2003

Intersection type systems and logics related to the Meyer-Routley system B+

Martin W. Bunder
University of Wollongong, mbunder@uow.edu.au

Intersection type systems and logics related to the Meyer-Routley system B+

Abstract

Some, but not all, closed terms of the lambda calculus have types; these types are exactly the theorems of intuitionistic implicational logic. An extension of these simple (→) types to intersection (or →∧) types allows all closed lambda terms to have types. The corresponding →∧ logic, related to the Meyer–Routley minimal logic B+ (without ∨), is weaker than the →∧ fragment of intuitionistic logic. In this paper we provide an introduction to the above work and also determine the →∧ logics that correspond to certain interesting subsystems of the full →∧ type theory.

Keywords

system, routley, meyer, b, related, intersection, logics, systems, type

Disciplines

Engineering | Science and Technology Studies

Publication Details

Bunder, M. (2003). Intersection type systems and logics related to the Meyer-Routley system B+. The Australasian Journal of Logic, 1 43-55.

Intersection Type Systems and Logics Related to the Meyer–Routley System B^+

Martin Bunder
Department of Mathematics
University of Wollongong
mbunder@uow.edu.au

Received by Greg Restall
Published September 16, 2003
http://www.philosophy.unimelb.edu.au/ajl/2003

Abstract: Some, but not all, closed terms of the lambda calculus have types; these types are exactly the theorems of intuitionistic implicational logic. An extension of these simple $(\to)$ types to intersection (or $\to\wedge$) types allows all closed lambda terms to have types. The corresponding $\to\wedge$ logic, related to the Meyer–Routley minimal logic B^+ (without $\vee$), is weaker than the $\to\wedge$ fragment of intuitionistic logic. In this paper we provide an introduction to the above work and also determine the $\to\wedge$ logics that correspond to certain interesting subsystems of the full $\to\wedge$ type theory.

1 Simple Typed Lambda Calculus

In standard mathematical notation "$f : \alpha \to \beta$" stands for "f is a function from α into β." If we interpret ":" as "$\in$" we have the rule:

$$\frac{f : \alpha \to \beta \quad t : \alpha}{f(t) : \beta}$$

This is one of the formation rules of typed lambda calculus, except that there we write ft instead of $f(t)$. In λ-calculus, $\lambda x.M$ represents the function f such that $fx = M$. This makes the following rule a natural one:

$$\frac{\begin{matrix}[x : \alpha] \\ \vdots \\ M : \beta\end{matrix}}{\lambda x.M : \alpha \to \beta}$$

We now set up the λ-terms and their types more formally.

DEFINITION 1 (λ-TERMS)

1. If x is a variable, x is a λ-term.
2. If M and N are λ-terms so is (MN) (APPLICATION).
3. If M is a λ-term and x a variable, $\lambda x.M$ is a λ-term. (λ-ABSTRACTION).

DEFINITION 2 (FREE AND BOUND VARIABLES) Any occurrence of a variable x in a subterm $\lambda x.N$ of M is a BOUND occurrence. Any occurrence of x in M that is not bound is a FREE occurrence. $FV(M)$ is the set of free variables occurring in M. If $FV(M) = \emptyset$, M is said to be CLOSED.

DEFINITION 3 ($\rightarrow$ TYPES)

1. $a, b, c, \ldots$ are ATOMIC TYPES.
2. If α and β are types, then so is $(\alpha \rightarrow \beta)$. $(\alpha \rightarrow \beta)$ is an ARROW TYPE.

DEFINITION 4 (TYPE ASSIGNMENTS, CONTEXTS) If M is a λ-term and α a type, $M : \alpha$ is a TYPE ASSIGNMENT. A CONTEXT is a set of type assignments where the terms are distinct variables. Contexts are denoted by $\Delta, \Delta', \Delta_1, \Delta_2, \ldots$

DEFINITION 5 (THE TYPE ASSIGNMENT SYSTEM TA_λ) $\rightarrow$ types are assigned to λ-terms as follows:

(VAR) $$\Delta, x : \alpha \vdash x : \alpha$$

($\rightarrow$E) $$\frac{\Delta \vdash M : \alpha \rightarrow \beta \quad \Delta \vdash N : \alpha}{\Delta \vdash MN : \beta}$$

($\rightarrow$I) $$\frac{\Delta, x : \alpha \vdash M : \beta}{\Delta \vdash \lambda x.M : \alpha \rightarrow \beta}$$

We will sometimes write "$\vdash_\lambda$" for the relation $\vdash$ of this system, to distinguish it from other consequence relations.

DEFINITION 6 (REDUCTION, NORMAL FORM) λ-terms REDUCE when parts are replaced as follows:

(β) $(\lambda x.M)N \ \triangleright\ [N/x]M$

(η) $\lambda x.Mx \ \triangleright\ M$ (if $x \notin FV(M)$).

A λ-term, no part of which can be reduced by (β) or (η), is said to be in STRONG NORMAL FORM. If a term can be reduced to a term in strong normal form it is said to HAVE STRONG NORMAL FORM.

(For more details on the λ-calculus see Hindley and Seldin [11].)

Example 7 Consider contexts $\Delta = \{x : a,\ y : a \to a \to b,\ z : (a \to b) \to c\}$ and $\Delta' = \{y : a \to a \to b,\ z : (a \to b) \to c\}$. We have the following type assignment:

$$\dfrac{\Delta' \vdash z : (a \to b) \to c \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash x : a \quad \Delta \vdash y : a \to a \to b}{\Delta \vdash yx : a \to b}\,(\to E) \qquad \Delta \vdash x : a}{\Delta \vdash yxx : b}\,(\to E)}{\Delta' \vdash \lambda x.yxx : a \to b}\,(\to I)}{}}{\dfrac{\dfrac{\Delta' \vdash z(\lambda x.yxx) : c}{y : a \to a \to b \vdash \lambda z.z(\lambda x.yxx) : ((a \to b) \to c) \to c}\,(\to I)}{\vdash \lambda y.\lambda z.z(\lambda x.yxx) : (a \to a \to b) \to ((a \to b) \to c) \to c}\,(\to I)}\,(\to E)$$

We note that, looking only at the types in the above type assignment, we have a natural deduction style proof of a theorem of the intuitionistic implicational logic $H_{\to}$. The final term $\lambda y.\lambda z.z(\lambda x.yxx)$ is a very compact representation of the whole proof. Each application represents a modus ponens step and each λ-abstraction a use of the $\to$ introduction rule.

This applies in general:

Theorem 8 (Equivalence of TA_{λ} and $H_{\to}$)

$$(\exists M) \vdash_{\lambda} M : \alpha \quad \Leftrightarrow \quad \vdash_{H_{\to}} \alpha$$

(For details on TA_{λ}, see Hindley [10].)

2 Intersection Types

There are closed terms that do not have a simple type. For example, for the term $\lambda x.xx$ to have a type, we must have $x : \alpha \to \beta$ as well as $x : \alpha$, which is impossible in TA_{λ}.

An *intersection type assignment* $x : (\alpha \to \beta) \wedge \alpha$ allows $x : \alpha \to \beta$ as well as $x : \alpha$ and so $xx : \beta$ and $\lambda x.xx : (\alpha \to \beta) \wedge \alpha \to \beta$. This is set up formally as follows:

Definition 9 ($\to\wedge$ or Intersection Types)

1. $a, b, c, \ldots$ are types.
2. If α and β are types, so are $(\alpha \to \beta)$ and $(\alpha \wedge \beta)$.

Definition 10 (The Type Assignment System $TA_{\lambda\wedge}$)
Types are assigned to λ-terms by (Var), ($\to$E), ($\to$I) and the following rules:

$$\frac{\Delta \vdash M : \alpha \quad \Delta \vdash M : \beta}{\Delta \vdash M : \alpha \wedge \beta}\,(\wedge I) \qquad \frac{\Delta \vdash M : \alpha \wedge \beta}{\Delta \vdash M : \alpha} \quad \frac{\Delta \vdash M : \alpha \wedge \beta}{\Delta \vdash M : \beta}\,(\wedge E)$$

$$\frac{\Delta \vdash \lambda x.Mx : \alpha}{\Delta \vdash M : \alpha}\ x \notin FV(M) \quad (\eta)$$

We will sometimes write "$\vdash_{\lambda\wedge}$" for the $\vdash$ of this system.

Example 11 Let $\Delta = \{x : (a \to b) \wedge (a \to c), y : a\}$. We have the following type assignment:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Delta \vdash x : (a \to b) \wedge (a \to c)}{\Delta \vdash x : a \to b}(\wedge E) \quad \Delta \vdash y : a}{\Delta \vdash xy : b}(\to E) \quad \dfrac{\dfrac{\Delta \vdash x : (a \to b) \wedge (a \to c)}{\Delta \vdash x : a \to c}(\wedge E) \quad \Delta \vdash y : a}{\Delta \vdash xy : c}(\to E)}{\Delta \vdash xy : b \wedge c}(\wedge I)}{\dfrac{x : (a \to b) \wedge (a \to c) \vdash \lambda y.xy : a \to b \wedge c}{x : (a \to b) \wedge (a \to c) \vdash x : a \to b \wedge c}(\eta)}(\to I)}{\vdash \lambda x.x : (a \to b) \wedge (a \to c) \to a \to b \wedge c}(\to I)$$

The $\to\wedge$ type theory was first introduced by Coppo and Dezani [5]. A useful survey article is Hindley [9].

An alternative formulation of $TA_{\lambda\wedge}$ replaces $(\wedge E)$ and (η) by

$$\frac{\Delta \vdash M : \alpha \quad \alpha \leq \beta}{\Delta \vdash M : \beta}\ (\leq)$$

where $\leq$ is a binary relation over types given by:

Definition 12 $(\leq)$

Axioms	Rules
1. $\alpha \leq \alpha$	5. $\alpha \leq \beta$ & $\beta \leq \gamma \;\Rightarrow\; \alpha \leq \gamma$
2. $\alpha \wedge \beta \leq \alpha$	6. $\alpha \leq \beta$ & $\alpha \leq \gamma \;\Rightarrow\; \alpha \leq \beta \wedge \gamma$
3. $\alpha \wedge \beta \leq \beta$	7. $\alpha \leq \beta$ & $\sigma \leq \tau \;\Rightarrow\; \beta \to \sigma \leq \alpha \to \tau$
4. $(\alpha \to \beta) \wedge (\alpha \to \gamma) \leq (\alpha \to \beta \wedge \gamma)$	

The standard (but equivalent) formulation replaces rule 6 by

$$\alpha \leq \alpha \wedge \alpha \quad \text{and}$$

$$\alpha \leq \beta \;\&\; \delta \leq \gamma \;\Rightarrow\; \alpha \wedge \delta \leq \beta \wedge \gamma.$$

We can define $=$ by

Definition 13 $(=)$ $\alpha = \beta$ is $\alpha \leq \beta$ & $\beta \leq \alpha$.

The commutative and associative properties for $\wedge$ are easy to prove.

3 B^+ THE $\leq$-LOGIC

Meyer realised that the $\leq$-postulates relate to his and Routley's minimal relevance logic B^+ [13, 14].

DEFINITION 14 THE LOGIC B^+ (WITHOUT $\vee$)

AXIOMS	
a1.	$\vdash \alpha \to \alpha$
a2.	$\vdash \alpha \wedge \beta \to \alpha$
a3.	$\vdash \alpha \wedge \beta \to \beta$
a4.	$\vdash (\alpha \to \beta) \wedge (\alpha \to \gamma) \to (\alpha \to \beta \wedge \gamma)$
RULES	
$(\to E)$	$\alpha \to \beta, \alpha \;\Rightarrow\; \beta$
$(\wedge I)$	$\alpha, \beta \;\Rightarrow\; \alpha \wedge \beta$
SUFFIXING	$\alpha \to \beta \;\Rightarrow\; (\beta \to \gamma) \to \alpha \to \gamma$
PREFIXING	$\beta \to \gamma \;\Rightarrow\; (\alpha \to \beta) \to \alpha \to \gamma$

We will sometimes write "$\vdash_{B^+}$" for the $\vdash$ of this system.

THEOREM 15 (EQUIVALENCE OF $\leq$ AND B^+)

1. If $\alpha \leq \beta$ then $\vdash_{B^+} \alpha \to \beta$.
2. If $\vdash_{B^+} \alpha$ then there are α_i and β_i where $\alpha \equiv (\alpha_1 \to \beta_1) \wedge \cdots \wedge (\alpha_n \to \beta_n)$ and for each i, $\alpha_i \leq \beta_i$.

PROOF Venneri [15], Theorem 4.5. ♯

Theorem 16 below, which is proved in [2], provides us with a decision procedure for B^+.

THEOREM 16 (DECISION PROCEDURE FOR B^+)

$\alpha \leq \beta$ if and only if α is some intersection of atomic types $a_1, \ldots, a_n$ and arrow types $(\alpha_1 \to \gamma_1), \ldots, (\alpha_m \to \gamma_m)$ and β is some intersection of atomic types $b_1, \ldots, b_k$ and arrow types $(\beta_1 \to \delta_1), \ldots, (\beta_e \to \delta_e)$ such that, (i) $\{b_1, \cdots, b_k\} \subseteq \{a_1, \ldots, a_n\}$ and (ii) for each i where $1 \leq i \leq e$, there are $j_1, \ldots, j_r \in \{1, \cdots, m\}$ where $\alpha_{j_1} \wedge \cdots \wedge \alpha_{j_r} \geq \beta_i$ and $\gamma_{j_1} \wedge \cdots \wedge \gamma_{j_r} \leq \delta_i$.

EXAMPLE 17 The following formula is a theorem of B^+

$$[(a \to b) \wedge (a \to c) \to (a \to b \wedge c)] \wedge$$
$$[a \wedge b \wedge g \wedge (a \wedge b \to c) \wedge (c \to a) \wedge (a \to e) \to$$
$$a \wedge (c \wedge a \wedge b \to e \wedge a) \wedge (a \wedge d \wedge b \to e) \wedge (a \wedge b \to c)]$$

since $(a \to b) \wedge (a \to c) \leq a \to b \wedge c$ and $a \wedge d \wedge g \wedge (a \wedge b \to c) \wedge (c \to a) \wedge (a \to e) \leq a \wedge (c \wedge a \wedge b \to e \wedge a) \wedge (a \wedge d \wedge b \to e) \wedge (a \wedge b \to c)$, so the result follows by Theorems 15 and 16.

4 The Logic of $TA_{\lambda\wedge}$

As the types of TA_λ were theorems of $H_\rightarrow$, a natural question arises: What logical system is represented by the types of $TA_{\lambda\wedge}$? This question was answered for a combinatory logic version $TA_\wedge$ of $TA_{\lambda\wedge}$ by Venneri [16] and thus, it was implicitly answered for $TA_{\lambda\wedge}$ using translations to and from λ-terms to combinatory terms [1,2].

Definition 18 (Combinatory Terms)

1. S, K, I and variables are combinatory terms.
2. If X and Y are combinatory terms so is (XY) (application).

Given a λ-term M we can find a corresponding combinatory term M_H and, conversely, for each combinatory term X there is a λ-term X_λ. The process of finding M_H relies on the presence of a bracket abstraction operator λ^*.

Definition 19 (H and λ)

Given λ^*, a bracket abstraction operator, the maps H from λ-terms to combinatory terms, and λ from combinatory terms to λ terms are defined as follows:

λ-terms to combinators	combinators to λ-terms
	$x_\lambda = x$
$x_H = x$	$K_\lambda = \lambda xy.x$
$(\lambda x.M)_H = \lambda^* x.M_H$	$S_\lambda = \lambda xyz.xz(yz)$
$(MN)_H = (M_H N_H)$	$I_\lambda = \lambda x.x$
	$(XY)_\lambda = X_\lambda Y_\lambda$

The details of the abstraction operator λ^* need not concern us here. The relevant requirement for a bracket abstraction operator is that it makes available the following equivalence.

Theorem 20 If M is a λ-term, $M_{\lambda H} = M$.

Proof Curry and Feys [7] or Dezani and Hindley [8]. ♯

The following is one of Venneri's equivalent type assignment systems for combinatory logic [16] that is best suited to our purposes:

Definition 21 (The Type Assignment System $TA^*_\wedge$)

Axioms	Rules
$\Delta \vdash^* I : \alpha \rightarrow \alpha$ $\Delta \vdash^* K : \alpha \rightarrow \beta \rightarrow \alpha$ $\Delta \vdash^* S : (\alpha_1 \rightarrow \beta \rightarrow \gamma) \rightarrow$ $(\alpha_2 \rightarrow \beta) \rightarrow \alpha_1 \wedge \alpha_2 \rightarrow \gamma$	(var), $(\rightarrow E)$, $(\leq)$ and $(\wedge I\text{-}s)$

where the new rule ($\wedge$I-s) is

$$\frac{\Delta \vdash^* X : \alpha \quad \beta = s(\alpha) \quad FV(X) = \emptyset}{\Delta \vdash^* X : \alpha \wedge \beta}$$

where where $s(\alpha)$ is a substitution instance of α.

The Venneri Hilbert-style logic, which we call V, that corresponds to this is:

Definition 22 (The Logic V)

Axioms	
a1.	$\vdash \alpha \to \alpha$
a2.	$\vdash \alpha \to \beta \to \alpha$
a3.	$\vdash (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$
Rules	
(sub-$\wedge$)	Any finite intersection of instances of the same axiom is a theorem of V.
(mp)	$\dfrac{\Delta \vdash \alpha \to \beta \quad \Delta \vdash \alpha}{\Delta \vdash \beta}$
(rmp)	$\dfrac{\vdash_{B^+} \alpha \to \beta \quad \Delta \vdash \alpha}{\Delta \vdash \beta}$

Here Δ is a set of formulas (or types) rather than a context.

The $\vdash$ defined here will sometimes be written as "$\vdash_V$." Venneri then proves:

Theorem 23 $(\exists X) \vdash^* X : \alpha \iff \vdash_V \alpha$

Note that this logic does not have (and cannot have) the full strength $\wedge$I rule:

$$\frac{\Delta \vdash \alpha \quad \Delta \vdash \beta}{\Delta \vdash \alpha \wedge \beta}$$

and that the $\leq$ rule is replaced using the Routley–Meyer logic B^+.
We also have the following connection between $\vdash_\lambda$ and $\vdash^*$:

Theorem 24 (Equivalence of $\vdash_\lambda$ and $\vdash^*$)

(i) $\vdash^* X : \alpha \iff \vdash_\lambda X_\lambda : \alpha$.

(ii) $\vdash_\lambda M : \alpha \iff \vdash^* M_H : \alpha$.

Proof By Venneri [16] Theorem 2.13 and Remark 2.14 and Dezani and Hindley [8] Theorem 3.11. Part (i) of the latter theorem gives (i) above and part (ii) gives (ii). ♯

Theorem 24 shows that the logic V is also the logic of the types of $\mathsf{TA}_{\lambda\wedge}$.

In [2] we proposed a λ-calculus version of Definition 21, $\mathsf{TA}'_{\lambda\wedge}$, from which a natural deduction style logic for $\mathsf{TA}_{\lambda\wedge}$ can be derived. This, of course, will be equivalent to V.

Definition 25 (The system $\mathsf{TA}'_{\lambda\wedge}$)

(VAR)	$\Delta, x : \alpha \vdash' x : \alpha$
$(\rightarrow I)$	$\dfrac{\Delta, x : \alpha \vdash' M : \beta}{\Delta \vdash' \lambda x.M : \alpha \rightarrow \beta}$
$(\rightarrow E)$	$\dfrac{\Delta \vdash' M : \alpha \rightarrow \beta \quad \Delta \vdash' N : \alpha}{\Delta \vdash' MN : \beta}$
$(\wedge I\text{-}s')$	$\dfrac{\Delta \vdash' M : \alpha \quad s(\Delta) \equiv \Delta}{\Delta \vdash' M : \alpha \wedge s(\alpha)}$
(RMP)	$\dfrac{\Delta \vdash' M : \alpha \quad \alpha \leq \beta}{\Delta \vdash' M : \beta}$

From this we define the corresponding natural deduction style logic.

Definition 26 (The logic V′)

(VAR)	$\Delta, \alpha \vdash \alpha$
$(\rightarrow I)$	$\dfrac{\Delta, \alpha \vdash \beta}{\Delta \vdash \alpha \rightarrow \beta}$
$(\rightarrow E)$	$\dfrac{\Delta \vdash \alpha \rightarrow \beta \quad \Delta \vdash \alpha}{\Delta \vdash \beta}$
$(\wedge I\text{-}s')$	$\dfrac{\Delta \vdash \alpha \quad s(\Delta) \equiv \Delta}{\Delta \vdash \alpha \wedge s(\alpha)}$
(RMP)	$\dfrac{\Delta \vdash \alpha \quad \vdash_{B+} \alpha \rightarrow \beta}{\Delta \vdash \beta}$

The $\vdash$ defined here will sometimes be written "$\vdash_{V'}$." We show in [2] that:

Theorem 27 $(\exists M)\Delta \vdash_{\lambda\wedge} M : \alpha \Leftrightarrow (\exists M)\Delta \vdash'_{\lambda\wedge} M : \alpha \Leftrightarrow \Delta' \vdash_{V'} \alpha \Leftrightarrow \Delta' \vdash_{V} \alpha$, where Δ' is Δ with the 'x_i :'s deleted.

5 Intermediate Type Systems

Urzyczyn has shown in [15] that, given a Δ and α, it is not decidable whether there is a term M such that $\Delta \vdash M : \alpha$. Kurata and Takahashi [12] have shown that this property is decidable when the rule $(\wedge I)$ (or $(\wedge I\text{-}s)$ is omitted. The question arises what happens when other rules such as $(\wedge E), (\leq)$ or (η) are omitted, as well as, or instead of $(\wedge I)$? This question is tackled in another paper [4]. What was needed first was an answer to the question: how *many* different intermediate systems are there?

Definition 28 ($\approx_1$)

If A and B are type systems, then $A \approx_1 B$ if and only if

$$(\forall \Delta, \alpha, M)(\Delta \vdash_A M : \alpha \;\Leftrightarrow\; \Delta \vdash_B M : \alpha)$$

Systems with equivalent "inhabitation properties," and so equivalent logics, are given by a different kind of equivalence:

Definition 29 ($\approx_2$)

If A and B are type systems, then $A \approx_2 B$ if and only if

$$(\forall \Delta, \alpha)[(\exists M)\Delta \vdash_A M : \alpha \;\Leftrightarrow\; (\exists M)\Delta \vdash_B M : \alpha]$$

It is these distinct systems that we are interested in here.

Theorem 30 The type systems in each of the following sets, denoted by the rules they have in addition to (var), $(\rightarrow I)$ and $(\rightarrow E)$ are $\approx_1$-equivalent:

1.	$(\wedge I) + (\eta) + (\wedge E), (\wedge I) + (\leq) + (\wedge E),$ $(\wedge I) + (\leq), (\wedge I) + (\leq) + (\wedge E) + (\eta)$
2.	$(\leq) + (\wedge E) + (\eta), (\leq)$
3.	$(\wedge E) + (\eta)$
4.	$(\wedge I) + (\wedge E)$
5.	$(\wedge I) + (\eta)$
6.	$(\wedge I)$
7.	$(\wedge E)$
8.	$-, (\eta)$

The systems, denoted by 1 to 8, are related as in the first graph in Figure 1, with downward lines leading from stronger to weaker systems, and systems not connected by downward lines in either direction (such as 2 and 5, 3 and 5, 3 and 4 etc.) are independent.

Proof Bunder [3] Theorems 2.3, 7.1 and 7.2. ♯

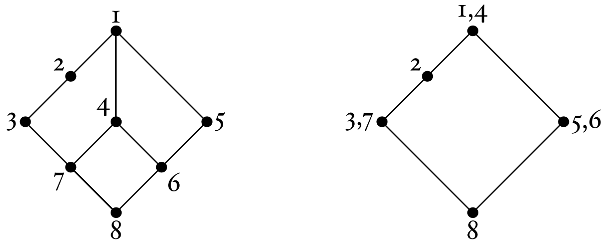

Figure 1: Systems under $\approx_1$ and $\approx_2$, respectively

THEOREM 31 The systems in the sets 1 to 8 in Theorem 30 satisfy: $1 \approx_2 4$, $3 \approx_2 7$ and $5 \approx_2 6$. These are related, with notation as in Theorem 30, in the second graph in Figure 1.

PROOF Bunder [3] Theorems 2.3, 7.1 and 7.2. ♯

6 The Logics of the Intermediate Systems

THEOREM 32 The type systems having (VAR), ($\to$I), ($\to$E) and either (i) ($\wedge$I)$+$ (η)$+$($\wedge$E), (ii) ($\wedge$I)$+$($\leq$), (iii) ($\wedge$I)$+$($\leq$)$+$($\wedge$E), (iv) ($\wedge$I)$+$($\leq$)$+$($\wedge$E)$+$(η) or (v) ($\wedge$I)$+$($\wedge$E) (i.e. those labelled 1. and 4. in Theorem 30) have the Venneri logic V or V$'$.

PROOF Immediate, by way of Theorems 31 and 27. ♯

THEOREM 33 The type systems having (VAR), ($\to$I) and ($\to$E) (with, or without (η)) has the logic $H_{\to}$, but with formulas that may involve $\wedge$.

PROOF Obvious. ♯

THEOREM 34 The type theory based on (VAR), ($\to$I), ($\to$E) and ($\leq$) together with either or both of ($\wedge$E) and (η) (i. e. system 2 in Theorem 30) has a logic based on (VAR), ($\to$I), ($\to$E) and (RMP).

PROOF By an easy induction on the derivation of any $\Delta \vdash M : \alpha$ in the type theory, and an application of Theorem 15. ♯

The remaining systems (3 and 7, on the one hand and 5 and 6 on the other) are $\approx_2$-equivalent to type systems with restrictions on the rules regarding $\leq$.

DEFINITION 35 ($\leq_{-2,3}$ AND $\leq_{2,3}$)

($\leq_{-2,3}$) is the ($\leq$) rule without postulates 2 and 3. ($\leq_{2,3}$) is the ($\leq$) rule with *only* postulates 2 and 3.

THEOREM 36 The type theory based on (VAR), ($\rightarrow$I), ($\rightarrow$E), ($\wedge$I) and (η) is $\approx_2$ equivalent to that based on (VAR), ($\rightarrow$I), ($\rightarrow$E), ($\wedge$I) and ($\leq_{-2,3}$).

PROOF [3] Corollary 5.5. ♯

THEOREM 37 The type theory based on (VAR), ($\rightarrow$I), ($\rightarrow$E) and ($\wedge$E) is $\approx_2$ equivalent to that based on (VAR), ($\rightarrow$ I), ($\rightarrow$ E) and ($\leq_{2,3}$).

PROOF [3] Corollary 5.5. ♯

To find the logics of types of these systems we need some results concerning the restricted ($\leq$) rules and a weaker version of B^+.

DEFINITION 38 (THE LOGIC B^-)

The logic B^- has axioms (a1) and (a4) and all the rules of B^+.

LEMMA 39 (i) If $\alpha \leq_{-2,3}$ then $\vdash_{B^-} \alpha \rightarrow \beta$.

(ii) If $\vdash_{B^-} \alpha$ then $\alpha \equiv (\alpha_1 \rightarrow \beta_1) \wedge \cdots \wedge (\alpha_n \rightarrow \beta_n)$ and for each i, $\alpha_i \leq_{2,3} \beta_i$.

PROOF As for Venneri [16] Theorem 3.4. ♯

The following lemma (with an obvious proof) is required

LEMMA 40 $\alpha \leq_{2,3} \beta$ if and only if for some γ, either $\alpha \equiv \beta \wedge \gamma$ or $\alpha \equiv \gamma \wedge \beta$.

Now we can proceed to our final theorems.

THEOREM 41 The type theory based on (VAR), ($\rightarrow$I), ($\rightarrow$E), ($\wedge$I) and (η) (i. e. systems 5 and 6 of Theorem 30) has the logic of types V', except with B^+ replaced by B^-.

PROOF If, as before, we let Δ' be Δ without the 'x_i :'s we prove by induction on the derivation of

$$\Delta \vdash M : \alpha$$

in this system,

$$\Delta' \vdash \alpha$$

in the given logic. For the type theory we use the system of rules, given in Theorem 36, that includes ($\leq_{-2,3}$). Every proof step taken in the type theory has an obvious counterpart in the logic. In the case of ($\leq_{-2,3}$) this is given by Lemma 39. ♯

THEOREM 42 The type theory based on (VAR), ($\to$I), ($\to$E), and ($\wedge$E) has the logic of types V′, except with (RMP) replaced by, for arbitrary sets of formulas Δ:

$$\Delta \vdash \alpha \wedge \beta \to \alpha$$

$$\Delta \vdash \beta \wedge \alpha \to \alpha$$

PROOF The proof is as for Theorem 41 except that any use of $\leq_{2,3}$ can be replaced by a use of one of the new axiom schemes. ♯

References

[1] Bunder, M. W., "Lambda terms definable as combinators", *Theoretical Computer Science*, Vol. 169 (1996) 3–21.

[2] Bunder, M. W., "Intersection types for lambda-terms and combinators and their logics", *Journal of the Interest Group in Pure and Applied Logics*, Vol. 10 (2002) 357–378.

[3] Bunder, M. W., "A classification of intersection types", *Journal of Symbolic Logic*, Vol. 67 (2002) 353–368.

[4] Bunder, M. W., "The inhabitation problem for intersection types", Preprint, The University of Wollongong, 2003.

[5] Coppo, M., and Dezani, M. "A new type assignment for lambda terms", *Archiv Math. Logik*, Vol. 19 (1978) 139–156.

[6] Coppo, M., Dezani, M. and Sallé, P. "Functional characterization of some semantic equalities inside λ-calculus", *Springer Lecture Notes in Computer Science*, Vol. 71 (1979) 133–146.

[7] Curry, H. B. and Feys, R. *Combinatory Logic*, Vol. 1, North Holland, Amsterdam 1958.

[8] Dezani, M. and Hindley, J. R., "Intersection types for combinatory logic", *Theoretical Computer Science*, Vol. 100 (1992) 303–324.

[9] Hindley, J. K. "Types with intersection, an introduction", *Formal Aspects of Computing*, Vol. 4 (1992) 470–486.

[10] Hindley, J. R. *Basic Simple Type Theory*, Cambridge University Press, 1997.

[11] Hindley, J. R. and Seldin, J. P. *Introduction to Combinators and λ-calculus*, Cambridge University Press, Cambridge 1986.

[12] Kurata, T. and Takahashi, M. "Decidable properties of intersection type systems" in *Lecture Notes in Computer Science*, Vol. 902, TCLA 1995, edited by M. Dezani and G. Plotkin, 1995, 297–311.

[13] Routley, R. and Meyer, R. K. "The semantics of entailment II," *Journal of Philosophical Logic*, Vol. 1, (1972) 53–73.

[14] Routley, R. and Meyer, R. K. "The semantics of entailment III", *Journal of Philosophical Logic*, Vol. 1 (1972) 420–441.

[15] Urzyczyn, P. "The emptiness problem for intersection types" *Proceedings of Logic in Computer Science*, IEEE, 1994.

[16] Venneri, B., "Intersection types as logical formulae" *Journal of Logic and Computation*, Vol.4 (1994) 109–124.