A BRIEF INTRODUCTION TO TYPED PREDICATE LOGIC*

Raymond Turner

December 6, 2010

Abstract

Typed Predicate Logic[1] was developed to offer a unifying framework for many of the current logical systems, and especially those that employ some notion of type or sort.

1 Background

Typed predicate logic [15] was developed to offer a unifying framework for many of the current logical systems, and especially those that employ some notion of type or sort. These include the simple and ramified type theories of Principia Mathematica [19], the intensional simple type theory of Montague [8], the intensional ramified type theories Thomason [10] and Turner [12] , the *Frege Structues* of Aczel [1], the logics of properties and truth of Turner [14] and [13], the constructive type theories of Martin-Löf [7], [16], the theories of operations and types of Feferman [6] and Turner [17], [18], and the Calculi of Constructions of Huet and Coquand [4].

In these notes we shall only sketch the logic itself. There is little technical development and no examples of how the above type theories find their representation in the logic[2]. We shall say more about this in the seminar.

2 Judgments, Contexts and Sequents

Typed predicate logic is framed in a sequent style version of natural deduction, however, unlike standard logical systems, where there is only one judgment form in conclusions, we admit several. More exactly, it is a many-sorted natural

*These notes have been written to provide some technical background for the seminar.

[2] These will be sketched in the presentation.

deduction system with the following four judgment forms:

$$T\ type$$
$$\phi\ prop$$
$$t : T$$
$$\phi$$

The first asserts that T is a *type.* In first order logic there is no need for such a judgment since there is only one type - a universal one where the variables range over everything. But we admit more than one. This has implications for the whole logical setup. In particular, the structure of propositions depends upon the type structure. This generates the need for the second judgement i.e., that something is a *proposition.* The third concerns type membership and the last that something (presumably a proposition) is true. We shall refer to the first three as type-inference judgments. Notice that in first order logic the last is the only judgement and the latter three are replaced with context free syntax rules. The same is true in simple type theory where the type structure is hard wired into the syntax.

These judgements are constructed from a syntax of terms that are built from variables (x_0, x_1, x_2, x_3...), constant, function, and relation symbols, including equality ($=$), and the logical connectives ($\Omega, \wedge, \vee, \neg, \rightarrow, \forall, \exists$). As metavariables for strings on these alphabets, we employ the Roman and Greek alphabets, where we reserve x, y, z, u, v, w to range over the object-level variables of the language. While this is the stuff of the syntax, the actual grammar is determined not by a traditional context-free syntax, but by a type-inference system that is constituted by the *membership* and *formation* rules for types and propositions ([11], [7], [10]). The rules for this rule-based grammar will form part of the overall proof system.[3].

Generally, judgments in the logic are made relative to a *context* Γ that is a finite sequence of terms. In the logic, these take one of the following two forms:

$$x : T$$
$$\phi$$

i.e., a *declaration* that a variable has a given type or the assumption that a proposition, ϕ, is true. The second is traditional while the first owes its existence

[3] This background syntax may be further refined via the following BNF grammar.

$$t ::= F(t_1, . - ., t_n) | R(t_1, . - ., t_n) | O(t_1, . - ., t_n) | t =_t t$$
$$t \vee t | t \wedge t | \neg t | \forall x : t \cdot t | \exists x : t . t$$

Similarly, the raw syntax of contexts might also be made explicit as follows.

$$\Gamma ::= t \mid t, \Gamma$$

However, such BNF style definitions do not play too much of a role, since they sanction way too much nonsense. They only provide the background strings for the actual grammar, which is rule-given, i.e., by the rules of the logic itself.

to the fact that the grammatical correctness of expressions that contain variables depends upon their type, which is supplied by the context.[4] Thus, sequents in the theory have the shape,

$$\Gamma \vdash \Theta$$

where Θ is one of our four judgment forms and Γ a context. Such sequents are the basic carriers of meaning in the logic. They determine not only what follows from what, but also what is grammatically legitimate. Rules in the system will take the form

$$\frac{\Gamma_1 \vdash \Theta_1, ..., \Gamma_m \vdash \Theta_m}{\Gamma \vdash \Theta}$$

and axioms will be expressed as single stand alone sequent. All such axioms and rules will be subject to a coherence constraint that ensues all the expressions are well-typed. We shall see this emerge as we proceed.

3 Relations, Functions and Types

The language of first order logic has relation and function symbols built in. A relation is introduced indicating the number of arguments, usually they are enumerated in the following way

$$R_1^n, R_2^n, R_3^n, R_4^n, ...$$

In standard type theory we also need to indicate the type of its arguments e.g.

$$R_1^{(T_1,.....,T_n)}, R_2^{(T_1,.....,T_n)}, R_3^{(T_1,.....,T_n)}, R_4^{(T_1,.....,T_n)}, ...$$

Likewise terms come decorated with their types $t_1^{T_1}, t_2^{T_2}, t_3^{T_3}, t_4^{T_4}, ...$But this is a *Church* approach to type systems where the types are hard-wired to the terms of the language. We are aiming at a more flexible *Curry* style approach. Consequently, relation symbols are introduced via sequents of the form

$$x_1 : T_1, ..., x_n : T_n \ \vdash R_{T_1,.....,T_n}(x_1, ..., x_n) \ prop$$

This introduces a new typed n-place function symbol $R_{T_1,.....,T_n}$ parameterised by the type symbols $T_1, ..., T_n$. It is typed in that it requires arguments of the right type given by the type symbols $T_1, ..., T_n$. The context $x_1 : T_1, ..., x_n : T_n$ must itself be well typed in the sense that

$$x_1 : T_1, ..., x_i : T_i \vdash T_{i+1} \ type, \ 0 \leq i < n$$

where the case $i = 0$ is interpreted as the judgment T_1 *type*.As we shall see, this guarantees that the whole sequent

$$x_1 : T_1, ..., x_n : T_n \ \vdash R_{T_1,.....,T_n}(x_1, ..., x_n) \ prop$$

[4] We shall call contexts that contain only type assignments, i.e., ones of the form $x : T$, *declaration contexts*. We shall use c, c', d, d', etc., as variables for these contexts and c_Γ for that part of the context Γ that consists of just its type assignments.

is *well-typed*. Indeed, the intention is best stated as a rule:

$$\frac{\vdash T_1\ type \qquad x_1 : T_1, ..., x_i : T_i \vdash T_{i+1}\ type \qquad 1 \leq i < n}{x_1 : T_1, ..., x_n : T_n\ \vdash R_{T_1,.....,T_n}(x_1, ..., x_n)\ prop}$$

Example 1 *We might introduce and equality relation for Numbers as follows.*

$$x : N, y : N \vdash x =_N y\ prop$$

Similarly, such a context maybe used to introduce function symbols

$$x_1 : T_1, ..., x_n : T_n\ \vdash F_{T_1,.....,T_n}(x_1, ..., x_n) : T_{n+1}[x_1, ..., x_n]$$

where the rule takes the shape

$$\frac{\vdash T_1\ type \qquad x_1 : T_1, ..., x_i : T_i \vdash T_{i+1}\ type \qquad 1 \leq i < n+1}{x_1 : T_1, ..., x_n : T_n\ \vdash F_{T_1,.....,T_n}(x_1, ..., x_n) : T_{n+1}[x_1, ..., x_n]}$$

This introduces a function symbol with results of type $T_{n+1}[x_1, ..., x_n]$.

Example 2 *We might introduce the union relation on sets of numbers as follows.*

$$u : Set(N), v : Set(N) \vdash x \cup_N y : Set(N)$$

These parallel the ways in which relations and functions are introduced in first order logic but take the types into account. In **TPL** we also require some basic rules for type symbols. These are introduced in a similar fashion to relation and function symbols.

$$x_1 : T_1, ..., x_n : T_n\ \vdash O_{T_1,.....,T_n}(x_1, ..., x_n)\ type$$

This is governed by the rule

$$\frac{\vdash T_1\ type \qquad x_1 : T_1, ..., x_i : T_i \vdash T_{i+1}\ type \qquad 1 \leq i < n}{x_1 : T_1, ..., x_n : T_n\ \vdash O_{T_1,.....,T_n}(x_1, ..., x_n)\ type}$$

In general, type terms may contain variables and their legitimacy as a type depends upon over the types of these variables.

Example 3 *The equality type of the constructive type theories of Martin-Löf furnishes a simple example.*

$$x : N, y : N \vdash Eq[N, x, y]\ type$$

4 Structural Rules

We begin with the structural rules, i.e., *assumption*, *thinning,* and *substitution*. The first two permit the addition of new (grammatically acceptable) assumptions. There is one for each kind of assumption i.e., one for attachment of types

to variables and one for propositions. Notice that grammatical constraints play a significant role. For example, in $\mathbf{A_1}$ we are only permitted to add type assignments involving terms that are types, whereas $\mathbf{A_2}$ only sanctions assumptions that are propositions. $\mathbf{W}_1$ and $\mathbf{W}_2$ allow weakening under the same grammatical constraints i.e., on the assumption that an expression is grammatically a type or proposition, it is permitted to occur as an assumption in a context. The final rule is a substitution rule. Note that it respects the fact that, in contexts, the order of the occurrence of assumptions is significant i.e., contexts are lists of assumptions not sets. This is a consequence of the fact that, in general, contexts will be dependent, i.e., the grammatical status of later assumptions may depend upon earlier ones. For example, the status of a purported proposition may depend upon the types of its free variables, and so may depend upon previous type declarations. A simple illustration of such dependence is generated by the equality rules. We shall see this shortly.

$$\mathbf{A_1} \qquad \frac{\Gamma \vdash T\ type}{\Gamma, x : T \vdash x : T} \qquad\qquad \mathbf{A_2} \qquad \frac{\Gamma \vdash \phi\ prop}{\Gamma, \phi \vdash \phi}$$

$$\mathbf{W}_1 \qquad \frac{\Gamma, \Delta \vdash \Theta \qquad \Gamma \vdash T\ type}{\Gamma, x : T, \Delta \vdash \Theta}$$

$$\mathbf{W}_2 \qquad \frac{\Gamma, \Delta \vdash \Theta \qquad \Gamma \vdash \phi\ prop}{\Gamma, \phi, \Delta \vdash \Theta}$$

$$\mathbf{Sub} \qquad \frac{\Gamma, y : S, \Delta \vdash \Theta \qquad \Gamma \vdash s : S}{\Gamma, \Delta[s/y] \vdash \Theta[s/y]}$$

In $\mathbf{A_1}$ and $\mathbf{W_1}$, x is fresh (i.e., it is not declared in Γ, Δ) and $\Theta[s/y]$ denotes Θ with the term s substituted for the variable y, with the usual demands to avoid accidently clashes of bound variables. We shall indicate the variable binders as we proceed.

Consistent with the assumption that contexts are lists, we do not have an exchange rule. The **Sub** rule could be avoided (it is partly covered by the universal elimination rule), but it will often prove convenient for the statement of our theories.

5 Equality

The formation rule for equality is a special case of the formation rule for relations. $\mathbf{E_1}$ insists that equality forms a proposition when the terms flanking it have the same type. In addition, distinguished symbols such as equality are given content by their associated axioms and rules. $\mathbf{E_2}$ and $\mathbf{E_3}$ are the standard rules of introduction and elimination; i.e., every element of every type is equal

to itself and equal objects can be substituted for each other in all contexts.

$$\mathbf{E_1}\quad \frac{\Gamma \vdash T\ type}{\Gamma, x_1 : T, y : T \vdash x =_T y\ prop} \qquad \mathbf{E_2}\quad \frac{\Gamma \vdash t : T}{\Gamma \vdash t =_T t}$$

$$\mathbf{E_3}\quad \frac{\Gamma \vdash t =_T s \qquad \Gamma \vdash \Theta[t/x]}{\Gamma \vdash \Theta[s/x]}$$

The following rule is directly derivable from the rule of substitution: given the assumptions this follows by two applications of **Sub**.

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash s : T}{\Gamma \vdash t =_T s\ prop}$$

Conversely, $\mathbf{E_1}$ is derivable from this using $\mathbf{A_1}$.

The equality rules illustrate how dependency in contexts can occur. For instance, a provable sequent such as

$$x : T, y : T, z : T, x =_T y, y =_T z \vdash x =_T z$$

demonstrates how the occurrences of equality in the context (as well as the conclusion) are legitimate (i.e., form propositions) only where their constituent terms have the same type. Observe that, as a suffix, the type of the equality symbol is explicitly marked. However, where the context determines matters, we shall often drop the subscript on the equality relation; i.e., we shall just write:

$$x : T, y : T, z : T, x = y, y = z \vdash x = z$$

This principle of parsimony will be adopted generally.

Types, relations and propositions and are not part of the object level notions. We shall later consider theories where they all are.

6 Logical Rules

We next provide the rules for the propositional connectives. The formation rules for these connectives capture their standard closure conditions, i.e., the ones normally given in a context-free style, while the introduction and elimination

rules are their standard introduction and elimination logical rules.

$$\mathbf{L_1}\ \frac{\Gamma \vdash \phi\ prop \qquad \Gamma \vdash \psi\ prop}{\Gamma \vdash \phi \wedge \psi\ prop} \qquad \mathbf{L_2}\ \frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}$$

$$\mathbf{L_3}\ \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \qquad \mathbf{L_4}\ \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}$$

$$\mathbf{L_5}\ \frac{\Gamma \vdash \phi\ prop \qquad \Gamma \vdash \psi\ prop}{\Gamma \vdash \phi \vee \psi\ prop}$$

$$\mathbf{L_6}\ \frac{\Gamma \vdash \phi \vee \psi \qquad \Gamma, \phi \vdash \eta \qquad \Gamma, \psi \vdash \eta \qquad \Gamma \vdash \eta\ prop}{\Gamma \vdash \eta}$$

$$\mathbf{L_7}\ \frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi\ prop}{\Gamma \vdash \phi \vee \psi} \qquad \mathbf{L_8}\ \frac{\Gamma \vdash \psi \qquad \Gamma \vdash \phi\ prop}{\Gamma \vdash \phi \vee \psi}$$

$$\mathbf{L_9}\ \ \Gamma \vdash \Omega\ prop \qquad \mathbf{L_{10}}\ \frac{\Gamma \vdash \phi \qquad \mathbf{\Gamma} \vdash \neg\phi}{\Gamma \vdash \Omega}$$

$$\mathbf{L_{11}}\ \frac{\Gamma \vdash \phi\ prop \qquad \Gamma \vdash \Omega}{\Gamma \vdash \phi}$$

$$\mathbf{L_{12}}\ \frac{\Gamma, \phi \vdash \psi\ prop}{\Gamma \vdash \phi \rightarrow \psi\ prop} \qquad \mathbf{L_{13}}\ \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \rightarrow \psi}$$

$$\mathbf{L_{14}}\ \frac{\Gamma \vdash \phi \rightarrow \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}$$

$$\mathbf{L_{15}}\ \frac{\Gamma \vdash \phi\ prop}{\Gamma \vdash \neg\phi\ prop} \qquad \mathbf{L_{16}}\ \frac{\Gamma, \phi \vdash \Omega}{\Gamma \vdash \neg\phi} \qquad \mathbf{L_{17}}\ \frac{\Gamma, \neg\phi \vdash \Omega}{\Gamma \vdash \phi}$$

There are additional grammatical assumptions in some of the rules. For instance, in the disjunction introduction rules ($\mathbf{L_7}, \mathbf{L_8}$), e.g.

$$\frac{\Gamma \vdash \psi \qquad \Gamma \vdash \phi\ prop}{\Gamma \vdash \phi \vee \psi}$$

we include the assumption that the alternate constituent of the disjunction has to be a proposition. These grammatical side conditions, as we shall see, are to ensure that only grammatically legitimate objects (i.e., propositions) are provable. One might think that the rule should be

$$\frac{\Gamma \vdash \psi \qquad \Gamma \vdash \phi\ prop \qquad \Gamma \vdash \psi\ prop}{\Gamma \vdash \phi \vee \psi}$$

Actually for completeness reasons it ought to be. However, as we shall see, for coherence, the original is enough. This observation applies generally i.e., we have adopted a minimalist approach to the inclusion of side conditions in that we have included just enough to ensure that coherence is guaranteed We shall discuss this shortly. Unless overridden by parentheses, we shall assume that negation takes precedence over conjunction and disjunction, which take precedence over implication. However, most of the time we shall use brackets.

7 Quantifier Rules

Aside from their generalized grammatical setting, the rules for the quantifiers are also classical. In particular, we assume the normal side conditions for the quantifier rules. More specifically, in $\mathbf{L_{20}}$, x must not be free in in any proposition in Γ, and it must not be free in T or η.In $\mathbf{L_{22}}$, x must not be free in any proposition in Γ.

$$\mathbf{L_{18}}\ \frac{\Gamma, x : T \vdash \phi\ prop}{\Gamma \vdash \exists x : T \cdot \phi\ prop}$$

$$\mathbf{L_{19}}\ \frac{\Gamma \vdash \phi[t/x] \qquad \Gamma \vdash t : T \qquad \Gamma, x : T \vdash \phi\ prop}{\Gamma \vdash \exists x : T \cdot \phi}$$

$$\mathbf{L_{20}}\ \frac{\Gamma \vdash \exists x : T \cdot \phi \qquad \Gamma, x : T, \phi \vdash \eta \qquad \Gamma \vdash \eta\ prop}{\Gamma \vdash \eta}$$

$$\mathbf{L_{21}}\ \frac{\Gamma, x : T \vdash \phi\ prop}{\Gamma \vdash \forall x : T \cdot \phi\ prop}$$

$$\mathbf{L_{22}}\ \frac{\Gamma, x : T \vdash \phi}{\Gamma \vdash \forall x : T \cdot \phi} \qquad \mathbf{L}_{23}\ \frac{\Gamma \vdash \forall x : T \cdot \phi \qquad \Gamma \vdash t : T}{\Gamma \vdash \phi[t/x]}$$

We shall assume that the scope of the quantifier in $\forall x : T \cdot \phi$, $\exists x : T \cdot \phi$ is the whole of ϕ. It is only overridden by explicit parentheses.

This concludes the rules of **TPL**. The underlying logic is classical logic. The Intuitionistic system is obtained by dropping $\mathbf{L_{17}}$.We shall often indicate matters explicitly and write

$$\Gamma \vdash_{\mathbf{TPL}} \Theta$$

if the sequent $\Gamma \vdash \Theta$ is derivable using the rules of **TPL**.

8 TPL Derivations

There is little here that is not a straightforward generalization that flows from the additional rules that replace the standard context-free grammar of a typed logic. However, given the novel nature of **TPL**, we illustrate its notion of deduction with some simple examples. Of course, we shall see many more throughout the book. However, they will be somewhat less completely and formally presented.

Example 4 *We deduce*

$$\forall x : B \cdot \forall y : B \cdot x =_B y \rightarrow y =_B x$$

By the first equality rule, $\mathbf{E}_1$*, we have*

$$\frac{x : B, y : B \vdash x : B \qquad x : B, y : B \vdash y : B}{x : B, y : B \vdash x =_B y\ prop} \tag{1}$$

In the following, (2) is an instance of the structural rule, $\mathbf{A_2}$.

$$\frac{x : B, y : B \vdash x =_B y \ prop}{x : B, y : B, x =_B y \vdash x =_B y} \tag{2}$$

Step (3) is an instance of the second equality rule $\mathbf{E}_2$.

$$\frac{x : B \vdash x : B}{x : B \vdash x =_B x} \tag{3}$$

The conclusion of (3) may be enriched to (4). This follows by a judicious use of $\mathbf{A}_1$ *and* $\mathbf{A_2}$.

$$\frac{x : B \vdash x =_B x}{x : B, y : B, x =_B y \vdash x =_B x} \tag{4}$$

By the third equality rule, the conclusions of (2) and (4), we may deduce the following

$$\frac{x : B, y : B, x =_B y \vdash x =_B y \qquad x : B \vdash x =_B x}{x : B, y : B, x =_B y \vdash y =_B x} \tag{5}$$

By the implication introduction rule $\mathbf{L}_{13}$ *and the conclusion of (5), we can deduce (6).*

$$\frac{x : B, y : B, x =_B y \vdash y =_B x}{x : B, y : B \vdash x =_B y \to y =_B x} \tag{6}$$

By $\mathbf{L}_{21}$ *and the conclusion of (6), we may conclude*

$$\frac{x : B, y : B \vdash x =_B y \to y =_B x}{x : B \vdash \forall y : B \cdot x =_B y \to y =_B x} \tag{7}$$

By the conclusion of (7) and $\mathbf{L}_{21}$, *we arrive at the following*

$$\frac{x : B \vdash \forall y : B \cdot x =_B y \to y =_B x}{\forall x : B \cdot \forall y : B \cdot x =_B y \to y =_B x} \tag{8}$$

Example 5 *We deduce*

$$\forall x : B \cdot \exists y : B \cdot x =_B y$$

By the first structural rule, $\mathbf{A}_1$, *we have*

$$\frac{B \ type}{x : B \vdash x : B} \tag{1}$$

By the first equality rule, $\mathbf{E}_1$, *we have*

$$\frac{x : B, y : B \vdash x : B \qquad x : B, y : B \vdash y : B}{x : B, y : B \vdash x =_B y \ prop} \tag{2}$$

The conclusion (3) is an instance of the equality rule $\mathbf{E_2}$

$$\frac{x : B \vdash x : B}{x : B \vdash x =_B x} \tag{3}$$

By the existential introduction rule, $\mathbf{L}_{19}$*, and conclusions of (1), (2), and (3), we may deduce the following*

$$\frac{x : B \vdash x =_B x \qquad x : B \vdash x : B \qquad x : B, y : B \vdash x =_B y \ prop}{x : B \vdash \exists y : B \cdot x =_B y} \tag{4}$$

By the conclusion of (4) and universal introduction, we obtain

$$\frac{x : B \vdash \exists y : B \cdot x =_B y}{\forall x : B \cdot \exists y : B \cdot x =_B y} \tag{5}$$

Our final example shows how the proofs may proceed with a minimal amount of grammatical checking.

Example 6 *Assume that*

$$x : A, y : B \vdash \phi[x, y] \ prop \tag{0}$$

We deduce

$$(\exists y : B \cdot \forall x : A \cdot \phi) \rightarrow (\forall x : A \cdot \exists y : B \cdot \phi) \tag{1}$$

Assume

$$\exists y : B \cdot \forall x : A \cdot \phi[x, y] \qquad u : A \qquad y : B \qquad \forall x : A \cdot \phi[x, y] \tag{2}$$

These are all grammatical by (0).By universal elimination,

$$\phi[u, y] \tag{3}$$

Given (0), we can apply existential elimination and by existential elimination,

$$\exists y : B \cdot \phi[u, y] \tag{4}$$

By universal elimination,

$$\forall x : A \cdot \exists y : B \cdot \phi \tag{5}$$

We have ignored some of the grammatical checking but it is all straightforward. ■

We shall not see a great many examples worked out in such great detail. But these should be enough for the reader to grasp the dynamics of deduction in the system.

9 Other Connectives

We now introduce other connectives in pretty much the standard way. For example, we define logical equivalence as follows.

Example 7 *Assume that*

$$\Gamma \vdash \phi \ prop \ and \ \Gamma \vdash \psi \ prop$$

We introduce new logical connectives via

$$\Gamma \vdash \phi \leftrightarrow \psi \triangleq (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$$

These is a new defined connective that illustrates the way that new notions are introduced. We shall discuss this in more detail later.

10 Coherence

TPL has been set up with to ensure that only propositions are provable and only grammatically correct expression occur in sequents. To establish this we require a lemma.

Lemma 8 *In **TPL** we have:*

1. $\Gamma \vdash \phi \circ \psi \ prop$ iff $\Gamma \vdash \phi \ prop$ and $\Gamma \vdash \psi \ prop$, where $\circ \ = \vee, \wedge$
2. $\Gamma \vdash \phi \rightarrow \psi \ prop$ iff $\Gamma, \phi \vdash \psi \ prop$,
3. $\Gamma \vdash \neg\phi \ prop$ iff $\Gamma \vdash \phi \ prop$,
4. $\Gamma \vdash Qx : T \cdot \phi \ prop$ iff $\Gamma, x : T \vdash \phi \ prop$ where $\circ \ = \forall, \exists$
5. $\Gamma \vdash t =_T s \ prop$ iff $\Gamma \vdash t : T$ and $\Gamma \vdash s : T$

Proof. The directions from right to left follow immediately from the rules. For the other direction, we use induction on the structure of derivations. Consider part 1. If the conclusion follows from the formation rules for the connective, the result is immediate. If the conclusion is the result of a structural rule, the result follows from using the structural rule itself. For example, suppose the last step in the derivation is the following instance of an application of W_1.

$$\frac{\Gamma \vdash T \ type \qquad \Gamma \vdash \phi \wedge \psi \ prop}{\Gamma, x : T \vdash \phi \wedge \psi \ prop}$$

Consider the premises. By the second premise and 1 and induction, we may suppose that $\Gamma \vdash \phi \ prop$ and $\Gamma \vdash \psi \ prop$. By the structural rule itself, $\Gamma, x : T \vdash \phi \ prop$ and $\Gamma, x : T \vdash \psi \ prop$. Hence, $\Gamma, x : T \vdash \phi \wedge \psi \ prop$. Next consider implication. Suppose the conclusion follows from the formation rules for the connective. Suppose that $\Gamma \vdash \phi \rightarrow \psi \ prop$. By the formation rule, $\Gamma, \phi \vdash \psi \ prop$. For the structural rules we illustrate with

$$\frac{\Gamma \vdash T \ type \qquad \Gamma \vdash \phi \rightarrow \psi \ prop}{\Gamma, x : T \vdash \phi \rightarrow \psi \ prop}$$

By the second premise, 2 and induction, we may suppose that $\Gamma, \phi \vdash \psi\ prop$. By the structural rule itself $\Gamma, x : T, \phi \vdash \psi\ prop$. Hence, by the formation rule, $\Gamma, x : T \vdash \phi \to \psi\ prop$. Parts 3 and 4 follow exactly the same pattern of argument. For part 5, the governing rule is the following.

$$\mathbf{E_1} \quad \frac{\Gamma \vdash T\ type}{\Gamma, x_1 : T, y : T \vdash x =_T y\ prop}$$

Assume $\Gamma \vdash t : T$ and $\Gamma \vdash s : T$. By **Sub**, we obtain $\Gamma \vdash t =_T s\ prop$. Conversely, assume $\Gamma \vdash t =_T s\ prop$. This must follow by an application of **Sub** where $\Gamma, x_1 : T, y : T \vdash x =_T y\ prop$ and $\Gamma \vdash t : T$ and $\Gamma \vdash s : T$. ■.

We may now establish the main result i.e., the coherence of the logic.

Theorem 9 ***(Coherence)***

1. If $\Gamma \vdash \phi$, then $\Gamma \vdash \phi\ prop$,
2. If $\Gamma \vdash t : T$, then $\Gamma \vdash T\ type$,
3. If $\Gamma, x : T, \Delta \vdash \Theta$, then $\Gamma \vdash T\ type$,
4. If $\Gamma, \phi, \Delta \vdash \Theta$, then $\Gamma \vdash \phi\ prop$

Proof. By simultaneous induction (for each of the four parts) on the structure of derivations. Most of the cases are routine.

For part 1, we first consider one of the most simply cases.

$$\mathbf{L_2} \quad \frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi}$$

By induction,

$$\Gamma \vdash \phi\ prop \qquad \Gamma \vdash \psi\ prop$$

Now use the conjunction formation rule. Next consider disjunction. For the introduction rule,

$$\mathbf{L_7} \quad \frac{\Gamma \vdash \phi \qquad \Gamma \vdash \psi\ prop}{\Gamma \vdash \phi \vee \psi}$$

we note that by the first premise and induction, $\Gamma \vdash \phi\ prop$. Using the second premise and the formation rule for disjunction, we are done. Next consider the disjunction elimination rule.

$$\frac{\Gamma \vdash \phi \vee \psi \qquad \Gamma, \phi \vdash \eta \qquad \Gamma, \psi \vdash \eta \qquad \Gamma \vdash \eta\ prop}{\Gamma \vdash \eta}$$

The last premise of the rule gives us the result. Finally consider the implication rules.

$$\mathbf{L_{13}} \quad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \to \psi} \qquad \mathbf{L_{14}} \quad \frac{\Gamma \vdash \phi \to \psi \qquad \Gamma \vdash \phi}{\Gamma \vdash \psi}$$

We first consider the elimination rule. By the previous lemma, the first premise yields that

$$\Gamma \vdash \phi \; prop \qquad\qquad \Gamma \vdash \psi \; prop$$

By $\mathbf{A_2}$ $\Gamma, \phi \vdash \psi \; prop$. So we are done by the formation rule for implication. For the introduction rule we use the premise and induction to yield the required $\Gamma, \phi \vdash \psi \; prop$. Next, consider the negation rules.

$$\mathbf{L_{16}} \quad \frac{\Gamma, \phi \vdash \Omega}{\Gamma \vdash \neg\phi} \qquad\qquad \mathbf{L_{17}} \quad \frac{\Gamma, \neg\phi \vdash \Omega}{\Gamma \vdash \phi}$$

For the first we employ induction with part 4 of present theorem to yield $\Gamma \vdash \phi \; prop$.We then use the formation rule for negation. For the second, we use induction, part 4 of present theorem to yield $\Gamma \vdash \neg\phi \; prop$. We then use part 3 of the previous lemma.

Finally, for part 1, we examine the quantifiers. First consider the existential introduction rule.

$$\frac{\Gamma \vdash \phi[t/x] \qquad \Gamma \vdash t : T \qquad \Gamma, x : T \vdash \phi \; prop}{\Gamma \vdash \exists x : T \cdot \phi}$$

By the assumption, $\Gamma, x : T \vdash \phi \; prop$. By the formation rule for the existential quantifier, $\Gamma \vdash \exists x : T \cdot \phi \; prop$. For the existential elimination rule:

$$\frac{\Gamma \vdash \exists x : T \cdot \phi \qquad \Gamma, x : T, \phi \vdash \eta \qquad \Gamma \vdash \eta \; prop}{\Gamma \vdash \eta}$$

we have only to observe that the premises $\Gamma \vdash \eta \; prop$ is the required conclusion. Next consider,

$$\mathbf{L_{22}} \quad \frac{\Gamma, x : T \vdash \phi}{\Gamma \vdash \forall x : T \cdot \phi}$$

By the premise and induction we have: $\Gamma, x : T \vdash \phi \; prop$, as required for the universal formation rule. For the elimination rule

$$\mathbf{L}_{23} \quad \frac{\Gamma \vdash \forall x : T \cdot \phi \qquad \Gamma \vdash t : T}{\Gamma \vdash \phi[t/x]} \;,$$

by the first premise and the previous lemma we have: $\Gamma, x : T \vdash \phi \; prop$. By the second premise and Sub, we have $\Gamma \vdash \phi[t/x] \; prop$.

For part 2, consider rule **F;** i.e., suppose that the last step is

$$\mathbf{F} \qquad \frac{\Gamma \vdash T_1 \; type \qquad \Gamma, x_1 : T_1, ..., x_i : T_i \vdash T_{i+1} \; type, \; 1 \leq i < n}{\Gamma \vdash x_1 : T_1, ..., x_{n-1} : T_{n-1} \vdash F(x_1, ..., x_{n-1}) : T_n}$$

By the premise we are done.

For part 3, the substantial case concerns the following instance.

$$\mathbf{W}_1 \qquad \frac{\Gamma, \Delta \vdash \Theta \qquad \Gamma \vdash T \; type}{\Gamma, x : T, \Delta \vdash \Theta}$$

It follows immediately that $\Gamma \vdash T\ type$. If a substitution is involved at the last step i.e.,

$$\frac{\Gamma, y : S, x : T, \Delta \vdash \Theta \qquad \Gamma \vdash s : S}{\Gamma, x : T[s/y], \Delta[s/y] \vdash \Theta[s/y]}$$

By induction,

$$\Gamma, y : S \vdash T\ type$$

Now use sub to yield $\Gamma \vdash T[s/y]\ type$. The same argument works for part 4 and

$$\mathbf{W}_2\ \frac{\Gamma, \Delta \vdash \delta \qquad \Gamma \vdash \phi\ prop}{\Gamma, \phi, \Delta \vdash \delta}$$

■

TPL is a generalization of a standard many-sorted logic in two ways. First, the types may be given axiomatically, and so it generalizes the simple fixed structure of standard many-sorted logic. Secondly, the variables of the theory range freely over the types. This has the knock-on effect that the grammatical legitimacy of the various syntactic constructs not only depends upon the types, but also depends dynamically on them; i.e., the expressions are only well formed relative to an assignment of types to the variables. This is a *Curry* (after Haskell Curry) approach to typing [2]. And for the natural and elegant development of our quite rich range of theories, we need all this flexibility. So the slightly complex nature of our logical framework will eventually reap its rewards.

11 Brief Bibliography

References

[1] Azcel, P. Frege Structures, the Notions of Proposition, Truth and Set. Studies in Logic and the Foundations of Mathematics. Volume 101, 1980, Pages 31-59. The Kleene Symposium.

[2] Barendregt, H.P. Lambda Calculus With Types. Oxford, 1992.

[3] Beeson, M.J. Foundations of Constructive Mathematics, Springer-Verlag Berlin, 1985.

[4] Coquand and Huet: The calculus of constructions. Technical Report 530, INRIA, Centre de Rocquencourt, 1986.

[5] Combinatory Logic Vol 1,2.

[6] Constructive Theories of Functions and Classes.

[7] Martin-Lof, P. An intuitionistic theory of sets, predicative part. In Logic Colloquium, 73. North-Holland, Amsterdam, 1975.

[8] Thomason, R. H. (Ed.) (1974). Formal Philosophy, Selected Papers of Richard Montague. New Haven and London: Yale University Press.

[9] Thomason, R. A Model Theory for Propositional Attitudes. Linguistics and Philosophy 4, pp47-70.

[10] Thompson, S. Type Theory. Addison-Wesley, Reading, MA, 1991.

[11] Turner, R. Type inference for set theory. Theor. Comput. Sci. 266(1–2): 951–974, 2001

[12] Turner, R . Semantics and Stratification. Journal of Logic and Computation, Vol. 15, no 2.

[13] Turner, R. A Theory of Properties. The Journal of Symbolic Logic, Vol. 52, No. 2. (Jun., 1987), pp. 455-472.

[14] Turner, R. Logics of Truth. Notre Dame Journal of Formal Logic. Volume 31, Number 2, Spring 1990.

[15] Turner, R. Computable Models. Journal of Logic and Computation. Volume18, Issue2 Pp. 283-318. Oxford.

[16] Turner, R (1997) Reading between the lines in constructive type theory. Journal of Logic and Computation 7: 2. 229-250

[17] Turner, R. Lazy Theories of Operations and Types Journal of Logic and Computation 3: 1. 77-102

[18] Turner, R. Weak theories of operations and types JLogicComp. 6: 1. 5-31

[19] Whitehead, A. N. and B. Russell (1925). Principia Mathematica (second ed.). Cambridge, England: Cambridge University Press. Three volumes.

[1] This name has been used before but for a different logical framework.