COMBINATORS AND CLASSES

by

Dana Scott

Oxford University

Abstract. The paper tries to answer the question: What is the relation between class abstraction and λ-abstraction in models for the λ-calculus?

Introduction. It seems fair to say that one of the original motivations for the study of λ-calculus and the combinators was to give a foundation for logic. The pure combinators were to be very general operators which would provide an analysis of "substitution" and the behaviour of variables. (An all-pervading confusion between use and mention haunts the literature on λ-calculus. Perhaps it would make more sense to say that the purpose of the combinators is to analyze the notion of functional dependence by producing a few basic combinators from which the others could be explicitly defined. And, besides this expressive power, it was also necessary to uncover the laws - usually identities - that hold among the combinators.) Adjoined to these general operators, we were then to have the quantifiers and logical connectives to be used in the analysis of propositions. All of this was to turn out to be a Fregian paradise of "type-free" functions. Alas, the first workers hardly had time to savour the forbidden fruit before they were turned out of this paradise by the discovery of the usual paradoxes. A cruel fate, but in retrospect, it is not a very surprising one.

In one way or the other we have all been trying to get back into this lost paradise, though some have been content to pause along the way to play with the non-paradoxical, pure λ-calculus part of the subject. It is a pity, however, to have a study without real motivation; and without the propositional component of the theory it is not all that easy to explain the interest in the combinators. This is not to say that they are not fun; but aside from formal amusement, what is the point? It is no good pointing to combinatory arithmetic, because there are far easier ways of explaining recursive number-theoretic functions.

There are two good answers which establish the mathematical value of the combinators. The first was given by Kleene when he defined in number theory $\{e\}(n)$, which certainly is a kind of type-free application. (We should also note that Kleene also defined $\{\alpha\}(\beta)$ for arbitrary number-theoretic functions α and β (see the Kleene-Vesley book). This application too has an interesting theory.) Kleene introduced many techniques from λ-calculus into recursive function theory by defining whatever he needed in terms of Gödel numbers. More recently we speak of the URS of Wagner-Strong which gives an abstract version of Kleene's idea (see the paper in this volume by Barendregt). But these structures have not been adequately related to "traditional" λ-calculus, because a URS is only a partial algebra in that the application $\{e\}(n)$ is not always defined.

Models in which application is always meaningful were discovered by the present author. The first were the D_∞ models, and the second kind (building on an idea of Plotkin) were the graph models, like $P\omega$. We shall not need any detail about the construction of such models in this paper except to note that they all satisfy these basic axioms:

$$(\alpha) \qquad \lambda x.\tau = \lambda y.\tau[y/x]$$

$$(\beta) \qquad (\lambda x.\tau)(y) = \tau[y/x]$$

$$(\xi) \qquad \lambda x.\tau = \lambda x.\sigma \leftrightarrow \forall x.\tau = \sigma \quad .$$

We can call these the axioms of extensional λ-calculus. The fact that the specific models mentioned above satisfy much more than what is implied by (α), (β), (ξ) is not relevant here, as everything we want to say can be done in the context of this rather weak system. (That term models for these axioms were already known as a consequence of the Church-Rosser Theorem is also irrelevant for the reasons pointed out in the author's other paper in this volume.)

Before going further, we should stress that the kind of truth definition that we shall propose in this paper could also be done for the URS. We stick to λ-calculus of the traditional sort, because it is more familiar and it illustrates a general idea neatly.

What was learned from the construction of the λ-calculus models is that there are many coherent notions of function and functional application which admit the closure properties demanded by the combinators. In all of these examples paradox is avoided because the functions used are of a limited sort and are far from arbitrary (e.g. the space of continuous functions rather than the full function space in the usual sense of set theory). Conflict is resolved by eliminating operators of an infinitary nature (like the quantifiers!) The trick was to retain enough to have the (pure) combinators. It was possible; and it was good for λ-calculus, but bad for logic, because it was just the propositional notions that had been eliminated. There was a gain for recursion theory, since the extensional combinators are less messy than Kleene's and ideas could be applied more abstractly. But still the original motivation was definitely not regained.

The one person who had climbed farthest back into "paradise" was Fitch. For one reason or the other his ideas are not very well known, partly because his presentations are highly formal and rather complicated. There is no claim here to have correctly interpreted his program, but his method was the direct inspiration. In somewhat different contexts both Feferman and Aczel have used the plan to advantage and have found connections with other theories, but their papers are not yet published. (As the method is very closely related to iterated truth definitions similar to Kleene's hyperarithmetic hierarchy, they may not feel that their inspiration can be traced directly to Fitch, though Feferman comments on Fitch's system and indicates how his own has advantages.) Earlier, in work with Fitch, Myhill had pusued the idea and mentioned it in conversation. The author at this point cannot remember whether Myhill came to any definite conclusions and cannot recall what he published about it. But the problem of priority is not very acute: Fitch started in the mid-thirties and gets the main credit. There are many variations, and we have to try to judge which of the roads back into paradise (if any!)are not a dead ends.Maybe they all are. The purpose of this paper is to encourage more exploration.

It comes to mind that Chuch mentions in his little booklet on λ-calculus that there were some lecture notes on a hierarchy of quantifiers to be added to his system. These notes were not published; and though he spent some time at Princeton, the author does not remember seeing them and does not know whether they are relevant to the present discussion. By the time Church published his monograph one feels that he had lost interest in giving any foundation <u>via</u> λ-calculus of the untyped kind. No one seems to have tried to follow up Church's ideas. As all the principals are still alive (except for Turing, of course), someone should perhaps do some historical investigation. It is not always so easy to deduce from the writings what was intended. In particular the recent publication by Fitch of "Elements of Combinatory Logic" is very disappointing in that he did not try to make a uniform exposition of his papers in the JSL which contain the details of the consistency proof (truth definition). He proposed over the years several different systems, and we could have hoped to see a complete, final version.

One conclusion we might reach in this paper is that the terminology "combinatory logic" is still premature dispite all the works of Curry and Fitch. We shall certainly establish connections with the usual kind of predicate logic, but it seems to this author that much remains to be done to determine whether these are the right connections or even especially useful ones. And the question of whether we have foundations in this way should also remain open.

§ 1. Syntax. In the background we are assuming that we have any non-trivial model for (α), (β), (ξ). "Non-trivial" here means a domain of individuals (or objects) of at least two elements. The axioms are all schematic, and the terms used (the τ and σ) are just pure λ-terms built up from variables by application and λ-abstraction.

We shall have to make a very conscious effort to avoid confusion between use and mention because we are going to formalize the syntax in the model. We shall not be quite as rigorous as Quine in keeping the distinction, but we shall be rigorous enough. The point is that certain objects of the model are going to represent formulas somewhat in the style of Gödel numbers. As we have an abstract model, however, we do not speak of its elements as numbers. Nevertheless, certain elements can be chosen to represent numbers, and we regard 0,1,2,3,4 as distinct elements of our model. (The choice of representation is not important. This is a standard construction as in the Curry volumes.)

Aside from the numbers we need combinators to form tuples like $\langle a,b,c\rangle$. Again, just how the tuple is defined is not important, though to save notation we will assume that a construction is effected whereby:

$$\langle a,b,c \rangle (0) = a \ ,$$

$$\langle a,b,c \rangle (1) = b \ ,$$

$$\langle a,b,c \rangle (2) = c \ ,$$

and similarly for other size tuples. We shall often write "u_k" for "$u(k)$", especially for numerical subscripts.

<u>Definition</u>. The <u>primitive</u> formula constructs are represented as follows:

$$\begin{array}{lcl} a = b & = & \langle 0,a,b \rangle \\ \forall x.\varphi & = & \langle 1,\lambda x.\varphi \rangle \\ \sim \varphi & = & \langle 2,\varphi \rangle \\ \varphi \wedge \psi & = & \langle 3,\varphi,\psi \rangle \\ \varphi \times\!\!\times \psi & = & \langle 4,\varphi,\psi \rangle \end{array}$$

We have tried to choose as few primitives as possible without being incomprehensible. Still we need various defined operations as well in order to make formulas look normal and familiar.

<u>Definition</u>. The <u>defined</u> formula constructs are represented as follows:

$$\begin{array}{lcl} a \in b & = & b(a) \\ \exists x.\varphi & = & \sim \forall x. \sim \varphi \\ \varphi \vee \psi & = & \sim [\sim \varphi \wedge \sim \psi] \\ \varphi \rightarrow \psi & = & \sim \varphi \vee \psi \\ \varphi \leftrightarrow \psi & = & [\varphi \wedge \psi] \vee [\sim \varphi \wedge \sim \psi] \\ T & = & 0 = 0 \\ F & = & \sim T \\ * & = & T \times\!\!\times F \end{array}$$

By way of example consider the formula:

$$\exists x.\ a(x) = x \quad .$$

In the model this is represented by the element:

$$\langle 2, \langle 1, \lambda x. \langle 2, \langle 0, a(x), x \rangle\rangle\rangle\rangle \; .$$

A little odd looking, but it is a perfectly good λ-term. (As we know, this is a *true* formula; but we do not get the truth definition until the next section.)

The point to keep in mind here is that we are doing *syntax* (except for the case of $a \in b$, but more of that later). At the moment we are concerned with *form* and not with meaning. What we have done is to make it possible to assign to every logical formula (containing possibly *constants* for elements of the model) a λ-term which, of course, denotes an element of the model. This element represents the formula itself, not its interpretation. And, as long as we do not use the $\in$-notation, the representation of formulas by elements is *unique*, because the whole system is based on tuples. (If we wanted full uniqueness, we could make the $\in$-combination a primitive - say $\langle 5,a,b \rangle$ - and save the $b(a)$-part for the truth definition.) We have not tried to respect the distinction between use and mention by going to the metalanguage and then defining a *mapping* from formulas to objects. Instead we have shown the effect of the mapping by regarding the logical connectives as combinators. But note - and this is very important - the approach is intensional. Our definition contributes almost nothing toward meaning.

A smaller point we should keep in mind is that we are not saying *which* elements represent formulas. All the definition does is to give laws of composition whereby new formulas can be obtained from old. Since $a = b$ is at once a formula (better: an element that *represents* a formula), we have something to start with. We just have no need to say how far we want to *iterate* formula construction. Whatever they are, they are elements of the model; and we shall find we do not need to be more definite than that.

Part of the trick of the definition (aside from the obvious use of tuples to give us an "abstract" syntax) is the use of λ-abstraction in the quantifier. It would not be unreasonable to say that the formulas represented in the model are those _without_ free variables. A formula _with_ free variables could be taken as the corresponding λ-abstract. Or if we like: a formula with a free variable is a _mapping_ from constants to the corresponding substitution instance. In this way we eliminate all fuss with variables in the formalization - the use of λ does all the work behind the scenes. This may not be quite the idea that Curry had in mind in the begining of the subject, but it does not seem like such a bad idea. (Thus, if we choose to regard $\langle 1,u \rangle$ as a quantified formula, then for any constant a, the element $u(a)$ is the corresponding "substitution" instance.) At least we can say we are putting λ-calculus to work.

§ 2. Semantics. We can now pass on to the truth definition, but we shall find it is a _transfinite_ one. This is something that Fitch has never made especially clear in his writings, but the method is actually very well known in logic. (For references see the recent book by Moschovakis.) We need only take a little care to formulate the definition in such a way that we will know - on the grounds of general principles - that the truth predicate actually _exists_. To do this it is helpful to define the predicate of _falsehood_ at the same time as truth in a mutually recursive way. This will be found to be only a very slight complication in concept which makes it easier to formulate the clauses of the definition. We will see that the definition is very much like that of first-order truth in model theory, but since we made syntax _part_ of the model there is extra "feed back" that drives us to transfinite lengths. The important question will be: how can we make use of this feed back? It is not likely that we can give a very convincing answer at once.

<u>Definition</u>. The subsets $\mathfrak{T}$ and $\mathfrak{F}$ are the least subsets of the model such that the following equivalences hold for all elements u :

$$
\begin{array}{llllll}
\mathfrak{T}u & \text{iff either} & u = \langle 0,u_1,u_2\rangle & \text{and} & u_1 = u_2 & \\
 & \text{or} & u = \langle 1,u_1\rangle & \text{and} & \mathfrak{T}u_1(x) & \text{for all } x \\
 & \text{or} & u = \langle 2,u_1\rangle & \text{and} & \mathfrak{F}u_1 & \\
 & \text{or} & u = \langle 3,u_1,u_2\rangle & \text{and} & \mathfrak{T}u_1 & \text{and } \mathfrak{T}u_2 \\
 & \text{or} & u = \langle 4,u_1,u_2\rangle & \text{and} & \mathfrak{T}u_1 & \text{and } \mathfrak{T}u_2
\end{array}
$$

$$
\begin{array}{llllll}
\mathfrak{F}u & \text{iff either} & u = \langle 0,u_1,u_2\rangle & \text{and} & u_1 \neq u_2 & \\
 & \text{or} & u = \langle 1,u_1\rangle & \text{and} & \mathfrak{F}u_1(x) & \text{for some } x \\
 & \text{or} & u = \langle 2,u_1\rangle & \text{and} & \mathfrak{T}u_1 & \\
 & \text{or} & u = \langle 3,u_1,u_2\rangle & \text{and} & \mathfrak{F}u_1 & \text{or } \mathfrak{F}u_2 \\
 & \text{or} & u = \langle 4,u_1,u_2\rangle & \text{and} & \mathfrak{F}u_1 & \text{and } \mathfrak{F}u_2
\end{array}
$$

Since we went to all the trouble to formalize the syntax <u>within</u> the model, the reader will notice that we have relapsed into English in the metalanguage. We have written "$\mathfrak{T}u$" to mean the same as "u belongs to the subset $\mathfrak{T}$". We will want te read "$\mathfrak{T}u$" as "u is <u>true</u>", but first we should see why the definition is a proper one.

We remark first that negation is never applied to the predicates $\mathfrak{T}$ and $\mathfrak{F}$ in the definition. So, by transfinite recursion, let us begin with $\mathfrak{T}_0$ and $\mathfrak{F}_0$ as the <u>empty</u> subsets. At each ordinal stage α in the recursion, put $\mathfrak{T}_\alpha$ and $\mathfrak{F}_\alpha$ on the right hand sides of the above, thereby defining by these equivalences new predicates $\mathfrak{T}_{\alpha+1}$ and $\mathfrak{F}_{\alpha+1}$. At a limit stage we take unions:

$$\mathfrak{T}_\alpha = \bigcup_{\beta<\alpha} \mathfrak{T}_\beta$$

$$\mathfrak{F}_\alpha = \bigcup_{\beta<\alpha} \mathfrak{F}_\beta$$

Because negation was avoided, we can prove by transfinite induction that if $\beta < \alpha$, then $\mathfrak{T}_\beta \subseteq \mathfrak{T}_\alpha$ and $\mathfrak{F}_\beta \subseteq \mathfrak{F}_\alpha$; that is, we have two chains of subsets. But the model is a <u>set</u> (that is, of limited cardinality), thus there must exist a stage α where the definition "closes off" in the sense that:

$$\mathfrak{T}_{\alpha+1} = \mathfrak{T}_\alpha \quad \text{and} \quad \mathfrak{F}_{\alpha+1} = \mathfrak{F}_\alpha \quad .$$

These are the desired predicates $\mathfrak{T}$ and $\mathfrak{F}$, and we call the least such α the <u>ordinal</u> of the model. (Question: Which ordinals are ordinals of models of λ-calculus?).

We also note that thanks to the separation of cases (by 0,1,2,3,4) and by the careful choice of the clauses, we can prove by transfinite induction that $\mathfrak{T}_\alpha$ and $\mathfrak{F}_\alpha$ are always <u>disjoint</u>. This means that no formula (element of the model) can be <u>both</u> true and false at the same time.

We note, too, that if we restrict attention to the usual first-order formulas (that is, start with equations between λ-terms, and then use only $\forall, \sim, \wedge$), then the definition is exactly the usual truth definition - word for word, except for our use of the formalized syntax. To check these points it is helpful to separate out cases making use of a readable notation:

<u>Lemma</u>. The following equivalences hold:

$\mathfrak{T}a = b$	iff	$a = b$	$\mathfrak{F}a = b$	iff	$a \neq b$
$\mathfrak{T}a \in b$	iff	$\mathfrak{T}b(a)$	$\mathfrak{F}a \in b$	iff	$\mathfrak{F}b(a)$
$\mathfrak{T}\forall x.\varphi$	iff	$\mathfrak{T}\varphi[a/x]$ all a	$\mathfrak{F}\forall x.\varphi$	iff	$\mathfrak{F}\varphi[a/x]$ some a

$\mathfrak{T}\exists x.\varphi$	iff	$\mathfrak{T}\varphi[a/x]$			$\mathfrak{F}\exists x.\varphi$	iff	$\mathfrak{F}\varphi[a/x]$		
		some a					all a		
$\mathfrak{T}\sim\varphi$	iff	$\mathfrak{F}\varphi$			$\mathfrak{F}\sim\varphi$	iff	$\mathfrak{T}\varphi$		
$\mathfrak{T}\varphi\wedge\psi$	iff	$\mathfrak{T}\varphi$	and	$\mathfrak{T}\psi$	$\mathfrak{F}\varphi\wedge\psi$	iff	$\mathfrak{F}\varphi$	or	$\mathfrak{F}\psi$
$\mathfrak{T}\varphi\vee\psi$	iff	$\mathfrak{T}\varphi$	or	$\mathfrak{T}\psi$	$\mathfrak{F}\varphi\vee\psi$	iff	$\mathfrak{F}\varphi$	and	$\mathfrak{F}\psi$
$\mathfrak{T}\varphi \times\!\!\times \psi$	iff	$\mathfrak{T}\varphi$	and	$\mathfrak{T}\psi$	$\mathfrak{F}\varphi \times\!\!\times \psi$	iff	$\mathfrak{F}\varphi$	and	$\mathfrak{F}\psi$

We do not bother to include the other connectives, except to remark that T is true, F is false, and $*$ is <u>neither</u>. Thus some formulas <u>lack</u> truth values.

This last remark seems very arbitrary indeed. Why put in that stupid connective $\times\!\!\times$ at all? There is an answer to this, but it was not just to find truth-value gaps. These are forced on us in any case by the Russel Paradox. Consider the abstract:

$$r = \lambda x.\ \sim x \in x$$

We ask as usual for the truth value of $r \in r$. If it is true, it is false; if false, then true. But true and false are exclusive; hence, $r \in r$ has no truth value.

Is this really the Russell Paradox? (Actually there is no paradox unless you thought every likely looking formula should have a truth value.) The answer is <u>yes</u>, because for any abstract we have:

$$\mathfrak{T}a \in \lambda x.\varphi \quad \text{iff} \quad \mathfrak{T}\varphi[a/x]$$
$$\mathfrak{F}a \in \lambda x.\varphi \quad \text{iff} \quad \mathfrak{F}\varphi[a/x]\ .$$

That is to say, λ-abstraction in conjunction with the truth definition works just like <u>class</u> abstraction. All we have to do is to interpret membership by functional application. This is something we have always wanted to do "clear back" to the genesis of the subject when Schoenfinkel created the combinators. We have now done it, but at a certain loss of innocence.

The price we have had to pay is the tax of intensionality. When we write:

$$\lambda x.\varphi = \lambda x.\psi \quad ,$$

we <u>cannot</u> mean that the two classes are extensionally equivalent. The way the truth definition is set up, what we mean is, for each constant a, that $\varphi[a/x]$ is the <u>same formula</u> as $\psi[a/x]$. This is a much stricter relationship than that we thought we were promised at the base of the tree of knowledge. Is the price worth the outcome? Fitch certainly thinks so, but one must say that in his book he does not explain all that well just what rules of extensionality he really wants. Maybe he will find the choice made here <u>too strict</u>; but if so, the alternatives will have to be spelled out more simply.

Since we see now as a consequence of allowing self-application that we are stuck with truth-value gaps, we realize that the "logic" of this truth definition is <u>three valued</u>. Using the symbols to stand for the truth "values", we have these tables:

$\sim$	
T	F
*	*
F	T

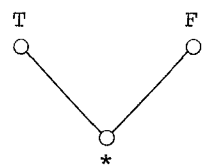
$\wedge$	T	*	F
T	T	*	F
*	*	*	F
F	F	F	F

$\vee$	T	*	F
T	T	T	T
*	T	*	*
F	T	*	F

$\bowtie$	T	*	F
T	T	*	*
*	*	*	*
F	*	*	F

We of course read "*" as "having no value". And now we can see the reason for the notation "$\bowtie$": this connective gives the common part of the <u>and</u> and the <u>or</u>. Since "$\vee$" is often called the wedge , we call "$\bowtie$" the <u>squadge</u>, since it is formed by squeezing symbols together making an ideogram. What is the point of this new connective?

The idea of $\bowtie$ may well have been thought of before, but it was discovered as necessary by Stephen Blamey of Oxford in his study of presuppositions and truth-value gaps. Without going into the details of Blamey's motivation, we can say that considering the truth values as partially ordered in this natural pattern:



Then the $\bowtie$-connective is just what you need to adjoin to $\sim, \wedge$ and $\vee$ to have a system complete for defining all monotone three-valued propositional operators (that is, monotone in the partial ordering). These are the connectives whose $\mathfrak{T}$-$\mathfrak{F}$ conditions can be defined positively in terms of $\mathfrak{T}$ and $\mathfrak{F}$. We could use no others.

(We might remark that if this method were applied to a URS, then in the clauses for $a = b$ we would only assign a truth value when both sides are defined, and in the quantifier clause the variable should range over just the defined elements. It does not seem reasonable to this author to treat the undefined element of a URS as a real element. It seems especially unnatural to let it enter into decidable equations (equations with truth values).)

A good question to ask is whether we need any other primitive quantifiers besides $\forall$, but the author does not have enough experience with the model theory of this three-valued logic to answer the question.

§ 3. Proof Theory. The sketch here will be quite brief; for the case of pure (three-valued) first-order logic, more details will be found in Blamey's Oxford thesis. It is useful to see something of what is achieved by the truth definition, however, in order to evaluate further the claims that there are connections between λ-calculus and logic.

<u>Definition</u>. The <u>consequence relation</u> $\Gamma \vdash \Delta$, which stands between two subsets of the model, holds if and only if we have these two conditions:

(i) if $\Gamma \subseteq \mathfrak{T}$, then $\Delta \cap \mathfrak{T} \neq \emptyset$; and

(ii) if $\Delta \subseteq \mathfrak{F}$, then $\Gamma \cap \mathfrak{F} \neq \emptyset$.

We have taken care to make the definition of consequence symmetric in the true and the false. In words $\Gamma \vdash \Delta$ means that whenever all formulas in Γ are true, then at least one in Δ must be true also; <u>and</u> whenever all formulas in Δ are false, then at least one in Γ is false also. In two-valued logic condition (ii) follows from (i), but in the three-valued case we must be more explicit. The definition was formulated for arbitrary subsets, but in the sequel we consider only finite Γ and Δ. In expressing various laws, we use a comma to indicate union of sets:

$$\Gamma, \Gamma' = \Gamma \cup \Gamma' \quad \text{and}$$

$$\Gamma, \varphi = \Gamma \cup \{\varphi\} .$$

We also write: $\sim \Gamma = \{\sim \varphi \mid \varphi \in \Gamma\}$

<u>The Laws of Consequence</u>. The following general principles hold for all subsets:

(R) $\Gamma \vdash \Delta$ provided $\Gamma \cap \Delta \neq \emptyset$

(M) $$\frac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

(T) $$\frac{\Gamma, \varphi \vdash \Delta \qquad \Gamma \vdash \varphi, \Delta}{\Gamma \vdash \Delta}$$

The consequence relation is a kind of multiary partial ordering, and these general laws are the reflexive, monotone, and transitive properties of $\vdash$. In logic the transitive law (T) is usually called the <u>cut rule</u>. It is a general form of <u>modus ponens</u>.

<u>The Laws of the Truth Values</u>.

$$(T * F) \qquad \vdash T \qquad F \vdash \qquad * \vdash \sim * \qquad \sim * \vdash *$$

We want T to be true, F to be false, and $*$ to have no value. We have not quite said the latter; rather the law gives $*$ the same truth value as its negation. We could write this more shortly as:

$$* \vdash\dashv \sim *$$

as an abbreviation for the two laws. After we exhibit the laws of negation we will discuss how $*$ fails to have a truth value.

<u>The Laws of Negation</u>.

$$(\sim) \quad \frac{\Gamma \vdash \Delta}{\sim\Delta \vdash \sim\Gamma} \qquad \varphi \vdash\dashv \sim\sim \varphi \qquad \varphi, \sim \varphi \vdash *$$

The first principle captures part of the symmetry between the true and the false; while the second, the law of double negation, provides the rest. In particular, the two together - with the aid of cut - give:

$$\frac{\sim \Gamma \vdash \sim \Delta}{\Delta \vdash \Gamma}$$

and other obvious variations. The last law is that non-contradiction: if φ and $\sim \varphi$ were both true, then something without a truth value would have one. But why does $*$ lack a truth value?

The answer to the question is that $*$ could indeed be given a truth value if we would want to be perverse, but there is no need to be so. In giving a completeness proof for (a suitable portion of) our logic, we would consider the situation where $\Gamma \vdash \Delta$ was not derivable. By cut, either $\Gamma, \vdash *, \Delta$ or $\Gamma, * \vdash \Delta$ would fail. In the latter case by all the various laws involving negation, we would equivalently say that $\sim \Delta \vdash *, \sim \Gamma$ fails; and this is now parallel to $\Gamma \vdash *, \Delta$.

When an entailment fails in such a formal system, we can usually argue that there is a total valuation v defined on formulas where all on the left of the failing $\vdash$ have the value T, while all on the right have the value F. In either case above we are making sure that $v(*) = F$. Now this equation does not mean that $*$ has the truth value false; rather it means that it is not true. The truth sets corresponding to v are defined by:

$$\mathfrak{T}_v = \{\varphi \mid v(\varphi) = T\}$$

$$\mathfrak{F}_v = \{\varphi \mid v(\sim\varphi) = T\}$$

Now we can see the reasoning behind the rules: we want $\mathfrak{T}_v$ and $\mathfrak{F}_v$ to have the biconditionals of a truth definition as done in the previous section. And we also see that $*$ belongs neither to $\mathfrak{T}_v$ nor to $\mathfrak{F}_v$, as was desired. (We are of course not giving the details of any completeness proof here, and these remarks correspond to just a few steps in such a proof.)

Inumerable writers have discussed three-valued logic. Feferman rejected it in his study an not practical, but the reasons do not seem just. Anderson/Belnap (partly influenced by Fitch one would guess) have mentioned it - also the four-valued case without the law of non-contradiction. But to the author's best knowledge, no paper has made any really serious contribution to a general study of the model theory. Perhaps the very rich λ-calculus models will provide some incentive.

The Laws of the Connectives.

$(\wedge)$	$\varphi, \psi \vdash \varphi \wedge \psi$	$(\vee)$	$\varphi \vee \psi \vdash \varphi, \psi$
	$\varphi \wedge \psi \vdash \varphi$		$\varphi \vdash \varphi \vee \psi$
	$\varphi \wedge \psi \vdash \psi$		$\psi \vdash \varphi \vee \psi$

$$(\times\!\!\times) \qquad \varphi,\psi \vdash *, \varphi \times\!\!\times \psi \qquad\qquad \varphi \times\!\!\times \psi, * \vdash \varphi, \psi$$

$$\varphi \times\!\!\times \psi \vdash *, \varphi \qquad\qquad \varphi, * \vdash \varphi \times\!\!\times \psi$$

$$\varphi \times\!\!\times \psi \vdash *, \psi \qquad\qquad \psi, * \vdash \varphi \times\!\!\times \psi$$

What has been done here is just to transcribe the three-valued truth tables for these connectives. Note how useful $*$ is in blocking the symmetry of the rules when necessary. Note too that the various De Morgan laws follow easily using laws of negation:

$$\sim [\varphi \wedge \psi] \dashv\vdash \sim \varphi \vee \sim \psi$$

$$\sim [\varphi \vee \psi] \dashv\vdash \sim \varphi \wedge \sim \psi$$

$$\sim [\varphi \times\!\!\times \psi] \dashv\vdash \sim \varphi \times\!\!\times \sim \psi$$

<u>The Laws of the Quantifiers</u>.

$$(\forall) \qquad \forall x.\varphi \vdash \varphi[a/x] \qquad\qquad (\exists) \qquad \varphi[a/x] \vdash \exists x.\varphi$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \forall x.\varphi, \Delta} \qquad\qquad \frac{\Delta, \varphi \vdash \Gamma}{\Delta, \exists x.\varphi \vdash \Gamma}$$

<u>provided that</u> x <u>is not free in</u> Γ .

These laws are the standard ones and they work just as well in the three-valued logic.

<u>The Laws of Equality</u>.

$$(=) \qquad \vdash a = a \qquad\qquad a = b,\ \varphi[a/x] \vdash \varphi[b/x]$$

Transitivity and symmetry of $=$ of course follow. If we use φ as $\sim x = b$, we have:

$$a = b\ ,\ \sim a = b \vdash \sim b = b \quad .$$

From the first law of equality and the laws of negation we derive:

$$\vdash a = b\ ,\ \sim a = b\ ,$$

that is, the law of the excluded middle for equality. As remarked earlier, we might not wish to assume this for all systems like those based on a URS.

Remember that formulas are elements of the model. Thus a special case of (=) above is the odd looking:

$$\varphi = \psi \, , \; \varphi \vdash \psi$$

In our type-free logic this make quite good sense, once you think it over. In his book (§ 5.5 pp. 43-44) Fitch seems to suggest the following law of "extensionality":

$$\frac{\varphi \dashv\vdash \psi}{\vdash \varphi = \psi}$$

The present author is quite unable to see why Fitch wants this. Perhaps, for some very restricted set of φ and ψ it should hold, but even this seems intuitively doubtful. Experience based on some of Feferman's arguments suggests that almost any form of extensionality is risky in such systems. Fitch of course claims a consistency proof, but his rules are so numerous (four double columns of index!) and the exceptions are so well hidden in the text, it is next to impossible to judge his claim. (In any case he does not give the proof in the book.) Certainly the very intensional view of formulas in our model makes this rule invalid (for example $* \neq \sim *$ even though $* \dashv\vdash \sim *$ holds). Possibly a further study of extensionality would be worthwhile, but a way out is to be satisfied with equivalence relations rather than identities. Equivalences and congruences are necessary sooner or later any way, and Fitch's rule of extensionality seems to have no very special merit, since we can assume all the equations we need in a more standard format. For example:

The Laws of λ-Calculus.

$$(\alpha) \qquad \vdash \lambda x.\tau = \lambda y.\tau[y/x] \qquad (y \text{ not free in } \tau)$$

$$(\beta) \qquad \vdash (\lambda x.\tau)(y) = \tau[y/x]$$

$$(\xi) \qquad \lambda x.\tau = \lambda y.\sigma \vdash\dashv \forall x.\tau = \sigma$$

The only difference with what we said at the beginning of this paper is the use of "$\vdash$" and "$\vdash\dashv$" and the view that the "$\forall x$" is now formalized. It all fits together quite neatly.

Because "ϵ" is a defined symbol, the (β) rule along with the rules of equality give us at once what might be called "Church's Rule":

$$y \in \lambda x.\varphi \vdash\dashv \varphi[y/x] \quad .$$

By using laws (and definitions!) of the connectives and the quantifiers, we can show:

$$\forall x[\varphi \vee \sim \varphi] \vdash \exists a \ \forall x[x \in a \leftrightarrow \varphi] \ ,$$

where a is not free in φ. Thus a <u>version</u> of naive set theory does come out, but it is one protected from contradiction by the three-valued logic (the basic idea of Fitch). The question remains: is this theory of classes any good? What-ever we do it will remain somewhat messy owing to the failure of extensionality.

The laws that have been stated in this section refer to a fixed but arbitrary model onto which we have grafted a fairly natural truth definition. (Truth is of course not directly definable in the original model without the external transfinite recursion. Note, however, that truth is definable <u>in terms of truth</u>:

$$\varphi \in \lambda x.x \vdash\dashv \varphi$$

That is, the combinator $\lambda x.x$ defines the class $\mathfrak{T}$ while $\lambda x.\sim x$ defines $\mathfrak{F}$.) Once formal laws have been isolated, though, they take on a life of their own - unfortunately. Even we cannot refrain from asking some formal questions: are the laws as formulated complete with respect to the intended interpretations? The author is afraid that the proposed truth definition takes on something of the character of truth in higher-order logic and is therefore not axiomatizable. But he does not see the answer off hand. Even if the rules are not complete for "standard" models, the formal system _might_ be interesting. Can there be a _cut-free_ version of this logic with a normalization theorem, or do the λ-terms without normal forms spoil things? The author simply does not know enough proof theory to answer this question.

It might be useful to look at other variations of this logic using, for instance, the lattice or cpo models of λ-calculus. For these, we should replace $=$ as a primitive by $\sqsubseteq$ and include laws of a partial ordering (even: lattice). Remember too that the paradoxical combinator Y may have special properties in such models, so one may also wish to formalize what is called "Scott's Induction Rule":

$$\frac{\Gamma,\ a(x) \sqsubseteq b(x) \vdash a(u(x)) \sqsubseteq b(u(x))}{\Gamma,\ a(\perp) \sqsubseteq b(\perp) \vdash a(Y(u)) \sqsubseteq b(Y(u))}$$

provided x is not free in Γ .

§ 4. Applications. There is no question now that we have well-grounded _connections_ between logic and λ-calculus: the truth definition can be faulted for its infinitistic character, but it cannot be ignored. We simply leave aside in this paper the question of whether using these ideas the tables can be turned so that we could argue that we have a _foundation_ for logic. Until we have the applications more firmly in hand, this question is unimportant. We understand set theory and model theory and truth definitions well enough to be able to investigate λ-calculus from this point of view more thoroughly without going all mystical or becoming militant type-libers.

One application that has in effect already been proposed is the investigation of <u>particular</u> models by trying to formalize more fully the rules that are valid for them. This activity is interesting only in-so-far as the models themselves are interesting. But there may be some point in connecting the models with the logic. Mc Carthy has suggested many times adjoining quantifiers to the LCF system of Milner. There is a serious question of whether a quantifier calculus is ever practical for any kind of automatic theorem proving or even theorem checking program; but if we put this question aside, the proposal of this paper does just what is asked for. Indeed the feed back generated by the syntax formalization goes far beyond what was expected. If we can show that the expressive power of the proposed system is of independent interest, then a further look at automated rules may be called for. Some preliminary hints about the expressive power will now be given.

Go back and look at the form of the truth definition in Section 2. We have a monotone, first-order, inductive definition. We are using (in the metalanguage) (1) equations and the given algebraic structure of the λ-calculus model; (2) positive connectives like <u>and</u> and <u>or</u>; (3) quantifiers over the model. (The fact that we had to define <u>two</u> predicates $\mathfrak{T}$ and $\mathfrak{F}$ is irrelevant.) Consider any other such definition defining, say, a one-place predicate $\mathfrak{P}$. The definition is done in the metalanguage remember, but we can formalize it by replacing $\mathfrak{P}\,x$ everywhere by $x \in p$ and by using the formal connectives and quantifiers. Having done so, we find corresponding to the right-hand side of the definition a formula φ with p and x as the only free variables. By the <u>fixed point</u> theorem of the λ-calculus, we can find a specific <u>element</u> p of the model such that the equation

$$p = \lambda x.\, \varphi$$

is true. By itself this is not the definition of $\mathfrak{P}$, but only a "formal" solution. We get the required solution by invoking the truth definition, for we can prove by transfinite induction that

$$\mathfrak{P}x \quad \text{iff} \quad \mathfrak{T}x \in p$$

for all x in the model. Thus the element p does in fact represent the predicate (or class) $\mathfrak{P}$. Another way of stating this result is to say that the $\mathfrak{T}$-predicate is universal for all such recursively defined predicates, since a direct formalization of their definitions provides a <u>reduction</u>. This makes it clear that the classes represented by the combinators are very, very complicated (and that the ordinal of the model is quite large).

As usual with recursive definitions, the law of the excluded middle fades. It is interesting to ask about those classes for which it holds. We define:

$$V = \lambda a.\ \forall x\, [x \in a \vee \sim x \in a]\ .$$

Note that the formula $a \in V$ can never be <u>false</u>; it often fails to be true, but that is something else. Those a where $a \in V$ is true are called <u>definite</u>: for each x the truth definition has decided membership one way or the other. There are many definite a because we made $=$ a definite relation. Thus

$$a = (\lambda x.\ u(x) = v(x))$$

is definite for any choice of u and v, and the author has shown that this can be a very extensive family of classes. Quite generally, however, we can show that V enjoys pleasant closure properties.

Let us define some new "logical" combinators:

$$\mathrm{F}ab = \lambda f.\ \forall x[\sim x \in a \vee f(x) \in b]$$

$$\Pi ab = \lambda f.\ \forall x[\sim x \in a \vee f(x) \in b(x)]$$

$$\Sigma ab = \lambda u.\ \exists x\ \exists y[x \in a \wedge y \in b(x) \wedge u = \langle x,y\rangle]$$

We want to relate F to Curry's ideas on <u>functionality</u> and Π and Σ to de Bruijn's and Martin-Löf's ideas on <u>types</u>.

As regards F, we can easily establish the validity of these rules:

$$f \in \mathrm{F}ab\ ,\ x \in a \vdash *\ ,\ f(x) \in b$$

$$\frac{\Gamma,\ x \in a \vdash *,\ f(x) \in b}{\Gamma,\ a \in V \vdash *,\ f \in \mathrm{F}ab}$$

provided x is not free in Γ. In other words, <u>when</u> a <u>is definite</u>, we can read $f \in \mathrm{F}ab$ as meaning that f maps a into b. This seems to formalize in our system what Curry had in mind about functionality, but he was never very definite about what it means to be <u>definite</u>.

A very important point about F and V is the closure condition:

$$a \in V\ ,\ b \in V \vdash \mathrm{F}ab \in V\quad .$$

This is already indication enough that V is a big universe since so many different definite classes can be represented by F-expressions. We should note too the functionality laws of the favourite combinators:

$$a \in V \vdash *\ ,\ I \in \mathrm{F}aa$$

$$a \in V\ ,\ b \in V \vdash *\ ,\ K \in \mathrm{F}a\,\mathrm{F}ba$$

$$a \in V\ ,\ b \in V\ ,\ c \in V \vdash *\ ,\ S \in \mathrm{FF}a\mathrm{F}bc\mathrm{FF}ab\mathrm{F}ac$$

In the case of I we can even write:

$$a \in V \vdash\dashv I \in \mathrm{F}aa\ ,$$

but this does not seem too important.

The combinator F gives a function-space construction, while Π makes a cartesian product. The notation $\Pi a\ \lambda x.b(x)$ would correspond to the notation $[x,a]b(x)$ of de Bruijn (see the paper by de Vrijer from this symposium). We have the rules:

$$f \in \Pi ab \ , \ x \in a \vdash * \ , \ f(x) \in b(x)$$

$$\frac{\Gamma,\ x \in a \vdash * \ , \ f(x) \in b(x)}{\Gamma,\ a \in V \vdash * \ , \ f \in \Pi ab}$$

provided x is not free in Γ. The relevant closure condition reads:

$$a \in V \ , \ b \in FaV \vdash \Pi ab \in V \ .$$

This is a principle of "type inclusion" (see § 1.1.3 of de Vrijer's paper), but we are not at all identifying b with Πab which would be very confusing. The rules for the Σ-combinator can be left to the reader.

We are thinking of V as the type of all types. Of course V is not really a type because $V \in V$ is not true; however, it is a semi-type, and what we wrote about Π and V above is quite coherent. Thus V represents a very interesting class (which gives some kind of a model for some kind of type theory) and we would like to do some recursive definitions on it. We cannot go very far, however, because V is not definite. Well, why not make it definite? We can, but this involves a new recursive truth definition. (The author learned this idea from Aczel at the Kiel Logic Summer School.) In fact we shall make a whole hierarchy of truth definitions in a way that would seem to incorporate an old idea of Fitch.

We add to our syntactical primitives a new equation:

$$\varphi^{(i)} = \langle\ 5+i, \varphi\ \rangle$$

Instead of one pair $\mathfrak{T}, \mathfrak{F}$ we have an infinite number $\mathfrak{T}^{(n)}, \mathfrak{F}^{(n)}$ where the recursion is such that the $n^{\underline{th}}$ pair is defined in terms of the pairs for $i < n$. Indeed we change the clauses of the truth definition to be able to adjoin these equivalences:

$$\mathfrak{T}^{(n)}\varphi^{(i)} \quad \text{iff} \quad i < n \quad \text{and} \quad \mathfrak{T}^{(i)}\varphi$$
$$\mathfrak{F}^{(n)}\varphi^{(i)} \quad \text{iff} \quad i < n \quad \text{and } \underline{\text{not}} \quad \mathfrak{T}^{(i)}\varphi \ .$$

We see at once that this is not a monotone definition because negation is being used. However, the pair $\mathfrak{T}^{(0)}, \mathfrak{F}^{(0)}$ is just our original $\mathfrak{T}, \mathfrak{F}$; so we know they exist. When we pass then to $\mathfrak{T}^{(1)}, \mathfrak{F}^{(1)}$ we are taking $\mathfrak{T}^{(0)}$ as given (that is to say, definite). As far as the recursion goes we are not defining the infinite number of predicates together, but one after the other. (The ordinals will get really large now.) From the infinitistic, set-theoretical point of view there is nothing wrong in this, and there is no reason why we cannot iterate it, as suggested.

Unless the author has made an oversight, we have the consistency conditions:

$$\mathfrak{T}^{(n)} \subseteq \mathfrak{T}^{(n+1)} \quad \text{and} \quad \mathfrak{F}^{(n)} \subseteq \mathfrak{F}^{(n+1)} \ ,$$

since each successive truth definition changes nothing of the previous ones but just makes more formulas "meaningful". Thus we can take the unions, obtaining $\mathfrak{T}^{(\infty)}, \mathfrak{F}^{(\infty)}$. (We could also iterate this passage, but enough is enough.) Using $\mathfrak{T}^{(\infty)}$ and $\mathfrak{F}^{(\infty)}$ for our logic, we can now employ a whole range of universes (somewhat in the way proposed by Martin-Löf), which are defined by:

$$V^{(i)} = \lambda a.\ \forall x[x \in a \vee \sim x \in a]^{(i)}$$

Now $V^{(0)}$ represents our old V; but since we changed the truth sets, the symbol V has a new meaning (that is, it represents a different class). In fact,

$$\vdash V^{(n)} \in V^{(n+1)}$$

and

$$\vdash V^{(n)} \in V \ .$$

The various $V^{(n)}$ will have very extensive closure properties.

Fitch once suggested a hierarchy of stronger and stronger negations.

The author is only guessing, but why not look at formulas $(\sim \varphi)^{(i)}$?
We would have

$$(\sim \varphi)^{(i)} \vdash_* , (\sim \varphi)^{(i+1)}$$

and

$$\vdash (\sim \varphi)^{(i)} \quad \text{iff} \quad \mathfrak{F}^{(i)}\varphi .$$

In other words, if we are right, Fitch introduced the series of $\mathfrak{F}$-classes. But since $(\sim \sim \varphi)^{(i)}$ is just as good as $\varphi^{(i)}$, the two ideas come to the same thing. Clearly this plan of making certain classes definite could be carried out in many different ways.

This would seem to be enough to show that there are applications of the truth definition to connect with other ideas. So the author now leaves it to someone else to make the next move.

Acknowledgement. Thanks are due to Oxford University for giving the author leave for one term and to the ETH, Zürich, whose very generous hospitality provided the excellent working conditions under which this paper could be written and very efficiently typed.