 1$ and $d_0(m) = \pi(n)$ then $F(\gamma'(pair(m,n))) = 1$. We write m as $triple(m_1, m_2, m_3)$ and proceed by list induction on m_2.

Base Case: $m_2 = 0$. Then

$$\begin{aligned} F(\gamma'(pair(m,n))) &= F(\gamma_1(pair(e(m_1),n))) \\ &= F(\gamma(pair(e(m_1),n))) \\ &= F(n), \text{ by the unit law} \\ &= 1, \text{ by assumption.} \end{aligned}$$

Inductive Step: Let $m \equiv triple(m_1, cons(m', m_2), m_3)$ with the result true for $triple(m_1, m_2, m_3)$. Suppose that

$$C(triple(m_1, cons(m', m_2), m_3)) = F(n) = 1 \quad \text{and} \quad d_1(m) = \pi(n)$$

Then

$$F(\gamma'(pair(m,n))) = F(\gamma(pair(m', \gamma'(pair(triple(m_1, m_2, m_3), n)))))$$

Recall that $\gamma : G_1 \times_{G_0} F \longrightarrow F$. We can apply our inductive hypothesis to see

$$(G_1 \times_{G_0} F)(pair(m', \gamma'(pair(triple(m_1, m_2, m_3), n)))) = 1$$

so $F(\gamma'(pair(m,n))) = 1$, as required.

To prove the desired commutativity $\pi \circ \gamma' = d_1 \circ \pi_1$ in $\check{E}$, we need to show that if $C(m) = F(n) = 1$ and $d_0(m) = \pi(n)$ then $\pi(\mu'(pair(m,n))) = d_1(m)$. This falls out very quickly from another list induction on n_2.

Finally, it is clear from the definition that $\gamma' \circ (\eta \times G_1) = \gamma$. Uniqueness of the γ' follows from yet another list induction, which we ommit.

This completes the proof. □

Observe again that we could not use the machinery currently available to us to prove the existence of action variants in an arbitary category with free category objects over graph objects because we would again need to form a recursive definition involving pullbacks.

5.2.2 The Exact Completion of $\check{E}$

Recall that the purpose of Joyal's construction was to produce a category which contained a model of itself. We can model Σ in $\check{\Sigma}$ but to extend this model to one for $\check{\Sigma}$ we need to form coequalisers of equivalence relations and this is not possible. We now show how to add the required structure.

Definition 5.64 *A category* C *is* regular *if it has:*

1. *Finite Limits*
2. *Stable surjections*
3. *A surjective image for every arrow.*

We showed in the preceeding section that $\check{E}$ is regular.

Definition 5.65 *A regular category which has coequalisers of equivalence relations is called* exact.

There is a standard completion process from a regular category to an exact category. From a regular category C, we form a category $Map(C)$ which has objects the equivalence relations on the elements of $Ob(C)$ and arrows the functional relations defined between them. The details appear in [FreySce 90, McL 94].

Taking the exact completion of a category of the form $\check{E}$ is rendered far simpler by the presence of split epi/monic factorisations and of the bounded minimisation operator. We will give the details of the procedure in that specific instance.

Firstly, we will prove some results which will give us a better understanding of equivalence relations.

Definition 5.66 *A* groupoid *is a category in which every arrow is invertible.*

Theorem 5.67 *We can identify equivalence relations in a cartesian left exact category* C *with the groupoids in* C *for which* $\langle dom, cod \rangle$ *is monic.*

Proof Suppose that

$$Arr(R_x) \underset{\xleftarrow{e}}{\overset{\xrightarrow{dom}}{\xrightarrow{cod}}} A_x$$

is a groupoid in C with inverse map $Arr(R_x) \xrightarrow{inv} Arr(R_x)$ and composition map $2Arr(R_x) \xrightarrow{comp} Arr(R_x)$, where the following is a pullback:

$$\begin{array}{ccc} 2Arr(R_x) & \longrightarrow & Arr(R_x) \\ \Big\downarrow & & \Big\downarrow{\scriptstyle dom} \\ Arr(R_x) & \xrightarrow{cod} & Arr(R_x) \end{array}$$

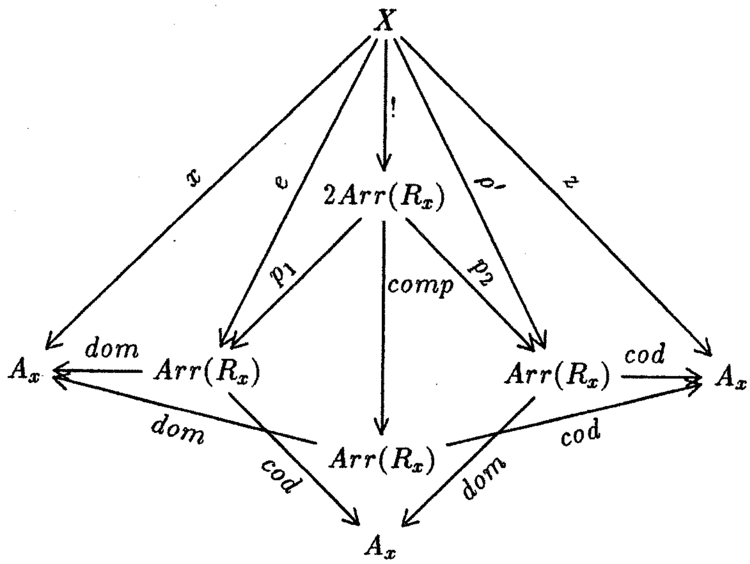
and suppose further that dom and cod are jointly monic. We show that $Arr(R_x) \overset{\langle dom,cod\rangle}{\rightarrowtail} A_x \times A_x$ is an equivalence relation:

Reflexivity $dom \circ e = cod \circ e = id$, so for any $X \xrightarrow{x} A_x$ the following diagram commutes:

$$\begin{array}{ccccc} & & X & & \\ & {\scriptstyle x}\swarrow & \downarrow{\scriptstyle x} & \searrow{\scriptstyle x} & \\ & & A_x & & \\ & & \downarrow{\scriptstyle e} & & \\ A_x & \xleftarrow{dom} & Arr(R_x) & \xrightarrow{cod} & A_x \end{array}$$

Symmetry If xRy then there is a z such that $dom \circ z = x$ and $cod \circ z = y$. Recall that $dom \circ inv = cod$ and $cod \circ inv = dom$, so $\langle y, x\rangle$ factors through R via $inv \circ z$.

Transitivity If $xRyRz$ then there are ρ and ρ' which compose with R to give $\langle x, y\rangle$ and $\langle y, z\rangle$ respectively. Since $cod \circ \rho = dom \circ \rho' = y$ we have a map $X \xrightarrow{!} 2Arr(R_x)$ which makes the following commute:



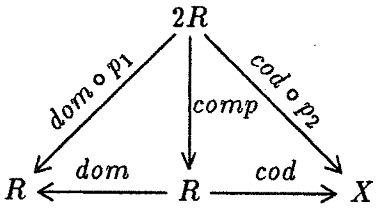
All of the subdiagrams in the above diagram which involve $2Arr(R_x)$ commute by the category axioms for $comp$, so $\langle x, z\rangle$ factors through R via $comp \circ !$.

For the converse, suppose that $R \xhookrightarrow{\langle dom, cod\rangle} X \times X$ is an equivalence relation in $\mathcal{C}$. Since R is reflexive, we must have $\langle X, X\rangle \in R$; define $e : X \longrightarrow R$ to be the map which satisfies $dom \circ e = X$ and $cod \circ e = X$.

We trivially have $\langle dom, cod\rangle \in R$ and so by symmetry of R we must have that $\langle cod, dom\rangle \in R$; define $inv : R \longrightarrow R$ to be the map which satisfies $dom \circ inv = cod$ and $cod \circ inv = dom$. Suppose that the following diagram is a pullback:

$$\begin{array}{ccc} 2R & \xrightarrow{p_1} & R \\ {\scriptstyle p_2}\downarrow & & \downarrow{\scriptstyle cod} \\ R & \xrightarrow{dom} & X \end{array}$$

We must have $\langle dom \circ p, cod \circ p_1\rangle \in R$, with p_1 the factorising arrow. Now $cod \circ p_1 = dom \circ p_2$ so $\langle cod \circ p_1, cod \circ p_2\rangle \in R$, with p_2 the factorising arrow. It follows from the transitivity of R that $\langle dom \circ p_1, cod \circ p_2\rangle \in R$, so there is an arrow $comp : 2R \longrightarrow R$ which makes the following diagram commute:



We show that *cod*, *dom*, *e*, *inv*, *comp* are together a groupoid in $\mathcal{C}$:

By definition, $dom \circ comp = dom \circ p_1$

$$cod \circ comp = cod \circ p_2$$

So
$$\begin{aligned} dom \circ comp \circ (Comp \times_X R) &= dom \circ p_1 \circ (comp \times_X R) \\ &= dom \circ comp \circ p_1 \\ &= dom \circ p_1 \circ p_1 \\ dom \circ comp \circ (R \times_X comp) &= dom \circ p_1 \circ (R \times_X comp) \\ &= dom \circ p_1 \circ p_1 \end{aligned}$$

Similarly, $cod \circ comp \circ (comp \times_X R) = cod \circ comp \circ (R \times_X comp)$, so since *dom* and *cod* are jointly monic, we must have

$$comp \circ (comp \times_X R) = comp \circ (R \times_X comp)$$

Similarly,
$$\begin{aligned} comp \circ (R \times e) \circ \langle R, dom \rangle &= R \\ comp \circ (e \times R) \circ \langle R, dom \rangle &= R \\ comp \circ \langle R, inv \rangle &= e \circ cod \\ comp \circ \langle inv, R \rangle &= e \circ dom \end{aligned}$$

which completes the proof. □

Definition 5.68 *Suppose that E is a Skolem Theory. We define the category $\bar{\bar{E}}$ as follows:*

Objects *Pairs $\langle X, R \rangle$, where $X \in Ob(\check{E})$ and R is an equivalence relation on X.*

Arrows *Given $\bar{\bar{E}}$-objects $\langle X,R\rangle$ and $\langle Y,S\rangle$, define*

$$Arr_{\langle X,S\rangle,\langle Y,S\rangle} = \{X \xrightarrow{f} Y \mid R \subseteq (f\times f)^{-1}(Y)\}$$

The relation $\sim$ on $Arr_{\langle X,S\rangle,\langle Y,S\rangle}$ is defined by $f \sim g$ iff $R \subseteq (f \times g)^{-1}(S)$. That $\sim$ is an equivalence relation on $Arr_{\langle X,S\rangle,\langle Y,S\rangle}$ is an easy consequence of the fact that R, S are equivalence relations. An arrow $\langle X,R\rangle \xrightarrow{[f]} \langle Y,S\rangle$ is an equivalence class of $Arr_{\langle X,S\rangle,\langle Y,S\rangle}$ under $\sim$.

Lemma 5.69 *Composition in $\bar{\bar{E}}$ is well-defined.*

Proof Suppose that $\langle X,R\rangle \xrightarrow{[f]} \langle Y,S\rangle \xrightarrow{[g]} \langle Z,T\rangle$ with $f \sim f'$ and $g \sim g'$.

Then $R \subseteq (f\times f)^{-1}(S) \subseteq (f\times f)^{-1}(g\times g)^{-1}(T)$
$= (g\circ f \times g\circ f)^{-1}(T)$ by the properties of p.b.'s

So $[g\circ f] : \langle X,R\rangle \longrightarrow \langle Z,T\rangle$.
Similarly, $[g'\circ f'] : \langle X,R\rangle \longrightarrow \langle Z,T\rangle$.

Now $R \subseteq (f\times f)^{-1}(S) \subseteq (f\times f')^{-1}(g\times g')^{-1}(T)$
$= (g\circ f \times g'\circ f')(T)$

So $g'\circ f' = g\circ f$. □

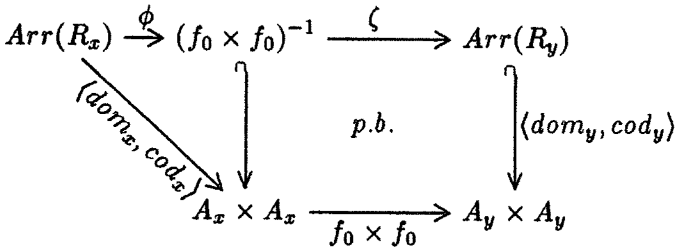
Lemma 5.70 *An equivalent definition of $\bar{\bar{E}}$ is the following:*

Objects *Groupoids $Arr(R_x) \underset{cod}{\overset{dom}{\rightrightarrows}} A_x$, with dom, cod jointly monic.*

Arrows *inv-preserving functors. (ie Groupoid homomorphisms), with $f \sim g$ iff there is a natural transformation $f \longrightarrow g$.*

Proof Equivalence of objects is immediate from Theorem 5.67.

To see the result for arrows, observe that an arrow between the groupoids $(Arr(R_x) \underset{cod_x}{\overset{dom_x}{\rightrightarrows}} A_x)$ and $(Arr(R_y) \underset{cod_y}{\overset{dom_y}{\rightrightarrows}} A_y)$ is an arrow $A_x \xrightarrow{f_0} A_y$ with some ϕ such that the following diagram commutes:



Put $f_1 = \zeta \circ \phi$. We claim that $f = \langle f_0, f_1 \rangle$ is a groupoid homomorphism.

From the definition of f we immediately have that:

$f_0 \circ dom_x = dom_y \circ f_1$

$f_0 \circ cod_x = cod_y \circ f_1$

Also, $dom_y \circ e \circ f_0 = f_0$ and $dom_y \circ f_1 \circ e = f_0 \circ dom_x \circ e = f_0$

$cod_y \circ e \circ f_0 = f_0$ and $cod_y \circ f_1 \circ e = f_0 \circ cod_x \circ e = f_0$

So f preserves units, since $\langle dom, cod \rangle$ is monic.

$dom_y \circ comp \circ (f_1 \times f_1) = dom_y \circ p_1 \circ f_1 \times f_1 = dom_y \circ f_1 \circ p_1$

$dom_y \circ f_1 \circ comp = f_0 \circ dom_x \circ comp = f_0 \circ dom_x \circ p_1 = dom_y \circ f_1 \circ p_1$

$cod_y \circ comp \circ (f_1 \times f_1) = cod_y \circ p_2 \circ (f_1 \times f_1) = cod_y \circ f_1 \circ p_2$

$cod_y \circ f_1 \circ comp = f_0 \circ cod_x \circ comp = f_0 \circ cod_x \circ p_2 = cod_y \circ f_1 \circ p_2$

So f preserves multiplication, since $\langle dom, cod \rangle$ is monic.

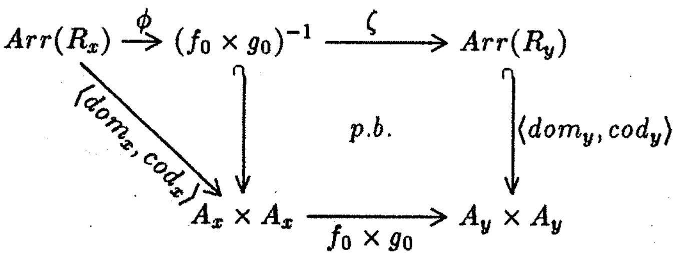
Preservation of inverses is equally trivial.

Conversely, if $f : \langle Arr(R_x), A_x \rangle \longrightarrow \langle Arr(R_y), A_y \rangle$ is a functor,

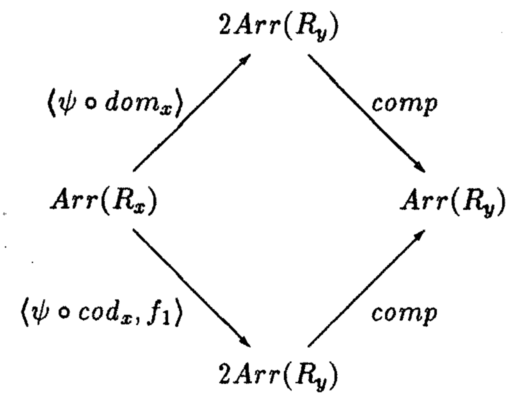
$$Arr(R_x) \xrightarrow{\langle dom_x, cod_x \rangle} A_x \times A_x$$

factors through $(f_0 \times f_0)^{-1}(Arr(R_y))$ via $\langle f_1, \langle dom_x, cod_x \rangle \rangle$.

Suppose that two arrows f, g in $\bar{\bar{E}}$ are equivalent; there is then some ϕ which makes the following diagram commute:



Define $\psi = \zeta \circ \phi \circ e$. Then $\langle dom_y, cod_y \rangle \circ \psi = \langle f_0, g_0 \rangle$, trivially; it is easy to show that the following commutes , using the usual argument with the monicness of $\langle dom, cod \rangle$:



So ψ is a natural transformation from f to g. □

Lemma 5.71 *$\check{E}$ is a full subcategory of $\bar{\check{E}}$.*

Proof Define $\theta : \check{E} \longrightarrow \bar{\check{E}}$ on objects A by $\theta A \equiv \bar{A}$, where $\bar{A}$ is the discrete groupoid on A with $dom = cod = e = inv = A$ and $comp \equiv \Delta_A \xrightarrow{\pi_1} A$ and on arrows $A \xrightarrow{f} B$ by $\theta f \equiv \bar{A} \xrightarrow{[f]} \bar{B}$, where $\bar{f}$ is the functor $\langle f, f \rangle : \bar{A} \longrightarrow \bar{B}$.

Then any functor $\langle f_0, f_1 \rangle : \bar{A} \longrightarrow \bar{B}$ must satisfy $f_0 \circ dom_{\bar{A}} = dom_{\bar{B}} \circ f_1$ ie $f_0 = f_1$, and is therefore the image under θ of $[f_0] : A \longrightarrow B$. □

Before we prove that $\bar{\check{E}}$ has the properties which we require, we will prove a couple of Lemmas which show us why the split-epi/monic factorisation is important. Recall that in **Set**, the Axiom of Choice can be stated as "All epis split" or alternatively as "For any relation $F \subseteq X \times Y$ satisfying $\forall x \in X \exists y \in Y.\langle x, y \rangle \in F$, there is a function $f : X \longrightarrow Y$ with $f \subseteq F$". The following lemmas are analogous and reflect the level of choice which the split epi/monic factorisation in $\check{E}$ gives us:

Lemma 5.72 *If B/R_B is a $\bar{\check{E}}$-object then there is a $\check{E}$-object C and an arrow $h : B \longrightarrow C$ in $\bar{\check{E}}$ such that $h : B/R_B \cong C/\Delta_C$ in $\bar{\check{E}}$.*

Proof Define $C : N \longrightarrow N$ by $n \mapsto \mu_{m<n}(B(m).(nR_Bm))$.

Define $C : N \longrightarrow N$ by $n \mapsto eq(1, \Sigma_{m \leq n} B(m).(nR_Bm))$.

Then C is clearly a predicate and $B(n) = 1 \Rightarrow C(h(n)) = 1$, so $h : B \longrightarrow C$ in $\bar{\bar{E}}$. h is clearly surjective and hence has a right inverse k : $C \longrightarrow B$, $h \circ k = id_C$.

To show that $h : B/R_B \longrightarrow C/\Delta_C$, suppose that xR_By.

Then if $h(x) = m_x$ and $h(y) = m_y$, m_xR_Bx and m_yR_By so that $m_xR_Bm_y$ by the properties of equivalence relations. Then m_x and m_y are both $\mu_{m<x}(B(m).(xR_Bm))$ and so we must have $m_x = m_y$. It follows that $R_B \subseteq (h \times h)^{-1}\Delta_C$.

$k : C/\Delta_C \longrightarrow B/R_B$ trivially:$x\Delta_Cy \Leftrightarrow x = y \Rightarrow k(x)R_Bk(y)$.

Since $h \circ k = id_C$ in E, k is the right inverse for h in $\bar{\bar{E}}$.

Suppose that $k(h(n)) = n'$. We need to show that nR_Bn'.

$h(n) = h(n')$ because k is the right inverse to h, so $h(n')R_Bn'$ and $h(n)R_Bn$, so we have $n'R_Bn$ as required, which completes the proof. □

Corollary 5.73 *If $f, g : B/R_B \longrightarrow C/R_C$ in $\bar{\bar{E}}$ with $f = g$ in $\bar{\bar{E}}$ then there is a D in $Ob(\bar{E})$ and an isomorphism $h : C/R_C \longrightarrow D/\Delta_D$ such that $h \circ g = h \circ f$ in E.*

Proof We use the object which we constructed in the above theorem. Then $xR_By \Rightarrow f(x)R_Bg(y) \Rightarrow (h \circ f)(x)\Delta_C(h \circ g)(x) \Rightarrow (h \circ g)(x) = (h \circ f)(y)$.

In particular, $x = y \Rightarrow (h \circ g)(x) = (h \circ f)(y)$ □

We are now in a position to prove that $\bar{\bar{E}}$ has the properties which we require:

Theorem 5.74 *$\bar{\bar{E}}$ has co-equalisers of equivalence relations.*

Proof Suppose that $A/R_A \overset{f}{\underset{g}{\rightrightarrows}} B/R_B$ is an equivalence relation on B/R_B. In view of Lemma 5.72, we can assume WLOG that $R_A = \Delta_A$ and $R_B = \Delta_B$. We know that $\Delta_B \subseteq \langle f, g \rangle$ by the properties of equivalence relations, from which is follows that both f and g are surjective in $\bar{E}$ and that they split. Call their right inverses f' and g' respectively.

Define R' to be the relation on B given by $xR'y$ iff $f'x = g'y$. We claim that R' is an equivalence relation on B.

R' is trivially reflexive. Suppose that $xR'y$. Then there is some a with $\langle f, g \rangle \circ a = \langle x, y \rangle$. Since $\langle f, g \rangle$ is an equivalence relation, it follows that for some b, $\langle f, g \rangle \circ b = \langle y, x \rangle$ - in other words, $yR'x$. Finally, if $xR'yR'z$ then

$f'x = g'y$ and $f'y = g'z$, so $\langle x,y\rangle \in \langle f,g\rangle$ and $\langle y,z\rangle \in \langle f,g\rangle$ and so from the transitivity of $\langle f,g\rangle$ we have that $xR'z \in \langle f,g\rangle$.

Now note that $id_B : B/R_B \longrightarrow B/R'$ in $\bar{\bar{E}}$ and that this map trivially has the universal properties of a coequaliser. □

Lemma 5.75 *$\bar{\bar{E}}$ has finite products.*

Proof Given any two $\bar{\bar{E}}$-objects, we have maps

$$A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$$

Define $R_{A\times B} \equiv R_A \times R_B \xrightarrow{tw\circ(R_A,R_B)} (A \times B) \times (A \times B)$, where tw is the twisting isomorphism $(A_1 \times A_2) \times (B_1 \times B_2) \xrightarrow{\sim} (A_1 \times B_1) \times (A_2 \times B_2)$.

The following diagram commutes:

$$\begin{array}{ccccc}
R_A & \xleftarrow{\pi_1'} & R_{A\times B} & \xrightarrow{\pi_2'} & R_B \\
\downarrow & & \downarrow & & \downarrow \\
A \times A & \xleftarrow{\pi_1 \times \pi_1} & (A \times B) \times (A \times B) & \xrightarrow{\pi_2 \times \pi_2} & B \times B
\end{array}$$

so that $A/R_A \xleftarrow{\pi_1\times\pi_1} A \times B/R_{A\times B} \xrightarrow{\pi_2\times\pi_2} B/R_B$ is a diagram in $\bar{\bar{E}}$.

It is trivially the case that if $A/R_A \xleftarrow{f_1} C/R_C \xrightarrow{f_2} B/R_B$ is any diagram in $\bar{\bar{E}}$ then the following commutes:

$$\begin{array}{ccccc}
 & & C/R_C & & \\
 & \swarrow^{f_1} & \downarrow{\scriptstyle \langle f_1,f_2\rangle} & \searrow^{f_2} & \\
A/R_A & \xleftarrow{\pi_1 \times \pi_1} & A \times B/R_{A\times B} & \xrightarrow{\pi_2 \times \pi_2} & B/R_B
\end{array}$$

For uniqueness, suppose that g also makes this diagram commute.

Then $R_C \subseteq (f_1 \times \pi_1 \circ g)^{-1} R_A$
$R_C \subseteq (f_2 \times \pi_2 \circ g)^{-1} R_A$

It follows that $R_C \subseteq (\langle f_1, f_2\rangle \times g)^{-1} R_{A\times B}$, so that $g = \langle f_1, f_2\rangle$ in $\bar{\bar{E}}$. □

Corollary 5.76 *The full inclusion $\bar{E} \hookrightarrow \bar{\bar{E}}$ preserves finite products.*

Proof Immediate from the above proof. □

Lemma 5.77 *$\bar{\bar{E}}$ has finite coproducts.*

Proof Given objects A, B in $\bar{\bar{E}}$, we have a map

$$A \xrightarrow{i_1} A+B \xleftarrow{i_2} B$$

Define $R_{A+B} \equiv R_A + R_B \xrightarrow{tw \circ [R_A, R_B]} (A+B) \times (A+B)$, where tw is the twisting isomorphism $(A_1 \times A_2) + (B_1 \times B_2) \xrightarrow{\sim} (A_1 \times B_1) + (A_2 \times B_2)$.

The following diagram commutes (in fact, since the ambient category is extensive and the top row is a coproduct the whole diagram must be a pullback):

$$\begin{array}{ccccc} R_A & \xrightarrow{i_1'} & R_{A+B} & \xleftarrow{i_2'} & R_B \\ \downarrow & & \downarrow & & \downarrow \\ A \times A & \xrightarrow{i_1 \times i_1} & (A+B) \times (A+B) & \xleftarrow{i_2 \times i_2} & B \times B \end{array}$$

Trivially, if $A/R_A \xrightarrow{f_1} C/R_C \xleftarrow{f_2} B/R_B$ is any $\bar{\bar{E}}$ -diagram then the following commutes:

$$\begin{array}{ccccc} & & C/R_C & & \\ & \nearrow^{f_1} & \uparrow [f_1, f_2] & \nwarrow^{f_2} & \\ A/R_A & \xrightarrow{i_1} & A+B/R_{A+B} & \xleftarrow{i_2} & B/R_B \end{array}$$

For uniqueness, suppose that g also makes this diagram commute.

Then $R_A \subseteq (f_1 \times g \circ i_1)^{-1} R_C = i_1 \circ ([f_1, f_2] \times g)^{-1} R_C$
$R_B \subseteq i_2 \circ ([f_1, f_2] \times g)^{-1} R_C$

It follows that $R_{A+B} \subseteq ([f_1, f_2] \times g)^{-1} R_C$, so $g = [f_1, f_2]$ in $\bar{\bar{E}}$. □

Corollary 5.78 *The full inclusion $C \hookrightarrow \bar{C}$ preserves coproducts.*

Proof Trivial from the proof of the preceeding Lemma. □

To show that $\bar{\bar{E}}$ is exact, we want to show that it has two further properties. The proofs are so similar to those above that we simply state both properties as Lemmas with extremely rapid sketches of the proofs:

Lemma 5.79 *$\bar{\bar{E}}$ has equalisers of parallel pairs of arrows and the full inclusion $\check{E} \hookrightarrow \bar{\bar{E}}$ preserves equalisers.*

Proof (Sketch) We use Lemma 5.72 to assume WLOG that any pair of arrows $A/R_A \overset{f}{\underset{g}{\rightrightarrows}} B/R_B$ are between discrete groupoids and then take the equaliser of f, g in $\check{E}$. The embedding of this object into $\bar{\bar{E}}$ as a discrete groupoid is the required equaliser; because the $\bar{\bar{E}}$-objects are *discrete* groupoids, any commuting diagram $C/R_C \longrightarrow A/R_A \overset{f}{\underset{g}{\rightrightarrows}} B/R_B$ commutes in $\check{E}$ and the universal property of the equaliser then follows easily from that of the equaliser in $\check{E}$. □

Lemma 5.80 *$\bar{\bar{E}}$ has surjective images which are stable under pullback. The full inclusion $\check{E} \hookrightarrow \bar{\bar{E}}$ preserves surjective images.*

Proof (Sketch) Given an arrow in $\check{E}$ we assume as usual that it is between discrete groupoids. Its surjective image is then easily seen to be the image under the full inclusion of the surjective image of the same arrow in $\check{E}$. □

To show that $\bar{\bar{E}}$ has nice recursive properties, we state a trivial Lemma which will be useful in the proof of the following Theorem.

Lemma 5.81 *If $E/R_E \overset{\partial_1}{\underset{\partial_0}{\rightrightarrows}} V/R_V$ is a category object in $\bar{\bar{E}}$ with composition operation m then if fR_Vg and $f'R_Vg'$ with $m(f,f')$ and $m(g,g')$ both defined then $m(f,f')R_Vm(g,g')$.*

Proof Straightforward from the definitions.

$$m : E/R_E \times_{V/R_V} E/R_E \longrightarrow E/R_E \text{ in } \bar{\bar{E}}.$$

$E/R_E \times_{V/R_V} E/R_E \subseteq E \times E/R_{E\times E}$ where $\langle f,g\rangle R_{E\times E}\langle f',g'\rangle$ iff fR_Ef' and gR_Eg', so the result follows from the definition of arrows in $\bar{\bar{E}}$. □

Theorem 5.82 *$\bar{\bar{E}}$ has free category objects over graph objects.*

Proof Suppose that $E/R_E \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} V/R_V$ is a graph object in $\bar{\bar{E}}$, with $R_E \overset{\langle u,v\rangle}{\rightarrowtail} V$. Then there is a free category object over $E \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} V$ in $\check{E}$, $\mathbf{C} \equiv \langle C, V, d_0, d_1, e, m\rangle$. Consider the following $\check{E}$-graph:

$$R_E \overset{\langle u,v\rangle}{\rightarrowtail} E \times E \xrightarrow{\eta\times\eta} C \times C \overset{d_0\times d_0}{\underset{d_1\times d_1}{\rightrightarrows}} V \times V$$

ie

$$R_E \underset{d_1 \circ \eta \circ u, d_1 \circ \eta \circ v}{\overset{d_0 \circ \eta \circ u, d_0 \circ \eta \circ v}{\rightrightarrows}} V \times V$$

Let $R'_C \underset{c_2}{\overset{c_1}{\rightrightarrows}} V \times V$ be the free category over this graph. A lengthy and unwieldy diagram chase of the form of Theorem 5.60 shows us that

$$R'_C \hookrightarrow (V \times V) \times R_E^* \times (V \times V) \xrightarrow{\lambda} (V \times E^* \times V) \times (V \times E^* \times V)$$

induces a map $R'_C \longrightarrow C \times C$ (because of the equaliser properties of C), where λ is the following map:

$$\begin{array}{rc} (V \times V) \times R_E^* \times (V \times V) & \xrightarrow{(V \times V) \times map(\langle u,v \rangle) \times (V \times V)} \\ (V \times V) \times (E \times E)^* \times (V \times V) & \xrightarrow{\langle \pi_1, map(\pi_1), \pi_1 \rangle \times \langle \pi_2, map(\pi_2), \pi_2 \rangle} \\ (V \times E^* \times V) \times (V \times E^* \times V) & \end{array}$$

Let $R_C \hookrightarrow C \times C$ be the image of this map. Then we claim:

1. R_C is an equivalence relation on C;
2. $C/R_C \underset{d_1}{\overset{d_0}{\rightrightarrows}} V/R_V$ is a category object in $\bar{\bar{E}}$;
3. $C/R_C \underset{d_1}{\overset{d_0}{\rightrightarrows}} V/R_V$ is the free category object over $E/R_E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V/R_V$ in $\bar{\bar{E}}$.

The proof runs as follows:

1. Reflexivity of R_C is trivial. One can use induction to shpw symmetry and transitivity. The proofs are similar; we show symmetry only.

   To show

   $$\begin{aligned} & triple(v_1, e_1, v_2) R_C triple(v_3, e_2, v_2) \\ & \quad \Rightarrow triple((v_3, e_2, v_2) R_C triple(v_1, e_1, v_2) \end{aligned}$$

   we use list induction on e_1.

   Base Case: If $e_1 = 0$ then we must have $e_2 = 0$, since

   $$\langle triple(v_1, e_1, v_2), triple(v_3, e_2, v_2) \rangle$$

factors through $(V \times V) \times R_E^* \times (V \times V)$ via λ, so by definition of R_C, $v_2 = v_3$ and $v_1 = v_2$. It follows trivially that

$$triple(v_3, e_2, v_2) R_C triple(v_1, e_1, v_2)$$

Inductive Step: Assume that the statement holds for e_1, e_2. and suppose that

$$triple(v_1, cons(e, e_1), v2) R_C triple(v_3, cons(e', e_2), v_2)$$

Then $e R_E e'$ and $triple(v_1, e_1, v_2) R_C triple(v_3, e_2, v_2)$. It follows that $e' R_E e$ and $triple(v_3, e_2, v_2)) R_C triple(v_1, e_1, v_2)$ and hence that the statement holds for $cons(e, e_1)$ and $cons(e', e_2)$.

2. Suppose that

$$\langle triple(v_1, e_1, v_2), triple(v_2, e_2, v_3)\rangle \in C \times_V C$$

and that

$$\langle triple(v_1', e_1', v_2'), triple(v_2', e_2', v_3')\rangle \in C \times_V C$$

with

$$triple(v_1, e_1, v_2) R_C triple(v_1', e_1', v_2')$$

and

$$triple(v_2, e_2, v_3) R_C triple(v_2', e_2', v_3')$$

It is then immediate from the definition of R_C that

$$triple(v_1, \mu(e_1, e_2), v_3) R_C triple(v_1', \mu(e_1', e_2'), v_3')$$

So that $comp : C/R_C \times_{V/R_V} C/R_C \longrightarrow C/R_C$ in $\bar{\bar{E}}$.

Similarly, e is well defined and then $C/R_C \rightrightarrows V/R_V$ is clearly a category, inheriting associativity and unit laws from $\bar{\bar{E}}$.

3. $R_E \subseteq R_C$, so $E/R_E \xrightarrow{\eta_1} C/R_C$ in $\bar{\bar{E}}$, so $\eta : \mathbf{G} \longrightarrow \mathbf{C}$ is a graph homomorphism in $\bar{\bar{E}}$.

Suppose now that $\mathbf{D} \equiv (D_1 \rightrightarrows D_0)$ is a category object in $\bar{\bar{E}}$ and suppose further that $\phi : \mathbf{G} \longrightarrow \mathbf{D}$ is a graph homomorphism in $\bar{\bar{E}}$. Then

$$d_1 \circ \phi_1 = \phi_1 \circ d_1 \qquad \text{and} \qquad d_0 \circ \phi_1 = \phi_1 \circ d_0$$

In view of Lemma 5.72, we can assume WLOG that these equations also hold in $\check{E}$. Then ϕ is a graph homomorphism in $\check{E}$, so there is an $\check{E}$-internal functor $\bar{\phi}$ with $\bar{\phi} \circ \eta = \phi$.

$\bar{\phi}$ is trivially a functor in $\bar{\check{E}}$ which satisfies $\bar{\phi} \circ \eta = \phi$ in $\bar{\check{E}}$. For uniqueness, suppose that ψ is a $\bar{\check{E}}$-functor $\mathbf{C} \longrightarrow \mathbf{D}$ with $\psi \circ \eta = \bar{\phi} \circ \eta$.

Since $\eta_0 \equiv id_{V/R_V}$, $\psi_0 = \bar{\phi_0}$ in $\bar{\check{E}}$.

For the arrow function, we need to show that

$$x = triple(v_1, e_1, v_2) R_C triple(v_3, e_2, v_4) = y \Rightarrow \bar{\phi}_1 R_{D_1} \psi_1(y)$$

As usual, we proceed by list induction on e_1.

Base Case: $e_1 = 0$. Then $e_2 = 0$ by the reasoning in (1) above and $\langle v_1, v_2 \rangle = \langle v_3, v_4 \rangle$, so the result is immediate.

Inductive Step: Assume that the statement is true for e_1 and suppose that

$$triple(v_1, \mu(\eta_1(e), e_1), v_2) R_C triple(v_3, \mu(\eta_1(e'), e_2), v_4)$$

Then by the definition of R_C,

$$e R_E e' \quad \text{and} \quad triple(v_1, e_1, v_2) R_C triple(v_3, e_2, v_4)$$

So $\bar{\phi}_1(\eta_1(e)) R_{D_1} \psi_1(\eta_1(e'))$ since

$$\psi \circ \eta = \bar{\phi} \circ \eta \quad \text{and} \quad triple(v_1, e_1, v_2) R_C triple(v_3, e_2, v_4)$$

So

$$m_{D_1}(\bar{\phi}_1(\eta_1(e)), \bar{\phi}_1(triple(v_1, e_1, v_2)))\ R_{D_1}$$
$$m_{D_1}(\psi_1(\eta_1(e')), \psi_1(triple(v_2, e_2, v_4)))$$

by Lemma 5.81.

It follows that

$$\bar{\phi}_1(triple(v_1, \mu(\eta_1(e), e_1), v_1)) R_{D_1} \psi_1(triple(v_3, \mu(\eta_1(e'), e_2), v_4))$$

since $\bar{\phi}$ and ψ both preserve multiplication.

Hence, $\bar{\phi} = \psi$, which completes the proof.

□

Corollary 5.83 *The full inclusion $C \longrightarrow \bar{C}$ preserves free category objects over graph objects.*

Proof Easy consequence of the above proof. □

Theorem 5.84 *$\check{\bar{E}}$ has action variants.*

Proof Suppose that $\mathbf{G} \equiv E/R_E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V/R_V$ is a graph object in $\check{\bar{E}}$ and that $\langle F, \pi, \gamma' \rangle$ is a graph diagram over $\mathbf{G}$ in $\check{\bar{E}}$.

We have by Lemma 5.72 an $\check{E}$-object W and an arrow i in $\check{\bar{E}}$ such that $V/R_V \cong W/\Delta_W$ with the property that if $A/R_A \underset{g}{\overset{f}{\rightrightarrows}} V/R_V$ commutes in $\check{\bar{E}}$ then $i \circ f = i \circ g$ in $\check{E}$. WLOG, we assume that $V \equiv W$, $R_V \equiv \Delta_W$.

Then $\langle F, \pi, \gamma \rangle$ is also a diagram in $\check{E}$ over $E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V$ and hence by Lemma 5.63 lifts to a category diagram $\langle F, \pi', \gamma' \rangle$ in $\check{E}$ over $C \underset{d_0}{\overset{d_1}{\rightrightarrows}} V$, the free category object over $E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V$.

Since $\pi \circ \gamma' = d_0 \circ \pi_1$ and $\gamma' \circ (\eta \times_{G_0} id) = \gamma$ in $\check{E}$, this is clearly the case in $\check{\bar{E}}$, so $\langle F, \pi, \gamma' \rangle$ is a diagram over $C/R_C \underset{d_0}{\overset{d_1}{\rightrightarrows}} V/\Delta_V$, the free category object over $\mathbf{G} \equiv E/R_E \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} V/R_V$ in $\check{\bar{E}}$ and we need only prove uniqueness to complete the proof. As we have chosen V canonically so that $R_V \equiv \Delta_C$, this follows since any other internal diagram satisfying the required equality would also be an internal diagram in $\check{E}$ satisfying the same equality. □

Corollary 5.85 *The full inclusion $C \hookrightarrow \bar{C}$ preserves action variants.*

Proof Follows immediately from the preservation of free category objects over graph objects and the above Lemma. □

Corollary 5.86 *$\check{\bar{E}}$ has list objects.*

Proof Immediate from Lemma 5.62. □

Chapter 6

Defining the Arithmetic Universe

We will now attempt to give a definition of "Arithmetic Universe" by analogy with the properties of $\bar{\bar{E}}$. Recall from the remarks in the introduction that the construction of $\bar{\bar{\Sigma}}$ was first undertaken by André Joyal to build a category which contained a model of itself, so that he could form categorical proofs of Gödel's results about self-referential mathematical structures. To this end, he took a simple category (the initial Skolem Theory) and added structure until he had enough to mimick the external construction of the resultant category ($\bar{\bar{E}}$) *internally* within that category.

We have seen that $\bar{\bar{E}}$ has a number of nice recursive properties, but in some sense, these were incidental to the construction - as we added the structure needed to Σ, those properties started to appear. If Arithmetic Universes are to have more general computational applications, we need to isolate the properties of $\bar{\bar{E}}$ which are most desirable and to present them as a definition. This may then generate similar structures which are not of the form $\bar{\bar{E}}$ for a Skolem Theory E.

Therefore, let us recall the properties of $\bar{\bar{E}}$:

1. $\bar{\bar{E}}$ is finitely complete;

2. $\bar{\bar{E}}$ has surjective images, which are stable under pullback;

3. $\bar{\bar{E}}$ has quotients of equivalence relations;

4. $\bar{\bar{E}}$ has finite disjoint co-products which are stable under pullback;

5. $\bar{\bar{E}}$ has free monoids;

6. $\bar{\bar{E}}$ has list objects;

7. $\bar{\bar{E}}$ has free category objects over graph objects;

8. $\bar{\bar{E}}$ has action variants.

We will start by briefly reviewing the *logical* consequences of the above.

Definition 6.1 *A category $\mathcal{C}$ which satisfies 1-4 above is called a* pre-topos.

Definition 6.2 *A first order formula is* positive *if it is built up from atomic formulae using $\exists, \wedge, \vee, \top, \bot$. It does not contain $\forall, \Rightarrow$ or $\neg$.*

Definition 6.3 *A* coherent theory *is a first order theory definable by axioms of the form*

$$\forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \Rightarrow Q(x_1, \ldots, x_n))$$

where P and Q are positive.

Coherent theories therefore have an observational nature, in the sense that the truth of their statements can be determined by a user with a finite amount of time available to him. They clearly have a relevance to computational theory. Makkai and Reyes ([MakRey 77]) showed that a pre topos is a category in which a coherent theory may be interpreted.

It therefore follows that a category with the above properties is one in which sensible recursive definitions can be formulated using coherent logic. We would therefore like to define an Arithmetic Universe as a pre-topos which satisfies 5-8 above. There remains some confusion about an appropriate minimal set of axioms. We state a few simple Lemmas:

Lemma 6.4 *Any pre-topos which satisfies (6) also satisfies (5).*

Proof This is an immediate consequence of Lemma 2.23. □

Lemma 6.5 *A pre-topos which satisfies (8) also satisfies (7).*

Proof By definition. □

Lemma 6.6 *A pre-topos which satisfies (7) also satisfies (5).*

Proof Suppose that $\mathcal{E}$ is a pre-topos with free category objects over graph objects. We claim that the free category object over the graph $A \overset{!}{\underset{!}{\rightrightarrows}} 1$ is the free monoid over A.

Let $M(A)$ be the object of arrows for the free category. Then $M(A) \times_1 M(A) \equiv M(A) \times M(A)$ and so the composition operator m in the free category is a map $M(A) \times M(A) \longrightarrow M(A)$ which is associative with unit e, where e is the unit map for the free category. It follows that $\langle M(A), m, e\rangle$ is a monoid.

Suppose that $\langle B, m_B, e_B\rangle$ is a monoid object in $\mathcal{E}$ and that $f : A \longrightarrow B$ is an arrow in $\mathcal{E}$. Then B is the object of arrows for a category with object of objects 1. It is easy to see that f is a graph homomorphism from $A \overset{!}{\underset{!}{\rightrightarrows}} 1$ to $B \overset{!}{\underset{!}{\rightrightarrows}} 1$ and that the unique induced internal functor from $M(A)$ to B is the unique required monoid homomorphsim $\langle M(A), m, e\rangle \xrightarrow{\bar{f}} \langle B, m_B, e_B\rangle$.
□

Lemma 6.7 *A pre-topos which satisfies (8) also satisfies (6).*

Proof This is Lemma 5.62. □

Here are some conjectures of non-implications, all of which are based on obstacles to proof:

Conjecture 6.8 *A pre-topos can satisfy (7) but not (8).*

Justification The justification for this remark is founded upon the apparent inability to perform inductive definitions over pullbacks in a recursive category and is discussed in the proof of Theorem 5.63. □

Conjecture 6.9 *A pre-topos can satisfy (7) but not (6).*

Justification Suppose that a pre-topos $\mathcal{E}$ has free category objects over graph objects and suppose that we do not have action variants. If $\mathcal{E}$ has list objects then they coincide with the free monoids and to show the existence of list objects, we therefore need to show that $M(A)$ as defined in the proof of Lemma 6.6 is a list object.

As we do not have action variants, the only machinery with which we can play appears to be the free category objects. Given recursive data $1 \xrightarrow{f} C \xleftarrow{g} A \times C$, we seem therefore to need a way of expressing C as the object of arrows in a category which has object of objects 1. In other words, we need to express C as a monoid object with unit f and a monoid homomorphism $\phi : M(E) \longrightarrow C$ which respects the unit and which satisfies $\phi \circ m \circ (\eta \times A) = g \circ (A \times \phi)$. This does not in general appear to be possible.

□

If this conjecture is correct,

Corollary 6.10 *A pre-topos can satisfy (5) but not (6).*

Proof Immediate from the above. □

Conjecture 6.11 *A pre-topos can satisfy (6) but not (7).*

Justification This is again based upon the apparent inability to perform recursive definitions over pullbacks and is discussed in the proof of Theorem 5.60. □

We will see in the next section that we really want to have property (8) to enable us to construct free Lex Theories. As this implies all of the other properties in our list, we formulate the following definition:

Definition 6.12 *An* Arithmetic Universe *is any pre-topos which has action variants.*

Of course, if any of the above conjectures prove to be false, we may be able to replace this with a simpler definition.

Lemma 6.13 *Any category of the form $\bar{\bar{E}}$ is an Arithmetic Universe.*

Note that with our definition for AU's, the converse need not hold - for example, **Set** is an AU but is not small. Any Arithmetic Universe of the form $\bar{\bar{E}}$ is small.

Definition 6.14 *A functor between Arithmetic Universes which preserves finite left limits, images, quotients, coproducts, list objects, free categories and action variants is called a* morphism of arithmetic universes.

Lemma 6.15 *There is a bijective correspondence between morphisms of Skolem Theories and morphisms of Arithmetic Universes of the form* $\bar{\bar{E}}$.

It follows from the above definitions that we have a category of Arithmetic Universes.

Lemma 6.16 $\bar{\bar{\Sigma}}$ *is the initial A.U.,* $\mathcal{A}_0$.

Proof Every AU $\mathcal{A}$contains the initial Skolem Theory Σ. By definition it contains images and quotients of equivalence theories and so it contains an isomorphic copy of $\bar{\bar{\Sigma}}$. □

Chapter 7

Theories in Arithmetic Universes

In this chapter we will firstly give a precise definition of the notion of a mathematical theory and we will consider the models of certain types of theories (Lex theories) in an arbitary arithmetic universe $\mathcal{A}$. In particular, we will give part of a proof that finitely presented Lex theories can be freely constructed within an Arithmetic Universe.

7.1 Theories

Broadly, we will define a theory to be a category and we will define a model of the theory to be a functor which preserves some of the properties of the theory. This approach clearly differs from the standard treatment in which a "theory" is explained in terms of a formal language with rules of deduction and axioms. Our categorical approach was first suggested by Lawvere, who considered finitary single-sorted equational theories. The work was extended to more complex theories by Freyd ([Frey72]), when he discussed essentially algebraic theories. A greater degree of generality still was achieved by Makkai and Reyes ([MakRey 77]) and later authors, but we will not draw upon their work. Our approach, using sketches from which theories may be induced, is largely derived from Barr and Wells ([BW 85]).

Definition 7.1 *A* sketch *is a 4-tuple* $\mathcal{S} = \langle \mathcal{G}, U, D, C \rangle$ *where*

1. *$\mathcal{G}$ is a graph* $G_0 \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} G_1$*;*

2. *$U : G_0 \longrightarrow G_1$ is a function which maps each $A \in G_0$ to an arrow $A \longrightarrow A \in G_1$;*

3. *D is a class of diagrams in $\mathcal{G}$;*

4. *C is a class of cones in $\mathcal{G}$.*

Definition 7.2 *A* cartesian sketch *or* FP-sketch *is one in which there are no arrows between distinct vertices in the base of any element of C.*

Definition 7.3 *A* Lex sketch *or* LE-sketch *is one in which every cone is over a finite diagram.*

Definition 7.4 *A* sketch morphism $\mathcal{S} \longrightarrow \mathcal{S}' = \langle \mathcal{G}', U', D', C' \rangle$ *is a graph homomorphism $h : \mathcal{G} \longrightarrow \mathcal{G}'$ such that :*

1. *$h \circ U = U' \circ h$;*

2. *Every diagram in D is mapped to a diagram of D';*

3. *Every cone in C is mapped to a cone in C'.*

We will use U to establish the identity arrow in our theory, D to describe the diagrams which must commute in the theory and C to describe limit cones; this will be neccessary as the graph $\mathcal{G}$ has no notion of a limit.

Definition 7.5 *If $\mathbf{C}$ is a category then the* underlying sketch $\langle \mathcal{G}, U, D, C \rangle$ *of $\mathbf{C}$ has the following elements:*

1. *$\mathcal{G}$ is the underlying graph of $\mathbf{C}$;*

2. *U is the map which picks out the identity arrows of $\mathbf{C}$;*

3. *D is the class of all commutative diagrams of $\mathbf{C}$;*

4. *C is the class of all limit cones of $\mathbf{C}$.*

Definition 7.6 *A* model *for a sketch $\mathcal{S}$ in a category $\mathbf{C}$ is a sketch morphism from $\mathcal{S}$ to the underlying sketch of $\mathbf{C}$.*

Note that a model forces all diagrams in the sketch to commute and all of the cones of the sketch to be limit cones.

Lemma 7.7 *The models of $\mathcal{S}$ in $\mathcal{C}$ form a category, $Mod(\mathcal{S},\mathcal{C})$.*

Proof Trivial - the morphisms are simply natural transformations (whose definition does not require composition in the source category). □

Definition 7.8 *The category of* graph *morphisms from $\mathcal{S}$ to $\mathcal{C}$ will be denoted $\mathcal{C}^{\mathcal{S}}$.*

Lemma 7.9 *Let $\mathcal{S}$ be a sketch with graph $\mathcal{G}$ and let Cat($\mathcal{S}$) be the category which is freely generated by $\mathcal{G}$. Then $\mathcal{C}^{\mathcal{S}}$ is equivalent to $Func(Cat(\mathcal{S}),\mathcal{C})$.*

Proof Trivial consequence of the definition of freely generated categories. □

Barr and Wells ([BW 85]) show how one may use a sketch to generate a theory for **Set** models - a cartesian sketch has an associated cartesian category; models of the theory in **Set** are finite product-preserving functors from the theory to **Set**. An LE-sketch has an associated Lex category as its theory whose models are the Lex functors into **Set**. Constructions of the theories rely upon the following:

Definition 7.10 *$\mathcal{C}$ is a* reflective subcategory *of $\mathcal{D}$ if the inclusion functor $\mathcal{C} \hookrightarrow \mathcal{D}$ has a left adjoint.*

Theorem 7.11 (Kennison's Theorem) *Let $\mathcal{S}$ be a small sketch. Then $Mod(\mathcal{S},\mathbf{Set})$ is a reflective subcategory of $\mathbf{Set}^{\mathcal{S}}$.*

Proof [BW 85, pp 146-149]. This proof and that of an extension by Freyd and Kelly ([FreyKel 72]) rely upon Freyd's Adjoint Functor Theorem; we cannot apply this in an arbitary Arithmetic Universe $\mathcal{A}$ as we cannot in general assume that $\mathcal{A}$ is complete and co-complete. □

We give two examples to illustrate the use of sketches:

Example - Theory of monoids

To sketch the theory of monoids we will require the following elements:

1. Objects $1, M, M^2, M^3$. Note that as yet M^2 is in no way a product - we have merely adopted suggestive notation for it;

2. Identity arrows are required for the definition of U;

3. Arrows $e : 1 \longrightarrow M$ and $\mu : M^2 \longrightarrow M$ will be required for the operations;

4. Projection arrows $p_1, p_2 : M^2 \longrightarrow M$ and $q_1, q_2, q_3 : M^3 \longrightarrow M$ for two cones in C with base the discrete diagram all of whose elements are M;

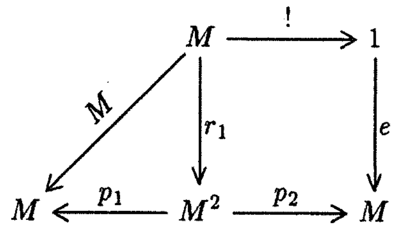
5. An arrow $M \longrightarrow 1$;

6. Arrows $r_1, r_2 : M \longrightarrow M^2$ which will be forced to be $e \times id$ and $id \times e$ respectively; recall that we must explicitly assume them as M^2 is not a product;

7. $s_1, s_2 : M^3 \longrightarrow M^2$ which will be forced to be $id \times \mu$ and $\mu \times id$;

8. $t_1, t_2 : M \longrightarrow M^2$ which will be $\langle id, e \rangle$ and $\langle e, id \rangle$ respectively.

The $\mathcal{G}$ and U are given by the above. C will contain the cones for the definition of the arrows p_i and q_i as above and an empty cone over 1. D will contain the diagrams of definition 2.2 and forcing diagrams to define $r_1, r_2, s_1, s_2, t_1, t_2$. For example, the following diagram in D will force r_1 to be $e \times id$:



Clearly, we can model any algebraic structure with finitary operations which satisfy universal equiations in this way.

We have described here an FP-sketch. We could equally regard it as an LE-sketch. It transpires ([BW 85, p. 156]) that every FP-theory has an extension to an LE-theory whose models in any Lex-category are the same as those of the FP-theory. The LE-theory of monoids will contain all the powers and arrows of the FP-theory and all constructions which can be made from those by forming finite limits, Since models of an LE-theory preserve limits, homomorphisms of models of the LE-theory of monoids in a fixed Lex category will preserve all constructions which can be made on the monoids using finite limits in the LE-theory of monoids. For example, homomorphisms of models of the LE-theory of monoids in **Set** will preserve the set $\{\langle x, y\rangle | xy = yx\}$, since this is an equaliser diagram in the Lex-theory of monoids.

As an example of Lex sketching, we have the following:

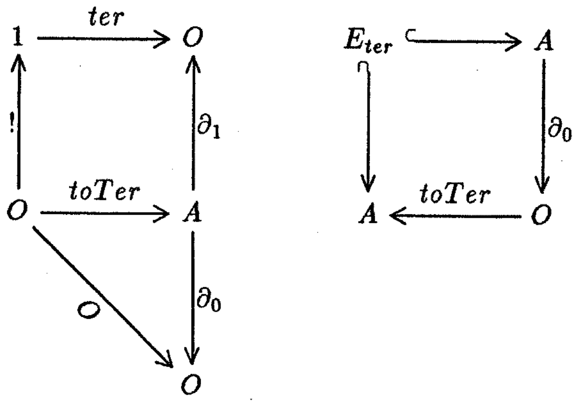
Lemma 7.12 *The theory of Skolem categories is sketchable.*

Proof Recall that the Theory of Skolem Categories is the theory of cartesian categories with Natural Numbers Objects.

We prove the Lemma in stages:

The *theory of categories* is sketchable using components $A \overset{\partial_0}{\underset{\partial_1}{\rightrightarrows}} O$ with $O \xrightarrow{i} A$ and $A_c \xrightarrow{c} O$ where A_c is forced to be the pullback of ∂_1 along ∂_0 with the obvious diagrams - these actually appear in [Mac 71, p. 49]

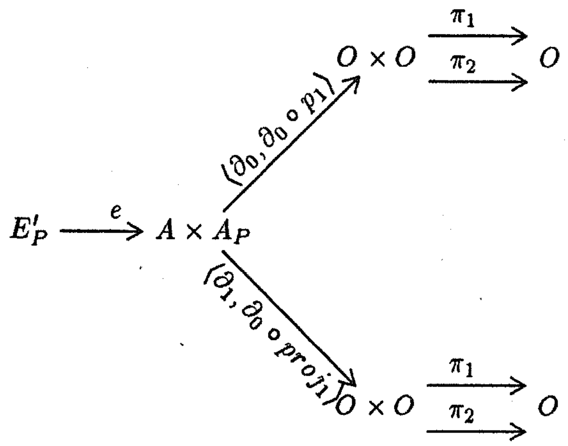
The existence of an *initial object* can be modelled; we need arrows $1 \xrightarrow{ter} Oo$ and $O \xrightarrow{toTer} A$ and $E_{ter} \xrightarrow{e} A$ where E will be the equaliser of $A \xrightarrow{!} 1 \xrightarrow{ter} O$ and $A \xrightarrow{\partial_1} O$ and we will require the commutativity of the following diagrams:



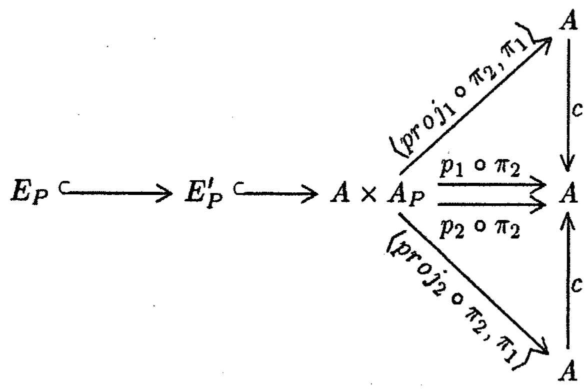
To describe *finite products* we will need arrows $O \times O \xrightarrow{prod} O \times A \times A$ and $A_P \xrightarrow{comb} A$ to represent the product formation and the generation of the unique arrow $\langle p, q\rangle$ from $B_1 \xleftarrow{p} A \xrightarrow{q} B_2$. We will force A_P to be th4e pullback of ∂_0 along ∂_0. For convenience, we define $proj_1 : A_P \longrightarrow A$ and $proj_2 : A_P \longrightarrow A$ to be the following concatenations:

$$proj_1 \equiv A_P \xrightarrow{\langle \partial_1 \circ p_1, \partial_1 \circ p_2 \rangle} O \times O \xrightarrow{prod} O \times A \times A \xrightarrow{\pi_2} A$$
$$proj_2 \equiv A_P \xrightarrow{\langle \partial_1 \circ p_1, \partial_1 \circ p_2 \rangle} O \times O \xrightarrow{prod} O \times A \times A \xrightarrow{\pi_3} A$$

We define an object $E'_P \xrightarrow{e} A \times A_P$ in $\mathcal{S}$ and we force the following to be a limit diagram:



in **Set**, E'_P would be the set of pairs $\langle s, \langle r_1, r_2 \rangle \rangle$, where $C_1 \xleftarrow{r_1} A \xrightarrow{r_2} C_2$ is a composable pair and $A \xrightarrow{s} C_1 \times C_2$ is an arrow. We then define $E_P \longrightarrow E'_P$ to be an arrow with the following forced to be a limit diagram:

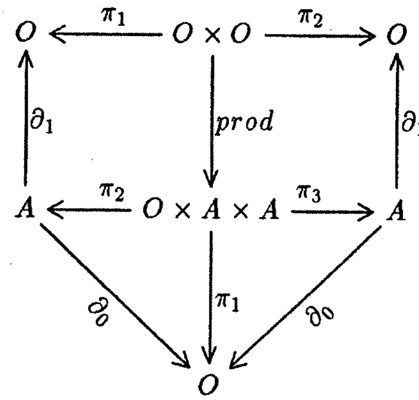


In **Set**,

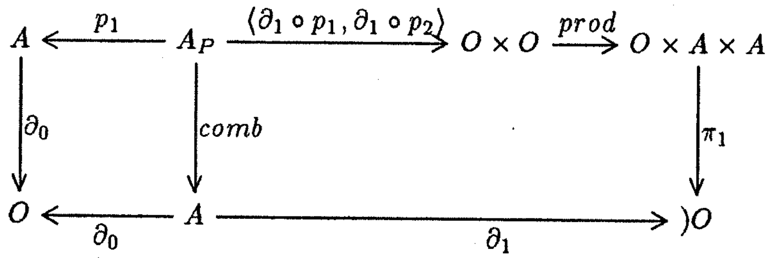
$$E_P = \{ \langle f, \langle r_1, r_2 \rangle \rangle : src(r_1) = src(r_2) = src(f), tar(f) = tar(r_1) \times tar(r_2) \\ , proj_i \circ f = r_i \}$$

The finite products will now come from the following commuting diagrams:

1. The source and targets of the projection arrows will be fixed by the commutativity of this diagram:

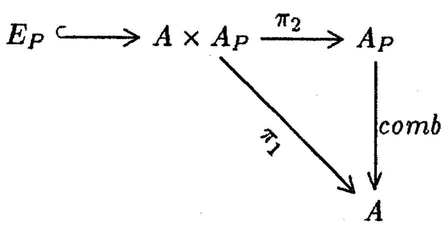


2. The source and target of $\langle p, q \rangle$ will be fixed by commutativity of the following:



3. The above will guarantee that $\partial_0 \circ proj_1 = \partial_1 \circ comb$. We then require $c \circ \langle proj_1, comb \rangle = p_1$ and $c \circ \langle proj_2, comb \rangle = p_2$, to force $\pi_i \circ \langle p_1, p_2 \rangle = p_i$.

4. We require the following diagram to commute, which willo give us the uniqueness of $\langle p, q \rangle$:



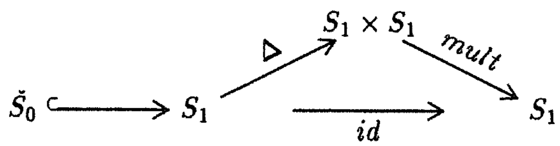
To define the Natural Numbers Object, we employ similar techniques to those used for finite products. The details are omitted. □

We can now prove in outline the following:

Theorem 7.13 *Every Arithmetic Universe contains an initial Aritmetic Universe object.*

Proof Since an Arithmetic Universe $\mathcal{A}$ is a Lex category, there will be a model from the Lex sketch of the initial Skolem Theory to $\mathcal{A}$ which will be a model of Σ in $\mathcal{A}$. We can then mimic the construction of $\check{\check{\Sigma}}$ internally. We will show how this works for the inclusion $\Sigma \longrightarrow \check{\Sigma}$; the details for the inclusion $\check{\Sigma} \longrightarrow \check{\check{\Sigma}}$ are similar and we ommit them.

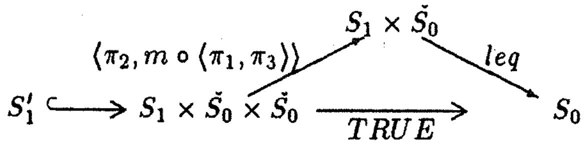
Suppose that $S_1 \overset{\partial_1}{\underset{\partial_0}{\rightrightarrows}} S_0 \xrightarrow{e} S_1$, $S_1 \times_{S_0} S_1 \xrightarrow{m} S_1$ is the model of Σ in $\mathcal{A}$. Then $\check{\Sigma}$ has as objects all predicates in Σ - the object of objects for $\check{\Sigma}$ in $\mathcal{A}$ is therefore the following equaliser:



The arrows from A to B in $\check{\Sigma}$ are equivalence classes of the set

$$\{f : N \longrightarrow N | A \leq B \circ f\}$$

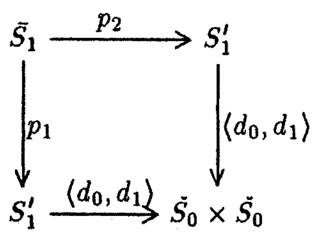
this collection is modelled in $\mathcal{A}$ as the following equaliser:



Define $d_0 \equiv S_1' \hookrightarrow S_1 \times \check{S}_0 \times \check{S}_0 \xrightarrow{\pi_2} \check{S}_0$

$d_1 \equiv S_1' \hookrightarrow S_1 \times \check{S}_0 \times \check{S}_0 \xrightarrow{\pi_3} \check{S}_0$

Then $f \sim g$ iff $A \leq eq(f,g)$. We represent $\bar{S}_1 \equiv \{\langle f,g \rangle | f,g$ have the same $src, tar\}$ by the following pullback:



Then the subset R of $\bar{S}_1$ with

$$\langle f,g\rangle \in R \Leftrightarrow src(f) \leq eq(f,g)$$

is the following equaliser:

$$R \hookrightarrow \bar{S}_1 \xrightarrow{\langle src \circ p_1, p_1, p_2\rangle} \check{S}_0 \times S_1' \times S_1' \xrightarrow{\check{S}_0 \times eq} \check{S}_0 \times S_0 \xrightarrow{leq} S_0 \qquad \bar{S}_1 \xrightarrow{TRUE} S_0$$

The arrow class of $\check{\Sigma}$ in $\mathcal{A}$ is then the coequaliser of $R \rightarrowtail \bar{S}_1$.

Completion $\check{\Sigma} \longrightarrow \check{\check{\Sigma}}$ is similar- we ommit the details. □

Note that the reason we took the exact completion of $\check{\Sigma}$ was in order that we might take the co-equaliser of the equivalence relation $R \rightarrowtail \bar{S}_1$ in the above proof - the exact completion process involves nothing more exotic than a further use of coequalisers of equivalence relations and so we stopped the construction process there.

7.2 Internal Categories in A.U.'s

Recall from Lemma 7.12 that the theory of categories is Lex-sketchable. We already know what a model of the associated theory looks like - we defined the notion of an internal category object in chapter 3; a homomorphism between models of the theory is an internal functor. We will now examine the relationship between internal categories in an Arithmetic Universe and external categories generated freely by finite graphs.

Theorem 7.14 *Suppose that $\mathcal{S} = \langle \mathcal{G}, U, D, C\rangle$ is a sketch with finite graph $\mathcal{G}$. Define Cat($\mathcal{S}$) to be the category generated by $\mathcal{G}$. Then Cat($\mathcal{S}$) exists as an internal category* **C** *in an arbitary Arithmetic Universe $\mathcal{A}$ in the sense that external functors Cat($\mathcal{S}$) $\longrightarrow \mathcal{A}$ are in bijective correspondence with category actions (F, π, μ) in $\mathcal{A}$.*

Proof $\mathcal{G}$ is finite. We can write $\mathcal{G}_0$ as $\{g_1, \ldots, g_n\}$ and $\mathcal{G}_1$ as $\{f_1, \ldots, f_m\}$. We represent $\mathcal{G}$ in $\mathcal{A}$ by $\mathbf{G}_0 = 1+1+\ldots+1$ (n times) and $\mathbf{G}_1$ by $1+1+\ldots+1$ (m times) with $\partial_0, \partial_1 : \mathbf{G}_1 \rightrightarrows \mathbf{G}_0$ the obvious projections.

Since Cat($\mathcal{S}$) is free over $\mathcal{G}$, functors $Cat(\mathcal{S}) \longrightarrow \mathcal{A}$ are entirely determined by their action as graph homomorphisms $\mathcal{G} \longrightarrow \mathcal{A}$, so it will suffice to show that these correpond to internal **G**-actions - we then obtain our correspondence by applying the A.U. definition to obtain a unique category action over $Free(\mathbf{G})\stackrel{def}{=}\mathbf{C}$ for each graph action over **G**.

Suppose that $H : \mathcal{G} \longrightarrow \mathcal{A}$ is a graph homomorphism.

Let $F\stackrel{def}{=}H(g_1) + \ldots + H(g_n)$.

Define $\pi : F \longrightarrow \mathbf{G}_0$ to be $!+!+\ldots+!$ (n times).

To define an arrow $G_1 \times_{G_0} F \longrightarrow F$, note that by distributivity and the finiteness of $\mathcal{G}$ that this is a finite sum of terms $1_j \times H(g_i)$, where the subscript on the 1 is included to indicate its providence. We therefore need an arrow for each such term; as there are finitely many of them we can clearly combine them to give us the desired arrow.

For a term $1_j \times H(g_i)$, where 1_j corresponds to $f_j : g_i \longrightarrow g_k$, we define μ by the following sequence of arrows:

$$1_j \times H(g_i) \xrightarrow{\pi_2} H(g_i) \xrightarrow{H(f_j)} H(g_k) \xrightarrow{i_k} F$$

Then $\pi \circ \mu = i_k \circ ! = d_0 \circ \pi_1$. We therefore have that (F, π, μ) is a $\mathcal{G}$-action; this mapping is clearly bijective so the theorem is proved. □

Lemma 7.15 *If $\mathcal{S}_C$ and $\mathcal{S}_D$ are sketches with finite graphs $\mathcal{G}_C$ and $\mathcal{G}_D$ and* **C** *and* **D** *are the internal representations of Theorem 7.14 for $Cat(\mathcal{S}_C)$ and $Cat(\mathcal{S}_D)$ respectively then functors $Cat(\mathcal{S}_C) \longrightarrow Cat(\mathcal{S}_D)$ are in bijective correspondence with internal functors* $\mathbf{C} \longrightarrow \mathbf{D}$.

Proof This is trivial. Any functor $F : Cat(\mathcal{S}_C) \longrightarrow Cat(\mathcal{S}_D)$ is uniquely determined by its behaviour as a graph homomorphism $\mathcal{G}_C \longrightarrow \mathcal{G}_D$. Suppose that $Ob(\mathcal{G}_C) = \{g_1, \ldots, g_n\}$, $Ar(\mathcal{G}_C) = \{f_1, \ldots, f_m\}$ and that $Ob(\mathcal{G}_D) = \{g'_1, \ldots, g'_{n'}\}$, $Ar(\mathcal{G}_C) = \{f'_1, \ldots, f'_{m'}\}$ with $F(g_i) = g'_{s_i}$ and $F(f_i) = f'_{t_i}$. Then F is represented on **C** by $\bar{F}$, where

$$\bar{F}_0 = [i_{s_1} \circ !, \ldots, i_{s_n} \circ !] \qquad \text{and} \qquad \bar{F}_1 = [i_{t_1} \circ !, \ldots, i_{t_m} \circ !]$$

It is easy to show that this correspondence is bijective. □

We now draw upon chapter 2 of [John 77].

Lemma 7.16 *Let $\mathcal{A}$ be any finitely complete category. Let $\langle F_0, \pi, \mu \rangle$ be an internal diagram on an object* **C** *of $Cat(\mathcal{A})$. Define F_1 to be $C_1 \times_{C_0} F$, the pullback of $C_n \xrightarrow{\partial_0} C_0$ along $F \xrightarrow{\pi} C_0$. Then F_1, F_0 are the arrow and* *object classes respectively of an internal category* $\mathbf{F}$ *and* $\pi : F_0 \longrightarrow C_0$, $\pi_1 : F_1 \longrightarrow C_1$ *is an internal functor* $\mathbf{F} \longrightarrow \mathbf{C}$.

Proof

Define $d_0 : F_1 \longrightarrow F_0 \equiv C_1 \times_{C_0} F \xrightarrow{p_2} F$

$d_1 : F_1 \longrightarrow F_0 \equiv C_1 \times_{C_0} F \xrightarrow{\mu} C_1$

$m \equiv (C_1 \times_{C_0} F) \times_{F_0} (C_1 \times_{C_0} F) \xrightarrow{\langle compo\circ\langle p_1\circ p_1, p_1\circ p_2\rangle, p_2\circ p_1\rangle} C_1 \times_{C_0} F$

$e \equiv F_0 \xrightarrow{\langle e\circ\pi, id\rangle} C_1 \times_{C_0} F_0$

It is easy to check that $\langle F_0, F_1, d_0, d_1, m, e\rangle$ is a category in $\mathcal{A}$ and that $\langle \pi, \pi_1\rangle$ is an internal functor: that the functor respects the source of F-arrows is by definition of the pullback $C_1 \times_{C_0} F_0$ and the definition of μ forces the functor to respect targets. Commutativity of the unit and composition diagrams is trivial. □

Definition 7.17 *An object* $\pi : \mathbf{F} \longrightarrow \mathbf{C}$ *of* $Cat(\mathcal{A})/\mathbf{C}$ *is called a* discrete opfibration *if the following diagram is a pullback:*

$$\begin{array}{ccc} F_1 & \xrightarrow{d_0} & F_0 \\ {\scriptstyle \pi_1}\downarrow & & \downarrow{\scriptstyle \pi_0} \\ C_1 & \xrightarrow{d_0} & C_0 \end{array}$$

Lemma 7.18 *An object* $\mathbf{F} \xrightarrow{\pi} \mathbf{C}$ *of* $Cat(\mathcal{A})/\mathbf{C}$ *is isomorphic to one which has been constructed from an internal diagram over* $\mathbf{C}$ *using the mechanism of Lemma 7.16 iff it is a discrete opfibration.*

Proof Trivial. □

We can therefore identify up to equivalence $IAct(\mathbf{C}, \mathcal{A})$ with the full subcategory of $Cat(\mathcal{A})/\mathbf{C}$ whose objects are the discrete opfibrations.

The following is [John 77, Lemma 2.19]:

Lemma 7.19 *Let* $\mathbf{C} \xrightarrow{f} \mathbf{D}$ *be a morphism of* $Cat(\mathcal{A})$. *The pullback functor* $f^* : Cat(\mathcal{A})/\mathbf{D} \longrightarrow Cat(\mathcal{A})/\mathbf{C}$ *preserves discrete opfibrations and so induces a functor* $f^* : IAct(\mathbf{D}, \mathcal{A}) \longrightarrow IAct(\mathbf{C}, \mathcal{A})$.

In the case where $\mathcal{A} = \mathbf{Set}$, this corresponds to the right composition functor $Func(\mathbf{D}, \mathcal{A}) \longrightarrow Func(\mathbf{C}, \mathcal{A})$ induced by f. When $\mathcal{C}$, $\mathcal{D}$ are free categories over finite graphs and $F : \mathcal{C} \longrightarrow \mathcal{D}$ is a functor,

$$F^* : Func(\mathcal{D}, \mathcal{A}) \longrightarrow Func(\mathcal{C}, \mathcal{A})$$

corresponds to

$$f^* : IAct(\mathbf{D}, \mathcal{A}) \longrightarrow IAct(\mathbf{C}, \mathcal{A})$$

which is induced by the internal representation $\mathbf{C} \xrightarrow{f} \mathbf{D}$ of F in $\mathcal{A}$.

The following theorem is [John 77, Theorem 2.34]. Its proof is rather lengthy and adds nothing to our exposition, so we ommit it.

Theorem 7.20 *If $\mathcal{A}$ has finite limits and reflexive coequalisers which are preserved under pullback then the f^* which is induced from a morphism $\mathbf{C} \xrightarrow{f} \mathbf{D}$ of $Cat(\mathcal{A})$ has a left adjoint $f_! : IAct(\mathbf{C}, \mathcal{A}) \longrightarrow IAct(\mathbf{D}, \mathcal{A})$.*

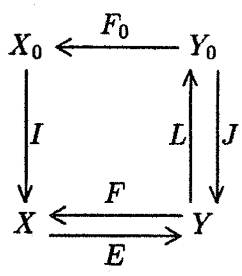
If $\mathcal{A}$ is an Arithmetic Universe then it trivially satisfies these preconditions.

Corollary 7.21 *If $\mathcal{A}$ is an Arithmetic Universe, $\mathcal{C}$ and $\mathcal{D}$ are categories freely generated by finite graphs and $F : \mathcal{C} \longrightarrow \mathcal{D}$ is a functor then the induced forgetful functor $F^* : Func(\mathcal{D}, \mathcal{A}) \longrightarrow Func(\mathcal{C}, \mathcal{A})$ has a left adjoint $F_! : Func(\mathcal{C}, \mathcal{A}) \longrightarrow Func(\mathcal{D}, \mathcal{A})$.*

Proof F corresponds to an internal functor $f : \mathbf{C} \longrightarrow \mathbf{D}$ by Lemma 7.15. This induces an internal functor $f^* : IAct(\mathbf{D}, \mathcal{A}) \longrightarrow IAct(\mathbf{C}, \mathcal{C})$ which has a left adjoint $f_! : IAct(\mathbf{C}, \mathcal{A}) \longrightarrow IAct(\mathbf{D}, \mathcal{A})$. This corresponds to a left adjoint $F_! : Func(\mathcal{C}, \mathcal{A}) \longrightarrow Func(\mathcal{D}, \mathcal{A})$ to F^*. □

We now borrow from [BW 85, p. 151].

Lemma 7.22 *In the following diagram of categories and functors, suppose that $L \dashv J$, I is full and faithful, $E \dashv F$ and $F \circ J$ is naturally equivalent to $I \circ F_0$. Then $E_0 \stackrel{def}{=} L \circ E \circ I \dashv F_0$.*



Proof

$$Hom(L \circ E \circ Ix_0, y_0) \cong Hom(E \circ Ix_0, Jy_0) \cong Hom(Ix_0, F \circ Jy_0)$$
$$isomHom(Ix_0, I \circ F_0 y_0) \cong Hom(x_0, F_0 y_0)$$

□

We now make a conjecture which is analogous to Kennison's Theorem (7.11):

Conjecture 7.23 *Let $\mathcal{S}$ be a finite LE-sketch and let $\mathcal{A}$ be an Arithmetic Universe. Then $Mod(\mathcal{S}, \mathcal{A})$ is a reflective subcategory of $\mathcal{A}^{\mathcal{S}}$.*

A proof of the above has so far eluded me - we want to freely make the diagrams of $\mathcal{S}$ commute and the cones to be limits in $\mathcal{A}$ for the image of any graph morphism $\mathcal{G} \longrightarrow \mathcal{A}$. It certainly *appears* possible - one can see how to deal with very simple cases. For example, if an element f of $\mathcal{A}^{\mathcal{S}}$ maps the graph $A \longleftarrow C \longrightarrow B$ which is a cone of $\mathcal{S}$ to $D \longleftarrow E \longrightarrow F$ in catA then the unit η of the adjunction ahould send f to the element of $Mod(\mathcal{S}, \mathcal{A})$ which maps the graph to $D \longleftarrow D \times F \longrightarrow F$. However, if this is only a subgraph of $\mathcal{G}$ and $A \longleftarrow C \longrightarrow B$ also forms a part of a commuting diagram in $\mathcal{A}$ then the situation is more complicated; we have to define the image of any subgraph which includes arrows with source or target C and this may disturb other cones in the graph. However, as the source graph is finite, I think that the problem may succumb eventually to an inductive approach which recursively defines η at each stage of the induction.

Subject to the truth of the Conjecture, we can prove that $\mathcal{A}$ contains freely generated finitely presented Lex theories, in the following sense:

Corollary 7.24 *Suppose that $\mathcal{S}_1$ and $\mathcal{S}_2$ are finite Lex sketches , that $f : \mathcal{S}_1 \longrightarrow \mathcal{S}_2$ is a sketch morphism and that $\mathcal{A}$ is an Arithmetic Universe. Then the map $f^* : Mod(\mathcal{S}_2, \mathcal{A}) \longrightarrow Mod(\mathcal{S}_1, \mathcal{A})$ has a left adjoint.*

Proof The map f induces an arrow $f^* : \mathcal{A}^{\mathcal{S}_2} \longrightarrow \mathcal{A}^{\mathcal{S}_1}$ which restricts to a map with the same name from $Mod(\mathcal{S}_1, \mathcal{A})$ to $Mod(\mathcal{S}_2, \mathcal{A})$. By the equivalence of Lemma 7.9, $f^* : \mathcal{A}^{\mathcal{S}_2} \longrightarrow \mathcal{A}^{\mathcal{S}_1}$ can be regarded as an arrow

$$f'^* : Func(Cat(\mathcal{S}_2), \mathcal{A}) \longrightarrow Func(Cat(\mathcal{S}_1), \mathcal{A})$$

which is induced by the arrow $f' : Cat(\mathcal{S}_1) \longrightarrow Cat(\mathcal{S}_2)$ which is derived from f. By Corollary 7.21, f'^* has a left adjoint $f'_!$. By the equivalence 7.9, f^* has a left adjoint $f_! : \mathcal{S}_1^{\mathcal{A}} \longrightarrow \mathcal{S}_2^{\mathcal{A}}$. We can therefore draw the following diagram of categories and functors:

$$\begin{array}{ccc}
Mod(\mathcal{S}_1, \mathcal{A}) & \xleftarrow{f^*} & Mod(\mathcal{S}_2, \mathcal{A}) \\
\uparrow \dashv \downarrow & & \uparrow \dashv \downarrow \\
\mathcal{S}_1^{\mathcal{A}} & \underset{f^*}{\overset{f_!}{\rightleftarrows}} & \mathcal{S}_2^{\mathcal{A}}
\end{array}$$

where the adjunction between the vertical arrows is that of Conjecture 7.23. Then by Lemma 7.22, f^* has a left adjoint. □

Chapter 8

Conclusions

This project was an attempt to find an appropriate definition for an Arithmetic Universe and to determine the equivalence or non-equivalence of the list of definitions which were mentioned in the introduction and then listed in chapter 6.

After carefully defining the other constructs in the list, we proved that Joyal's initial Arithmetic Universe $\bar{\bar{\Sigma}}$ has all of the properties which appear in the list and demonstrated that by defining an Arithmetic Universe in general as a pre-topos with action variants we guarantee the presence of all of the recursive properties which $\bar{\bar{\Sigma}}$ has. While checking that $\bar{\bar{\Sigma}}$ posesses these properties we made a number of conjectures which are listed in chapter 6 and which together imply that this definition is the minimal one.

All of our conjectured non-equivalences are based upon the apparent impossibility of performing a general recursive definition of an arrow ϕ with the following form in a List-Arithmetic category:

$$\begin{array}{ccccc} 1 & \longrightarrow & D & \xleftarrow{cons} & A \times_{D_0} D \\ & \searrow^{f} & \downarrow{\phi} & & \downarrow{A \times \phi} \\ & & C & \xleftarrow{g} & A \times_{C_0} C \end{array}$$

The difficulties here are discussed in chapter 5 and they all relate to the problems associated with building in pre-conditions via pullbacks in recursive definitions. Regrettably, we were not able to produce a counter-example which would prove this difficulty to be insurmountable and this is one of the items which requires further work.

We stated Steve Vicker's conjecture concerning finitely-generated Lex Theories in the Introduction and one of the goals of this project was to prove it. In chapter 7 we split the work into two parts. We managed to prove one part by using a theorem which allows us to represent a category $\mathcal{C}$ freely generated by a finite graph in an Arithmetic Universe $\mathcal{A}$ in the sense that the internal functors between the internal representation and those of other finitely generated categories and the diagrams over it correspond exactly to external functors between finitely generated categories and $\mathcal{C}$ and functors $\mathcal{C} \longrightarrow \mathcal{A}$ respectively.

The proof of the second half is still outstanding, although we conjectured its truth. In view of the remarks in chapter 7, if the conjecture *is* true then it is likely to be because the sketches concerned are finite and not because $\mathcal{A}$ is an Arithmetic Universe. As we stated in chapter 7, all previous work devoted to similar theories has required the cocompletenss of $\mathcal{A}$ so that Freyd's Adjoint Functor Theorem can be applied. It seems likely that the existence of the left adjoint which we seek will have to be demonstrated constructively and so a different angle of attack will be required.

The other remaining unproved conjecture was the correspondence theorem for the semantics of PRIM in a Skolem Theory. Demonstrating this could perhaps give us a route into a categorical treatment of computability theory. This would be in line with Joyal's original approach to the subject, which was motivated by a desire to prove Gödel's incompleteness theorem without recourse to the concrete numbering schema which bedevil traditional recursion theory.

I have not touched here upon the interpretation of programs with WHILE loops in Arithmetic Universes. [GCW Notes] makes some interesting remarks about the interpretability of such programs in the category of partial arrows of an Arithmetic Universe. It would be interesting to develop a denotational sematics for such programs in this context and to prove some sort of correspondence result. This would give us a deeper understanding of the categorical issues here and could conceivably connect these categories to Domain Theoretic work.

In all of the above, the ability to interpret coherent logics in Arithmetic Universes has largely been ignored. One potential application of all of this in Computer Science has been suggested by Steve Vickers; since AU's give us all of the computational power that we need (modulo the proposed semantic work in $Part(\mathcal{A})$), one could employ them to give the semantics of a specification language which gave the specifier exactly enough power to specify coherently and no more.

In summary, I am painfully aware that in conducting this work I have only scratched the surface of a part of category theory which has the potential to shed light upon some foundational issues and which is also of real potential value to computer scientists. The above paragraphs give a small sample of some of the possibilities for further work; I am sure that more will be uncovered.

Bibliography

[BW 85] M.Barr, C.Wells *Toposes, Triples and Theories*, Springer-Verlag, Berlin 1985

[CLW 93] A. Carboni, S. Lack, R.F.C. Walters *Introduction to extensive and distributive categories* J. Pure and Applied Algebra **84** (1993) pp 145-158

[Co 90] J.R.B. Cockett *List-Arithmetic Distributive Categories: Locoi* J. Pure and Applied Algebra, **66** (1990) pp 1-29.

[Frey72] P. Freyd *Aspects of Topoi* Bull. Austral.Math. Soc. **7** (1972) pp. 1-72

[FreyKel 72] P. Freyd, G.M. Kelly *Categories of Continuous Functors* J. Pure and Applied Algebra **2** (1972) pp 169-191

[FreySce 90] P. Freyd, A. Scedrov *Categories, Allegories* North Holland

[Jay 93] C.B. Jay *Tail recursion through universal invariants* Theoretical Computer Science **115** (1993) pp 151-189

[John 77] P.T. Johnstone *Topos Theory* Academic Press, New York

[MakRey 77] M.Makkai, G.E. Reyes *First-Order Categorical Logic*, Springer LNM 611, Springer-Verlag, Berlin, 1977

[McL 94] C. McLarty *Elementary Categories, Elementary Toposes* OUP

[Mac 71] S. MacLane *Categories for the Working Mathematician* Springer Verlag

[GCW Notes] G.C. Wraith *A.U. Notes* Unpublished notes about Arithmetic Universes

Action variants - 5.61
AU - 6.12