

# SEMANTICS FOR THE $\lambda$ -CALCULUS

ARNOUD VAN DER LEER  
DELFT UNIVERSITY OF TECHNOLOGY  
ARNOUDVANDERLEER@GMAIL.COM

## 1. INTRO

Hi, it is so good to see you all here. For those who don't know me, I am Arnaud van der Leer, master's student at TU Delft, and am currently working on my master's thesis, supervised by Benedikt and Kobe.

Today, I am going to talk about the paper 'Classical lambda calculus in modern dress'. This is a paper from 2013, written by Cambridge professor Martin Hyland. In this paper, he talks about models for the  $\lambda$ -calculus, and proves three 'big' theorems about them. I will also talk a little bit about my thesis, which revolves around this paper. My job is to 'annotate' this paper, so to speak.

## CONTENTS

1. Intro	1
2. Talking about the $\lambda$ -calculus	2
2.1. Models	2
2.2. Semantics	3
3. The main theorems	3
3.1. Scott's representation theorem	3
3.2. The fundamental theorem of the $\lambda$ -calculus	3
3.3. The category of retracts	3
4. My contribution	3
4.1. Annotating the paper	3
4.2. Mechanization	3
5. Conclusion	3

So here is the outline of my talk. We are currently in the introduction. To prove things about the  $\lambda$ -calculus and its models, we first need establish the way in which we will formally talk about the  $\lambda$ -calculus. We will also talk a bit about denotational semantics. Then we will have a look at the three main theorems. The first main theorem is Scott's representation theorem, which shows that every model of the  $\lambda$ -calculus can be obtained from some category. Hyland calls the second main theorem the fundamental theorem of the  $\lambda$ -calculus. This is a somewhat stronger result, which says that there is an equivalence between models and denotations for the  $\lambda$ -calculus. For me, the last main theorem seems to be the most cryptic one, to me. It states that if we have a model for the  $\lambda$ -calculus, then create its 'category of retracts', then this category is 'relatively cartesian closed'. It basically says: if we

have a model for the  $\lambda$ -calculus, we can create a category in which we can do some form of dependent type theory. Lastly, I will talk a bit about my contribution to this, so annotating the paper, and especially mechanizing it.

## 2. TALKING ABOUT THE $\lambda$ -CALCULUS

So, first of all, in this talk, I will talk almost exclusively about the **untyped**  $\lambda$ -calculus. The untyped  $\lambda$ -calculus essentially is about collections that consist of only functions. With the objects of these collections, (so functions), we can do a couple of things. First of all, we can create new objects, which are just the ‘variables’. Then, given two of these objects, we can apply one to the other. Lastly, given a function with  $n + 1$  free variables, we can create a function with  $n$  free variables by abstracting.

Again, note that everything here is a function.

We can impose two types of equalities on these terms:  $\beta$ -equality and  $\eta$ -equality. One is about what happens when you first abstract and then apply. The other is about what happens if you first apply to a variable and then abstract again.

So a model for the  $\lambda$ -calculus is something that exhibits this structure. Something that has variables, application and abstraction, and maybe  $\beta$  and/or  $\eta$ -equality.

One such collection is the *pure*  $\lambda$ -calculus. One can view this as an inductive type, given by these three constructors. We can also create the pure  $\lambda$ -calculus with  $\beta$ -equality, which arises from this inductive type via a quotient.

**2.1. Models.** So, the first stepping stone in talking formally about models are algebraic theories. These are structures with variables and some substitution operation.

One example of this is the  $\Lambda$ -calculus.

Another example is a polynomial ring. It has variables, you can add and multiply polynomials together. And if you have two polynomials, you can substitute one in the place of a variable in the other to get a new polynomial.

The formal definition of an algebraic theory is a sequence of sets  $T_n$ , you can think of those as containing the terms in  $n$  free variables, with variables  $x_1, \dots, x_n$  in set  $n$ , and a substitution. You can view the substitution as taking a term with  $m$  free variables, and  $m$  terms with  $n$  free variables, and substituting the  $g$ ’s for the free variables in  $f$ .

I have seen structures like algebraic theories in a lot of literature. They are known by different names: Lawvere theories, cartesian operads, abstract clones or algebraic theories.

Now we are able to start talking formally about models for the  $\lambda$ -theory, by extending the notion of algebraic theory. A  $\lambda$ -theory is an algebraic theory, together with abstraction functions, that take a term with  $n + 1$  free variables and give a term with  $n$  free variables, and application functions that go the other way around.

A somewhat important example of a  $\lambda$ -theory is the pure  $\lambda$ -calculus.

We can also consider models with  $\beta$ - and/or  $\eta$ -equality, by saying that  $\lambda$  after  $\rho$  or the other way around must be the identity.

I feel like  $\lambda$ -theories are a lot less known than algebraic theories. They were not mentioned in the literature that I saw about algebraic theories, and I have only seen the term in this specific paper.

**2.2. Semantics.** Now we will talk a bit about semantics, interpretations or denotations, because we want to interpret the terms in our algebraic theories as functions on some set. More specifically, we want to interpret the terms in  $n$  variables as functions that take  $n$  elements of a set and give an element of the set again.

For example, in the case of our polynomial ring, we can take as the set the rational numbers, and a polynomial in  $n$  variables then gives an  $n$ -ary function.

The formal definition is ...

### 3. THE MAIN THEOREMS

Now we get to the main theorems.

**3.1. Scott's representation theorem.** The first theorem, Scott's representation theorem, states that (...).

**3.2. The fundamental theorem of the  $\lambda$ -calculus.**

**3.3. The category of retracts.**

### 4. MY CONTRIBUTION

**4.1. Annotating the paper.**

**4.2. Mechanization.**

### 5. CONCLUSION