Electronic Notes in Theoretical Computer Science 50 No. 2 (2001) – Proc. BOTH 2001
URL: `http://www.elsevier.nl/locate/entcs/volume50.html` 18 pages

Hyperformulae, Parallel Deductions and Intersection Types

Beatrice Capitani

Dipartimento di Matematica
Università di Siena
Via Del Capitano, 15, Siena, Italy
E-mail: `capitanib@unisi.it`

Michele Loreti   Betti Venneri

Dipartimento di Sistemi ed Informatica
Università di Firenze
Via Lombroso, 6/17, Firenze, Italy
E-mail: `{loreti,venneri}@dsi.unifi.it`

Abstract

We aim at investigating the intersection-type assignment system for lambda calculus, with the Curry-Howard approach. We devise a propositional logic, whose notable characteristic is the presence of the hyperformulae denoting parallel compositions of formulae. As such, this logic formalizes a novel notion of parallel deductions, while forming a simple generalization of the standard natural deduction framework.

We prove that the logical calculus is isomorphic to the intersection type system, by mapping logical deductions into typed lambda terms, encoding those deductions, and conversely. In this context the intersection type constructor, which comes out to be a proof-theoretic operator, is now interpreted as a standard propositional connective.

1 Introduction

Intersection types originated in [6] as an extension of Curry's basic type system. Its notable characteristic is the presence of a new type constructor ($\wedge$) for denoting the intersection on types. The system so devised turns out to be

⋆ This work was partly supported by *MURST COFIN'99 TOSCA* Project and *CNR-GNSAGA*.

extremely powerful, since it allows the typing of all (and only) the strongly normalizing lambda terms.

However a debated question concerns the logical interpretation of this type theory, since intersection types do not fit into the Curry-Howard paradigm.

It is well known that, in the Curry-Howard approach, reading formulae as types, constructive proofs of formulae are mapped into lambda terms having related types and conversely. Thus, for instance, functional type theory and Girard's system F correspond to implicational and second-order logics, respectively.

In this perspective, intersection on types seems to be somewhat esoteric, because of the crucial shape of the introductory rule. Namely, the $\wedge-$ Introduction rule says that a term M has type $\sigma \wedge \tau$ if and only if the *same* term M has both type σ and type τ. Thus, in logical terms, $\wedge$ becomes a proof-functional connective, restricting the classical conjunction; the proof of the $\wedge-$formula depends in an essential way upon intensional aspects of the component subformulae, namely they must be proved by the *same* proof.

It is for giving a logical account of the intersection that a Hilbert-style logic is proposed in [9], where intersection type inference is, however, investigated in the context of Combinatory Logic instead of lambda calculus. In that paper the $\wedge-$Introduction rule is avoided by splitting it into two components, a relevant conjunction and the following inference rule:

(Sub) "any finite intersection of different instances of the same theorem is a theorem"

This solution is unsatisfactory for our goal, because the (Sub)-rule prevents from extending that result to lambda calculus by translating the Hilbert-style logic of [9] into a natural deduction version. In fact, a distinguishing feature of the natural deduction framework is the treatment of assumptions, that are fixed and could loose their status (by being discharged) but not be modified. On the contrary, the (Sub)-rule assumes that any assumption may duplicate in several different instances during the deductive process. On the other hand, the strict relation between natural deduction and lambda calculus is a well-known matter, since the introduction and elimination rules for implication correspond quite naturally to the $\lambda-$abstraction and application rules of term formation in assigning types to $\lambda-$calculus.

In the present paper we define a natural deduction propositional logic and we prove that it is isomorphic to the intersection type assignment system for lambda calculus. The novelty of this logic comes up from its syntax, involving hyperfomulae as well as implicative and conjunctive formulae. Hyperformulae are intended as sequences of formulae composed by a parallel operator, so that a notion of parallel deductions is represented inside the logical system without requiring any proof-functional condition in the deduction rules. As a result, we exploit derivability of hyperformulae for giving a logical interpretation of the intersection as a standard truth-functional connective.

The paper is organized as follows. In section 2 we briefly outline the Intersection Type Assignment system for $\lambda-$calculus ($TA_\wedge$). In section 3 the logic HL is presented, its main properties are proved and the degree of parallelism represented here is discussed. In section 4 logical proofs are decorated by lambda terms as a technical tool for proving, in section 5, the isomorphism between HL and $TA_\wedge$.

2 Intersection Type Assignment for Lambda-calculus

This section outlines the intersection type assignment system for λ-calculus. Intersection types have been introduced in [7] and [5] to overcome some weaknesses of Curry's basic system, while retaining the normalization property. The arrow-based type language of Curry's system is enriched by a new type constructor, $\wedge$, for denoting the intersection of two types, and the inference system is extended with rules for assigning $\wedge$-types to $\lambda-$terms.

We remark that there are several formulations of the intersection type theory in the literature. For instance, the complete system, presented in [5], also considers the universal type ω and a preorder relation on types.

The system considered here, denoted by $TA_\wedge$, is the simplest one and only involves the basic rules for introducing and eliminating the $\wedge-$type constructor. The main motivation in choosing the simple system $TA_\wedge$ relies on the fact that the present paper aims at investigating, in logical terms, the $\wedge-$derivability without dealing with constants or others features.

We briefly recall that λ-terms are defined by the following syntax:

$$M, N ::= x|\lambda x.M|MN$$

Definition 2.1 Assume that we have infinitely many type variables $\alpha, \beta, \gamma, \ldots$. The set $\mathcal{T}$ of types is inductively defined thus:

- type variables are types,
- if σ, τ are types, then so are $\sigma \to \tau$ *(arrow type)* and $\sigma \wedge \tau$ *(intersection type)*.

Notation 1 *Parentheses are omitted from types assuming that $\to$ associates to right and $\wedge$ has precedence over $\to$. Moreover, intersection types $\sigma_1 \wedge \ldots \wedge \sigma_n$ are considered equal up to permutations and repetitions of σ_i's.*

Definition 2.2 (The system $TA_\wedge$)

- **A statement** is an expression of the form $M : \tau$ where M *(subject)* is a λ term and τ *(predicate)* is a type.
- **A basis** B is a finite set of statements whose subjects are all distinct variables.

  We will use $B, x : \tau$ for $B \cup \{x : \tau\}$, where x does not belong to B.
- **A statement** $M : \tau$ is derivable from a basis B if $B \vdash_\wedge M : \tau$ can be proved using the following axioms and inference rules.

Axioms

$$B, x : \alpha \vdash_\wedge x : \alpha \quad (Var)$$

Rules

$$\frac{B \vdash_\wedge M : \sigma \quad B \vdash_\wedge M : \tau}{B \vdash_\wedge M : \sigma \wedge \tau} \quad (\wedge I)$$

$$\frac{B \vdash_\wedge M : \sigma \wedge \tau}{B \vdash_\wedge M : \sigma \ (\tau)} \quad (\wedge E)$$

$$\frac{B, x : \sigma \vdash_\wedge M : \tau}{B \vdash_\wedge \lambda x.M : \sigma \to \tau} \quad (\to I)$$

$$\frac{B \vdash_\wedge M : \sigma \to \tau \quad B \vdash_\wedge N : \sigma}{B \vdash_\wedge MN : \tau} \quad (\to E)$$

We write $\Delta : B \vdash_\wedge M : \sigma$ to denote a proof Δ of $B \vdash_\wedge M : \sigma$.

The system $TA_\wedge$ enjoys the main property that all the strongly normalizing $\lambda-$terms are typeable and viceversa.

Further details on Lambda-Calculus and Intersection Type Theory can be found in [3] and [4], respectively.

3 The Logic HL of Hyperformulae

In this section the propositional logic HL is defined in natural deduction-style. This calculus will be proved to be isomorphic to $TA_\wedge$ in section 5, so providing a logical setting for interpreting intersection-type assignment.

The novelty of HL comes up from its syntax, involving both formulae and *hyperformulae*. As usual, formulae are built from propositional variables by means of connectives, namely $\to$ (*implication*) and $\wedge$ (*conjunction*) in our case. Then hyperformulae are defined as finite sequences of formulae, composed by the *parallel constructor* $|$. As such, the system HL represents a simple generalization of the standard natural deduction framework, while capturing a novel notion of parallel deductions.

As formal definitions will clarify, both the order and the position of each formula in a hyperformula are significant. However, during the derivation, a component of a hyperformula can move to a different position for fusing with another component. For this reason, we use a special marker, ε, to denote a hole in a parallel composition. In other words ε can be considered as a logical constant, whose meaning is just the *lack of information*. It does not contribute to forming implicative and conjunctive formulae, but to forming hyperformulae.

In section 3.3 we will discuss the parallel operator $|$, looking at formulae as processes and the relation between the method of hyperformulae and that of hypersequents [1,2].

3.1 Syntax

Definition 3.1 (Formulae) Let $\mathcal{V}$ be a denumerable set of variables. The set $\mathcal{F}$ of formulae is inductively defined thus:

- $\mathcal{V} \subseteq \mathcal{F}$
- $\varepsilon \in \mathcal{F}$
- $\sigma,\ \tau \in \mathcal{F} \setminus \{\varepsilon\} \implies \sigma \to \tau,\ \sigma \wedge \tau \in \mathcal{F}$

As for types, we assume that $\wedge$ binds stronger then $\to$ and $\to$ associates to the right. Furthermore, we assume that $\wedge$-formulae are equal up to contraction (i.e. $\sigma \wedge \sigma = \sigma$), commutative and associative properties; for instance $\sigma \wedge (\tau \wedge \sigma) = \sigma \wedge \tau$.

Definition 3.2 (Hyperformulae) A *hyperformula* φ is a structure of the form

$$\varphi = \sigma_1 | \cdots | \sigma_n \quad (1 \le n)$$

where $\sigma_1 \cdots \sigma_n$ are formulae and $\sigma_i \neq \varepsilon$ for some i $(1 \le i \le n)$. Let $\mathcal{H}$ denote the set of hyperformulae.

Observe that, in the previous definition, the condition $\sigma_i \neq \varepsilon$ is assumed for simplicity, in order to avoid dealing with totally empty hyperformulae, like $\varepsilon|\varepsilon$.

Notation 2

- *Propositional variables are denoted by* $\alpha, \beta, \gamma, \delta$*. Formulae and hyperformulae are denoted by* σ, τ, ρ *and* φ, ψ*, respectively (with or without subscripts).*
- *Given a hyperformula* $\varphi = \sigma_1 | \cdots | \sigma_n$*, each* σ_i *is called a* component *of* φ*, and this is denoted by* $\sigma_i \in \varphi$ $(i = 1, \ldots, n)$*.*
- *If* $\varphi = \sigma_1 | \cdots | \sigma_n$ *then we write* $\varphi|\sigma$ *for* $\sigma_1 | \cdots | \sigma_n | \sigma$*.*
- *The function length* $: \mathcal{H} \to \mathbf{N}$ *is defined on the structure of hyperformulae, that is* $length(\varphi) = 1+$*(the number of* $|$ *operators occurring in* φ*).*
- *We write* $(\varphi)_i$ *to denote the selection of the i-th component of* φ*.*
  Namely $(\sigma_1 | \ldots | \sigma_n)_i = \sigma_i$ *if* $1 \le i \le n$*,* ε *otherwise.*

We observe that $\varphi|\varepsilon \neq \varphi$. However the $i-$th selection function is defined as a total function, hence $(\varphi)_i = \varepsilon$ if $i \ge length(\varphi)$, only for technical reasons concerning definitions and proofs of section 3.2.

Let us formalize a special kind of substitution for denoting the replacement of components of hyperformulae.

Definition 3.3 (Component substitution)

- $\varphi[i \mapsto \tau] = \varphi$ if $i > length(\varphi)$, otherwise it is the hyperformula ψ such that:
  - $length(\psi) = length(\varphi)$;
  - for every $j \neq i$, $(\psi)_j = (\varphi)_j$;
  - $(\psi)_i = \tau$.

- $\varphi[\sigma \mapsto \tau] = \psi$ where ψ is such that:
  - $\cdot$ $length(\psi) = length(\varphi)$;
  - $\cdot$ $(\psi)_i = (\varphi)_i$, if $(\varphi)_i \neq \sigma$
  - $\cdot$ $(\psi)_i = \tau$, if $(\varphi)_i = \sigma$ $(1 \leq i \leq length(\varphi))$.

Definition 3.4

i) A *context* Γ of *assumptions* is any finite multiset of hyperformulae.

ii) We generalize the function $()_i$ to the context Γ in the following way:

$$(\Gamma)_i = \begin{cases} \emptyset & \text{if } \Gamma = \emptyset \\ \{(\varphi)_i\} \uplus (\Gamma')_i \ ^1 & \text{if } \Gamma = \{\varphi\} \uplus \Gamma' \end{cases}$$

Where not ambiguous, we will write Γ_i for $(\Gamma)_i$.

iii) We say that Γ is $i-j-$*monovalent* $(i, j \geq 1)$ if and only if

$$\forall \varphi \in \Gamma.\ (\varphi)_i = (\varphi)_j$$

The Hyperformulae Logic, denoted as HL, is a natural deduction style logic. The following is the inductive definition of the HL-consequence relation.

Definition 3.5 (HL-derivability) The relation $\vdash$ is defined by the following axioms and rules.

Axiom

$$\Gamma, \varphi \vdash \varphi \quad (Ax)$$

$|-$Weakening

$$\frac{\Gamma \vdash \varphi|\sigma}{\Gamma \vdash \varphi} \quad (| - w)$$

$\wedge-$Introduction

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi[i \mapsto (\varphi)_i \wedge (\varphi)_j][j \mapsto \varepsilon]} \quad (\wedge I) \quad 1 \leq i, j \leq length(\varphi)$$

if Γ is $i-j-$monovalent.

$\wedge$-Elimination.

$$\frac{\Gamma \vdash \varphi \quad (\varphi)_i = \sigma \wedge \tau}{\Gamma \vdash \varphi[i \mapsto \sigma]} (\wedge E) \qquad \frac{\Gamma \vdash \varphi \quad (\varphi)_i = \sigma \wedge \tau}{\Gamma \vdash \varphi[i \mapsto \tau]} (\wedge E)$$

$\rightarrow$-Introduction.

$$\frac{\Gamma, \tau_1|\ldots|\tau_n \vdash \sigma_1|\ldots|\sigma_s}{\Gamma \vdash \rho_1|\ldots|\rho_s} (\rightarrow I) \quad (s \leq n),$$

where ρ_i $(1 \leq i \leq s)$ is such that $\rho_i = \tau_i \rightarrow \sigma_i$ if $\sigma_i \neq \varepsilon$, $\rho_i = \varepsilon$ otherwise.

[1] $\uplus$ denotes the standard multiset union

$\to$-Elimination.

$$\frac{\Gamma \vdash \sigma_1|\ldots|\sigma_n \quad \Gamma \vdash \tau_1|\ldots|\tau_n}{\Gamma \vdash \rho_1|\ldots\ |\rho_n}\ (\to E)$$

where ρ_i $(1 \leq i \leq n)$ is such that $\rho_i = \zeta_i$ if $\sigma_i = \tau_i \to \zeta_i$, $\rho_i = \varepsilon$ otherwise.

We write $\mathcal{D}:\ \Gamma \vdash \varphi$ to denote that $\Gamma \vdash \varphi$ is provable by the proof $\mathcal{D}$ using the axiom and rules defined above.

The $(|-w)$ rule is the only structural rule on parallel deductions, a kind of internal weakening that allows to drop a final component in a hyperformula.

During the deductive process, a hyperformula can loose some final components (by $|-w$) or render some others *inactive*, by replacing ε to them, but it never increases its length. The underlying idea can be rephrased as

only what was in parallel, will remain in parallel,

that is formalized by the absence of any introductory rule for $|$.

The other rules are logical rules and can be divided into global rules and local rules.

The global ones are $(\to I)$ and $(\to E)$, which involve the whole hyperformula. By $(\to I)$ one may discharge an assumption only if the *same* activity is done in all the other components at the same time. When saying that the performed action is the same, we also mean that what is really discharged is a component of a parallel composition of assumptions. Namely each component of the hyperformula-assumption is discharged by each component of the hyperformula-conclusion.

In the same sense the action of eliminating implication $(\to E)$ in one component of a parallel composition must synchronize with an $(\to E)$ action from all the other components in order to occur.

The local rules are $(\wedge I)$ and $(\wedge E)$ which affect only some components inside a hyperformula.

The $(\wedge I)$ rule is very important because it is the only rule that brings moments of fusion into parallel deductions. Two formulae σ_1 and σ_2, running in parallel in a deduced hyperformula $\sigma_1|\sigma_2$, must have the same *deductive history* (with respect to the applied global rules). If they also depend from equal assumption, that is $(\varphi)_1 = (\varphi)_2$ for every assumption φ, then they can fuse in one conjunctive formula. We recall that our main goal was to provide a logical account of the $\wedge-$connective as a truth-functional connective. Hence we achieved this goal by defining the $(\wedge I)$ rule in a such way that it does not involve any proof-functional condition.

Notation 3 *If $\mathcal{D} : \Gamma \vdash \varphi$ then we write $\Gamma \restriction \mathcal{D}$ to denote the* relevant *context containing all and only the assumptions of Γ that are actually used in $\mathcal{D}$.*

Obviously $\mathcal{D} : \Gamma \vdash \varphi$ implies $\mathcal{D} : \Gamma \restriction \mathcal{D} \vdash \varphi$.

The advantage of considering $\Gamma \restriction \mathcal{D}$ instead of Γ is that the relevant context enlightens the ordered dependency between derived parallel formulae and related parallel assumptions. In fact, by a simple inspection of axiom and rules of Definition 3.5, it is easy to verify that if $\mathcal{D} : \Gamma \upharpoonright \mathcal{D} \vdash \psi$ then for any φ belonging to $\Gamma \upharpoonright \mathcal{D}$:

- $lenght(\varphi) \geq length(\psi)$;
- $(\psi)_i \neq \varepsilon$ implies $(\varphi)_i \neq \varepsilon$;

that is the deduction of $(\psi)_i$ depends on the $i-th$ component of each assumption in $\Gamma \upharpoonright \mathcal{D}$.

Example 3.6 Let $\Gamma = \{(\sigma_1 \to \sigma_2)\wedge(\sigma_1 \to \sigma_3)|(\sigma_1 \to \sigma_2)\wedge(\sigma_1 \to \sigma_3),\ \sigma_1|\sigma_1\}$, notice that Γ is $1-2-$monovalent. The following is a proof in HL.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \sigma_1|\sigma_1 \qquad \dfrac{\dfrac{\Gamma \vdash (\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3)|(\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3)}{\Gamma \vdash (\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3)|\sigma_1 \to \sigma_3}(\wedge E)}{\Gamma \vdash \sigma_1 \to \sigma_2|\sigma_1 \to \sigma_3}(\wedge E)}{\dfrac{\Gamma \vdash \sigma_2|\sigma_3}{\Gamma \vdash \sigma_2 \wedge \sigma_3|\varepsilon}(\wedge I)}(\to E)}{(\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3)|(\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3) \vdash \sigma_1 \to \sigma_2 \wedge \sigma_3|\varepsilon}(\to I)}{\vdash (\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3) \to \sigma_1 \to \sigma_2 \wedge \sigma_3|\varepsilon}(\to I)}{\vdash (\sigma_1 \to \sigma_2) \wedge (\sigma_1 \to \sigma_3) \to \sigma_1 \to \sigma_2 \wedge \sigma_3}(|-w)$$

3.2 Main Syntactic Properties

This section is devoted to state some basic properties that clarify how proofs are constructed in HL and will be used in following sections.

As far as structural rules are concerned, we first notice that Γ is a multiset of assumptions, hence it does not change when its elements are permuted. It is also easy to verify that $\mathcal{D} :\ \Gamma \vdash \varphi$ if and only if $\mathcal{D} :\ \Gamma' \vdash \varphi$ when Γ is equal to Γ' but for repetition of some assumptions.

With regards to the weakening property, we have already observed that $(|-w)$ is a kind of structural rule, which says that any assumption φ can be weakened to $\varphi|\sigma$ (for any formula σ).

Moreover a context Γ of a deduction $\mathcal{D}$ can be weakened by adding useless (dummy) assumptions. In this case, however, we have to require the new assumptions to be $i-j$-monovalent if Γ was so, in order to guarantee that possible applications of the $(\wedge - I)$ rule in $\mathcal{D}$ still hold.

This is summarized by the following property.

Property 1 (Weakening) *If $\Gamma \vdash \varphi$ then $\Gamma^* \vdash \varphi$ for any Γ^*, extending Γ, such that:*

i) for all φ, if φ belongs to Γ then $\varphi|\sigma_1|\cdots|\sigma_n$ belongs to Γ^* for any $\sigma_1,\ldots,\sigma_n$ $(n \geq 0)$.

ii) if Γ is $i-j$-monovalent then Γ^* is $i-j$-monovalent too.

Definition 3.7 Let Γ be a context, φ be a hyperformula and p be a permutation of $\{1,\ldots,m\}$. We define the permutation of φ by p, denoted by $p(\varphi)$, as the hyperformula ψ such that:

- $(\psi)_{p(i)} = (\varphi)_i$ $(1 \leq i \leq m)$;
- $length(\psi) = max(length(\varphi), size(p))$

where $size(p)$ is the greatest i such that $p(i) \neq i$.

We generalize the notion of permutation to contexts, in an obvious way. Thus $p(\Gamma)$ denotes the context obtained by applying p to each hyperformula in Γ.

For example, let p a permutation of $\{1,2,3\}$ such that $p(1) = 2$, $p(2) = 3$ and $p(3) = 1$, then

$$p(\sigma|\tau|\sigma'|\tau') = \sigma'|\sigma|\tau|\tau'$$

$$p(\sigma|\tau) = \varepsilon|\sigma|\tau$$

The following lemma shows how deductions running in parallel can be permuted to obtain them in a different order.

We observe that any permutation of the conclusion requires the same permutation to be applied to the assumptions that belong to the context of the deduction.

Lemma 3.8 (Commutation Property) *Let $\mathcal{D}$ be a proof of $\Gamma \vdash \varphi$ and let p be a permutation of $\{1,\ldots,m\}$ such that $m \leq length(\varphi)$. Then there exists a proof $\mathcal{D}'$ of $p(\Gamma) \vdash p(\varphi)$.*

Proof. The proof proceeds by induction on $\mathcal{D}$. The base case is trivial. In the inductive step, the only interesting cases are $(| - w)$ and $(\wedge I)$.

In the first case we have:

$$\frac{\Gamma \vdash \varphi|\sigma}{\Gamma \vdash \varphi} \quad (| - w)$$

Since $length(\varphi) \geq m$ then $length(\varphi|\sigma) \geq m$. By induction hypothesis there exists a proof of $p(\Gamma) \vdash p(\varphi|\sigma)$. Moreover p is a permutation of $\{1,\ldots,m\}$ where $m \leq length(\varphi)$, then $p(\varphi|\sigma) = p(\varphi)|\sigma$. Thus

$$\frac{p(\Gamma) \vdash p(\varphi)|\sigma}{p(\Gamma) \vdash p(\varphi)} \quad (| - w)$$

For the second case, the thesis follows from the induction hypothesis, because if Γ is $i - j-$monovalent then $p(\Gamma)$ is $p(i) - p(j)-$monovalent.

$\square$

Lemma 3.9 *If $\mathcal{D} : \Gamma, \psi \vdash \varphi$ then there exists a proof $\mathcal{D}'$ such that $\mathcal{D}' : \Gamma, \psi[\sigma \mapsto \sigma \wedge \tau] \vdash \varphi$ for any formula τ.*

Proof. By induction on $\mathcal{D}$. If $\mathcal{D} : \Gamma, \psi \vdash \psi$ by (Ax), that is $\varphi = \psi$, then construct $\mathcal{D}'$ thus:

$$\frac{\Gamma, \psi[\sigma \mapsto \sigma \wedge \tau] \vdash \psi[\sigma \mapsto \sigma \wedge \tau]}{\Gamma, \psi[\sigma \mapsto \sigma \wedge \tau] \vdash \varphi} \ (\wedge E)$$

In the inductive step, the only interesting case is when the last applied rule is $(\wedge I)$. Then the thesis follows from the induction hypothesis, because all formulae σ, such that $\sigma \in \varphi$, are simultaneously replaced by $\sigma \wedge \tau$, so than any $i - j-$monovalency of the context is preserved by the component-substitution. □

The lemma above allows to perform a more careful analysis of the relationship between hyperformulae and $\wedge-$formulae, that is stated in the following theorem.

Theorem 3.10 (From $|$ to $\wedge$) *If $\mathcal{D} : \Gamma \vdash \sigma|\tau$ then there exists a proof $\mathcal{D}'$*

$$\mathcal{D}' : \Gamma' \vdash \sigma \wedge \tau$$

where Γ' is such that, for any φ, φ belongs to Γ if and only if $\varphi[1 \mapsto (\varphi)_1 \wedge (\varphi)_2][2 \mapsto (\varphi)_1 \wedge (\varphi)_2]$ belongs to Γ'.

Proof. Construct a proof of $\Gamma' \vdash \sigma|\tau$ by using Lemma 3.9. Hence, $(\wedge I)$ applies since Γ' is $1 - 2-$monovalent, thus $\mathcal{D}' : \Gamma' \vdash \sigma \wedge \tau$. □

To sum up, we observe that assumptions are intended as packets of formulae in parallel, that are used to deduce formulae in parallel, by keeping assumed and deduced formulae in lockstep during the deduction.

Instead, we need a richer context for deducing an $\wedge-$formula. First, all the assumptions, that are used in all the conclusions, must be composed in one $\wedge-$formula. Then several copies of this $\wedge-$formula, composed by $|$ in one hyperformula, are available as assumptions for the deductions, which have to go on in a parallel way in order to fuse their conclusions at the final step. In other words, the main difference between a deduction of $\sigma|\tau$ and a deduction of $\sigma \wedge \tau$ consists in using different resources (assumptions) in a different way during the parallel deductions of σ and τ, respectively.

3.3 Hyperformulae, Hypersequents and Parallelism

The key idea of the logic HL is the notion of hyperformulae, that allows handling the (metalogic) concept of packets of parallel deductions by only using logical rules in standard natural deduction style.

This approach is closely related to the method of *Hypersequents*, introduced by Avron for representing the proof theory of non-classical logics (see [1,2]).

Hypersequents are defined as finite sequences of *Sequents* composed by a *parallel* operator. Thus they form a generalization of the sequential framework as well as the HL logic generalizes the natural-deduction framework.

However the main difference between the two approaches relies on the degree of parallelism represented by the logical rules. The interpretation of Avron's parallel constructor is disjunctive, since most of the deduction rules treat only one component of the hypersequent, in other words rules can be applied concurrently.

Instead, in the logic HL the interpretation of $|$ is strongly conjunctive.

From a computational view point, one may look at hyperformulae as compositions of processes running in parallel. In this perspective, each application of a deduction rule can be represented as a labeled event, where a label α contains both the name of the applied rule and the related argument, for instance the discharged assumption in the $(\rightarrow I)$ and the minor premise in the $(\rightarrow E)$.

Then an $\alpha-$labeled event in a process of a parallel composition must synchronize with $\alpha-$labeled events from all the other components in order to form a synchronization event labeled by α.

To sum up, in the logic HL the maximum parallelism is represented by hyperfomulae, rules cannot be applied asynchronously, and the kind of synchronization required in deductions is more close to that of CSP [8] than to the synchronization implicit in Avron's logic.

4 Labeling proofs by lambda-terms

In this section we define a labeling procedure that associates λ-terms to HL-proofs. This annotation of proofs by terms is instrumental to the isomorphism between $TA_\wedge$ and HL. Informally speaking, the first step will consist in labeling the context of a deduction by associating distinct lambda variables to hyperformulae which are assumptions. Then, once a labeled set Γ_λ of assumptions is provided, any $(\rightarrow I)$ and $(\rightarrow E)$-rule application will correspond to perform a $\lambda-$abstraction and an application on the label, respectively. Instead, local rules, that is $(\wedge I)$, $(\wedge E)$, and $(| - w)$, do not modify the $\lambda-$term decorating the proof. Finally, the $\lambda-$term associated to the whole proof will encode, by its structure, the deductive history of the proof.

Definition 4.1 Let Γ be a context. Let λ be a function associating all distinct $\lambda-$variables to the hyperformulae in Γ. Then Γ_λ is the labeled version of Γ such that, for all φ in Γ,

$$\text{if } \lambda(\varphi) = x \text{ then } x : \varphi \in \Gamma_\lambda.$$

We extend to Γ_λ all the functions and notations defined on Γ, in an obvious way.

Let us notice that $(\Gamma_\lambda)_i$ is a set while $(\Gamma)_i$ is a multiset.

Definition 4.2 Let $\mathcal{D}$ be a proof of $\Gamma \vdash \varphi$. For any given Γ_λ, the $\lambda-$term M labeling $\mathcal{D}$ is defined by induction on $\mathcal{D}$ using the following rules

Case (Ax)

$$If\ \Gamma \vdash \varphi,\ where\ \varphi \in \Gamma,\ then\ \Gamma_\lambda \vdash \lambda(\varphi) : \varphi$$

Case $(| - w)$

$$\Gamma_\lambda \vdash M : \varphi|\sigma \quad \Rightarrow \quad \Gamma_\lambda \vdash M : \varphi$$

Case $(\wedge E)$

$$\Gamma_\lambda \vdash M : \varphi,\ (\varphi)_i = \sigma \wedge \tau \ \Rightarrow \quad \Gamma_\lambda \vdash M : \varphi[i \mapsto \sigma]$$

Case $(\wedge I)$

$$\Gamma_\lambda \vdash M : \varphi \quad \Rightarrow \quad \Gamma_\lambda \vdash M : \varphi[i \mapsto (\varphi)_i \wedge (\varphi)_j][j \mapsto \varepsilon]$$

Case $(\rightarrow I)$

$$\Gamma_\lambda, x : \tau_1|\dots|\tau_n \vdash M : \sigma_1|\dots|\sigma_s,\ (s \leq n) \quad \Rightarrow \quad \Gamma_\lambda \vdash \lambda x.M : \rho_1|\dots|\rho_s$$

according to rule $(\rightarrow I)$ in Definition 3.5

Case $(\rightarrow E)$

$$\Gamma_\lambda \vdash M : \sigma_1|\dots|\sigma_n,\ \Gamma_\lambda \vdash N : \tau_1|\dots|\tau_n \quad \Rightarrow \quad \Gamma_\lambda \vdash MN : \rho_1|\dots|\rho_n$$

according to rule $(\rightarrow E)$ in Definition 3.5

We write $\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \varphi$ to denote the labeling, by the $\lambda-$term M, of $\mathcal{D}$ for a given Γ_λ and we say that $\mathcal{D}_\lambda$ is M-labeled.

Notice that, for different Γ_λ, different $\lambda-$terms can be associated to a deduction $\mathcal{D}$. However all these terms are equal on their structure, they only differ in names of free variables. In fact, the structure of the proof strictly corresponds to the structure of the associated $\lambda-$term, but for rules $(| - w)$, $(\wedge E)$ and $(\wedge I)$.

Example 4.3 Let $\Gamma_\lambda = \{x : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3)|(\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3),\ y : \sigma_1|\sigma_1\}$ then labeling of the proof in Example 3.6 is the following.

$$\dfrac{\dfrac{\dfrac{\Gamma_\lambda \vdash y : \sigma_1|\sigma_1 \qquad \dfrac{\dfrac{\Gamma_\lambda \vdash x : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3)|(\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3)}{\Gamma_\lambda \vdash x : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3)|\sigma_1 \rightarrow \sigma_3}(\wedge E)}{\Gamma_\lambda \vdash x : \sigma_1 \rightarrow \sigma_2|\sigma_1 \rightarrow \sigma_3}(\wedge E)}{\dfrac{\Gamma \vdash xy : \sigma_2|\sigma_3}{\Gamma \vdash xy : \sigma_2 \wedge \sigma_3|\varepsilon}(\wedge I)}(\rightarrow E)}{\dfrac{x : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3)|(\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3) \vdash \lambda y.xy : \sigma_1 \rightarrow \sigma_2 \wedge \sigma_3|\varepsilon}{\vdash \lambda x.\lambda y.xy : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3) \rightarrow \sigma_1 \rightarrow \sigma_2 \wedge \sigma_3|\varepsilon}(\rightarrow I)}(\rightarrow I)}{\vdash \lambda x.\lambda y.xy : (\sigma_1 \rightarrow \sigma_2) \wedge (\sigma_1 \rightarrow \sigma_3) \rightarrow \sigma_1 \rightarrow \sigma_2 \wedge \sigma_3}(| - w)$$

4.1 Properties of labeled proofs

We first reformulate Lemma 3.8 and Theorem 3.10 for labeled proofs in an obvious way. Then, we prove the *Composition Lemma*, which is the main result of the present section.

Lemma 4.4 (Commutation Property for labeled proofs) *Let $\mathcal{D}_\lambda$ be an M-labeled proof of $\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \varphi$, and let p be a permutation of $\{1, \ldots, m\}$, where $lenght(\varphi) \geq m$. Then there exists an M-labeled proof $\mathcal{D}'_\lambda$ of $p(\Gamma'_\lambda) \vdash M : p(\varphi)$.*

Proof. The proof is the same as in Lemma 3.8, but using labels as in Definition 4.2. □

Lemma 4.5 (From | to ∧ for labeled proofs) *If $\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \sigma|\tau$ then there exists an $M-$labeled proof $\mathcal{D}'$ such that $\mathcal{D}'_\lambda : \Gamma'_\lambda \vdash M : \sigma \wedge \tau$, where Γ'_λ is such that $x : \varphi$ belongs to Γ_λ if and only if $x : \varphi[1 \mapsto (\varphi)_1 \wedge (\varphi)_2][2 \mapsto (\varphi)_1 \wedge (\varphi)_2]$ belongs to Γ'_λ.*

Proof. The proof is the same as in Theorem 3.10, but using labels as in Definition 4.2. □

We define now how to construct a labeled context by a parallel mixing of two contexts.

Definition 4.6 Let Γ'_λ and Γ''_λ be two labeled contexts, then the context $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$ $(n, m \geq 1)$ is defined in the following way:

- if $x : \varphi$ belongs to Γ'_λ and $x : \psi$ belongs to Γ''_λ, $length(\varphi) = \ell$ and $length(\psi) = k$, then

$$x : (\varphi)_1|\cdots|(\varphi)_n|(\psi)_1|\cdots|(\psi)_m|(\varphi)_{n+1}|\cdots|(\varphi)_\ell|(\psi)_{m+1}|\cdots(\varphi)_k$$

  belongs to $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$
- if $x : \varphi$ belongs to Γ'_λ and no x labeled assumption belongs to Γ''_λ, then

(i) if $length(\varphi) \leq n$ then $x : \varphi$ belongs to $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$

(ii) if $n < length(\varphi)$ then

$$x : (\varphi)_1|\cdots|(\varphi)_n| \underbrace{\varepsilon|\cdots|\varepsilon}_{\text{m times}} |(\varphi)_{n+1}|\cdots|(\varphi)_{length(\varphi)}$$

  belongs to $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$
- if $x : \psi$ belongs to Γ''_λ, where $length(\psi) = k$, and no x labeled assumption belongs to Γ'_λ, then

$$x : \underbrace{\varepsilon|\cdots|\varepsilon}_{\text{n times}} |\psi$$

  belongs to $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$
- nothing else belongs to $\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$

We point out that any context $\Gamma_\lambda \|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$ is built up in such a way that, for any i and j,

- if Γ'_λ is $i - j-$monovalent $(i, j \leq n)$ then Γ_λ is $i - j-$monovalent;

- if Γ''_λ is $i-j-$monovalent $(i,j \leq m)$ then Γ_λ is $(n+i)-(n+j)-$monovalent.

This is a crucial property in the proof of the next lemma.

Lemma 4.7 (Composition Lemma) *Let $\mathcal{D}'_\lambda$ and $\mathcal{D}''_\lambda$ be two M-labeled proofs of $\Gamma'_\lambda \vdash M : \sigma_1|\ldots|\sigma_n$ and $\Gamma''_\lambda \vdash M : \tau_1|\ldots|\tau_m$ respectively. Then there exists an M-labeled proof*

$$\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n|\tau_1|\ldots|\tau_m$$

such that $\Gamma_\lambda = \|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$

Proof. By induction on the sum of the depths of $\mathcal{D}'_\lambda$ and $\mathcal{D}''_\lambda$.

- If $\mathcal{D}' : \Gamma'_\lambda, x : \sigma_1|\ldots|\sigma_n \vdash \sigma_1|\ldots|\sigma_n$ and $\mathcal{D}'' : \Gamma''_\lambda, x : \tau_1|\ldots|\tau_m \vdash \tau_1|\ldots|\tau_m$ then
$$x : \sigma_1|\ldots|\sigma_n|\tau_1|\ldots|\tau_m \in \Gamma_\lambda$$
by definition of Γ_λ.
- Let $(| - w)$ be the last rule in $\mathcal{D}'_\lambda$ or in $\mathcal{D}''_\lambda$, for example:
$$\mathcal{D}'_\lambda : \frac{\Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n|\sigma}{\Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n} \; | - w$$
By the induction hypothesis there exists a proof of
$$\|_{n+1,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle \vdash M : \sigma_1|\ldots|\sigma_n|\sigma|\tau_1|\ldots|\tau_m$$
Let p be a permutation of $\{1,\ldots,m+n+1\}$ such that:
· $p(i) = i$ $(1 \leq i \leq n)$;
· $p(n+1) = n+m+1$;
· $p(n+1+i) = n+i$ $(1 \leq i \leq m)$;.
Using the Commutation Property for labeled proofs we obtain a proof of
$$p(\|_{n+1,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle) \vdash M : \sigma_1|\ldots|\sigma_n|\tau_1|\ldots|\tau_m|\sigma$$
It is easy to verify that
$$p(\|_{n+1,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle) = \|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle$$
Then from the proof of
$$\|_{n,m} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle \vdash M : \sigma_1|\ldots|\sigma_n|\tau_1|\ldots|\tau_m|\sigma$$
we obtain $\mathcal{D}_\lambda$ by applying the $(| - w)$-rule.
- Let $(\wedge E)$ be the last rule in $\mathcal{D}'_\lambda$ or $\mathcal{D}''_\lambda$, for example
$$\mathcal{D}'_\lambda : \frac{\Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n \quad \sigma_i = \sigma \wedge \tau}{\Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n[i \mapsto \sigma]} \; \wedge E.$$
By induction hypothesis there is a proof of $\Gamma_\lambda \vdash M : \sigma_1|\ldots|\sigma_n|\tau_1|\ldots|\tau_m$, then the $(\wedge E)-$rule can be applied on the $i-$th component, and the thesis hold.

- For $(\wedge I)$ we proceed as for $(\wedge E)$. First two proofs are mixed, then $(\wedge I)$ rule is applied.
- Let $\to E$ be the last applied rule in $\mathcal{D}'_\lambda$ and $\mathcal{D}''_\lambda$, that is

$$\mathcal{D}'_\lambda : \frac{\Gamma'_\lambda \vdash M : \sigma_1|\dots|\sigma_n \qquad \Gamma'_\lambda \vdash N : \tau_1|\dots|\tau_n}{\Gamma'_\lambda \vdash MN : \rho_1|\dots\ |\rho_n} (\to E)$$

$$\mathcal{D}''_\lambda : \frac{\Gamma''_\lambda \vdash M : \sigma'_1|\dots|\sigma'_m \qquad \Gamma''_\lambda \vdash N : \tau'_1|\dots|\tau'_m}{\Gamma''_\lambda \vdash MN : \rho'_1|\dots\ |\rho'_m} (\to E)$$

By using induction hypothesis we construct the proof $\mathcal{D}_\lambda$ as

$$\mathcal{D}_\lambda : \frac{\Gamma_\lambda \vdash M : \sigma_1|\dots\ |\sigma_n|\sigma'_1|\dots|\sigma'_m \quad \Gamma_\lambda \vdash N : \tau_1|\dots|\tau_n|\tau'_1|\dots|\tau'_m}{\Gamma_\lambda \vdash MN : \rho_1|\dots|\rho_n|\rho'_1|\dots|\rho'_m} \to E.$$

- If $(\to I)$ is the last applied rule then we proceed as for $(\to E)$.
- No other cases are possible.

□

5 The Curry-Howard isomorphism between HL and $TA_\wedge$

Now we can prove that HL is isomorphic to $TA_\wedge$, which is the main goal of this paper, using the Curry-Howard approach. The idea of this isomorphism is thus to interpret logical deductions as type-derivations and conversely, through the labeled proofs defined in the previous section.

A first step toward this result is to define an interpretation of formulae and hyperformulae as types. Formulae of Definition 3.1 can be read as types, mapping propositional variables, $\to$ and $\wedge$ into type variables, arrow and intersection on types.

Moreover, we read hyperformulae as types, by mapping the parallel constructor $|$ into the $\wedge-$type constructor.

We will write $\overline{\varphi}$ to denote the type so associated to φ; for instance $\overline{\sigma|\tau \wedge \rho} = \sigma\wedge\tau\wedge\rho$. Since ε denotes the empty formula, $\overline{\varphi|\varepsilon} = \overline{\varphi}$. Extending this mapping to contexts (labeled contexts) we write $\overline{\Gamma}$ $(\overline{\Gamma_\lambda})$ for denoting the set of types $\overline{\Gamma} = \{\overline{\varphi}|\varphi \in \Gamma\}$ $(\overline{\Gamma_\lambda} = \{x : \overline{\varphi}|x : \varphi \in \Gamma_\lambda\})$.

Let us notice that, if $\varphi = \sigma|\tau$ and $\psi = \sigma \wedge \tau$ then $\overline{\varphi} = \overline{\psi}$. Analogously, if $\Gamma'_\lambda = \{x : \sigma|\tau\}$ and $\Gamma''_\lambda = \{x : \sigma \wedge \tau\}$, then $\overline{\Gamma'_\lambda} = \overline{\Gamma''_\lambda} = \{x : \sigma \wedge \tau\}$

This is not surprising, because the treatment of the intersection in $TA_\wedge$ as a proof-functional type constructor flattens too many aspects in the $(\wedge I)-$ typing rule.

Completely different, our quest for a logical system interpreting the $\wedge-$ derivability, without any metalinguistic constraint in the deduction rules, required using two logical operator, namely $|$ and $\wedge$. Therefore two assumptions of the form $\sigma|\tau$ and $\sigma \wedge \tau$, even if interpreted as the same type $\sigma \wedge \tau$, give raise to different proofs in HL.

Theorem 5.1 (Types as proofs) *If $\Delta : B \vdash_\wedge M : \sigma$ then there exists an*

$M-$labeled proof $\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \sigma$ such that $\overline{\Gamma_\lambda} = B$.

Proof. By induction on Δ.

- If $\Delta : B, x : \sigma \vdash_\wedge x : \sigma$ then we construct the $x-$labeled proof $\mathcal{D}_\lambda$ as:

$$\mathcal{D}_\lambda : \Gamma_\lambda, x : \sigma \vdash x : \sigma$$

  where λ is the mapping defined by $B \cup \{x : \sigma\}$.
- If the last applied rule is $(\wedge I)$, i.e.

$$\frac{B \vdash_\wedge M : \sigma \quad B \vdash_\wedge M : \tau}{B \vdash_\wedge M : \sigma \wedge \tau} \ (\wedge I)$$

  then, by induction hypothesis, there are two $M-$labeled proofs $\mathcal{D}'_\lambda : \Gamma'_\lambda \vdash M : \sigma$ and $\mathcal{D}''_\lambda : \Gamma''_\lambda \vdash M : \sigma$. Since both $\mathcal{D}'_\lambda$ and $\mathcal{D}''_\lambda$ are $M-$labeled, then we apply Composition-Lemma and we obtain an $M-$labeled proof of

$$\|_{1,2} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle \vdash M : \sigma | \tau$$

  where $\overline{\|_{1,2} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle} = B$, by construction. Finally $\Gamma_\lambda \vdash M : \sigma \wedge \tau$ follows from Lemma 4.5 where $\overline{\Gamma_\lambda} = \overline{\|_{1,2} \langle \Gamma'_\lambda, \Gamma''_\lambda \rangle} = B$, since the parallel operator $|$ is mapped into $\wedge$.

All the other cases, concerning $(\wedge E)$, $(\rightarrow I)$ and $(\rightarrow E)$, directly follow from the induction hypothesis.

□

Lemma 5.2 *If $\mathcal{D}_\lambda$ is an $M-$labeled proof*

$$\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \varphi$$

then, for every i such that $(\varphi)_i \neq \varepsilon$, $B_i \vdash_\wedge M : (\varphi)_i$, where $B_i = (\Gamma_\lambda)_i$ $(1 \leq i \leq length(\varphi))$.

Proof. By induction on $\mathcal{D}_\lambda$. The base case is straightforward. In the inductive step, the cases $(\wedge E)$ and $(|w)$ follow from the induction hypothesis.

Let us consider the case when the last applied rule is the $(\rightarrow E)$. Then, by induction hypothesis, $B_i \vdash_\wedge M : \tau_i \rightarrow \sigma_i$ and $B_i \vdash_\wedge N : \tau_i$, where $B_i = (\Gamma_\lambda)_i$. Hence by using the typing rule $(\rightarrow E)$ we obtain the thesis $B_i \vdash_\wedge MN : \sigma_i$.

The proof is similar when the last applied rule is $(\rightarrow I)$ or $(\wedge I)$.

□

Theorem 5.3 (Proofs to types) *If $\mathcal{D}_\lambda : \Gamma_\lambda \vdash M : \varphi$ then there exists a type derivation*

$$\Delta : B \vdash_\wedge M : \overline{\varphi}$$

such that $B = \overline{\Gamma_\lambda}$.

Proof. By Lemma 5.2, $B_i \vdash_\wedge M : (\varphi)_i$, for every i such that $(\varphi)_i \neq \varepsilon$, where $B_i = \overline{(\Gamma_\lambda)_i}$ $(1 \leq i \leq length(\varphi))$. Then construct the basis $B = \bigcup B_i$ as

$$\bigcup B_i = \{x : \tau_1 \wedge \cdots \wedge \tau_n | x : \tau_i \in B_i\}$$

It is trivial to verify that $B \vdash_\wedge M : (\varphi)_i$ and $B = \overline{\Gamma_\lambda}$, hence $B \vdash_\wedge M : \overline{\varphi}$ by using the $(\wedge I)-$rule of $TA_\wedge$.

□

Theorem 5.4 **($TA_\wedge$ and HL are isomorphic)** $\Delta : B \vdash_\wedge M : \sigma$ *if and only if* $\mathcal{D} : \Gamma \vdash \varphi$ *where* $\overline{\varphi} = \sigma$ *and* $B = \{x : \sigma | \sigma \in \overline{\Gamma}\}$.

Proof. By Theorems 5.1 and Theorem 5.3. □

6 Conclusion and Future work

We presented a propositional logical calculus which formalizes a (metalogic) notion of parallel deductions by using hyperformulae as parallel compositions of formulae.

The main feature of this logic is that deduction rules are in the standard shape of the natural deduction framework and do not involve any intensional requirement on the subproofs. We showed that this calculus is isomorphic to the intersection type assignment for lambda calculus, reading formulae and hyperformulae as types by the well known Curry-Howard paradigm. Namely, every deduction is associated to a type inference for a lambda term, where the term encodes the "history" of the deduction.

As a result, the intersection type constructor , which comes out to be proof-functional in the type theory, is interpreted as a standard propositional connective.

The most interesting application of the logic we proposed here should be the definition of an explicitly typed lambda calculus with intersection types. This would be a very powerful functional language, in which all (and only) strongly normalizing terms-programs have a well-typed version. It is in this context that we are currently investigating such a typed lambda calculus. Deduction rules of HL and labeled proofs suggest a smooth solution to this further issue.

Acknowledgments

We would like to thank a group of Ph.D. students in Siena, C. Bianchini, D. Luchi and S. Sadocco, for many helpful discussions and ideas that originated the notion of parallel deductions presented in this paper. We are grateful to F.Barbanera and M.Dezani for useful suggestions.

References

[1] A. Avron. Hypersequents, logical consequence and intermediate logics for concurrency. *Annals of Mathematics and Artificial Intelligence*, 4(3–4):225–248, 1991.

[2] A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn, and John Truss, editors, *Logic: from foundations to applications. Proc. Logic Colloquium, Keele, UK, 1993*, pages 1–32. Oxford University Press, New York, 1996.

[3] H. P. Barendregt. *The Lambda-Calculus: Its Syntax and Semantics.* North-Holland, Amsterdam, 1980.

[4] H. P. Barendregt. Lambda calculi with types. In D. M. Gabbai Samson Abramski and T. S. E. Maiboum, editors, *Handbook of Logic in Computer Science.* Oxford University Press, Oxford, 1992.

[5] Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *The Journal of Symbolic Logic*, 48(4):931–940, December 1983.

[6] M. Coppo and M. Dezani-Ciancaglini. A new type assignment for lambda-terms. *Archive f. math. Logic u. Grundlagenforschung*, 19:139–156, 1979.

[7] M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Functional characters of solvable terms. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 27:45–58, 1981.

[8] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs (NJ), USA, 1985.

[9] B. Venneri. Intersection types as logical formulae. *Journal of Logic and Computation*, 4(2):109–124, April 1994.