Hyperintensions for Probabilistic Computations

Giuseppe Primiero
Logic, Uncertainty, Computation and Information Group
University of Milan, Italy
giuseppe.primiero@unimi.it

Abstract

HTLC, for Hyperintensional Typed Lambda Calculus, is an extension of the typed λ-calculus with hyperintensions and related rules, introduced in [11]. This contribution introduces HTLC$_p$, which adapts the former to hyperintensions for non-deterministic processes. We formulate appropriate conditions for reasoning with such objects, discuss its metatheory and conclude with some observation comparing it with the semantics of Transparent Intensional Logic.

1 Introduction

Hyperintensions, as concepts which distinguish between necessarily equivalent contents, are enjoying a large philosophical success. Since their introduction in [3], a number of notions have been identified as hyperintensional up to some extent: information, belief, knowledge, meaning, explanation, to name just a few of the most common ones. This success does not extend to the representation of computations in the physical and digital domains, where hyperintensions are still little

This research has been funded by the Project PRIN2020 BRIO - Bias, Risk and Opacity in AI (2020SSKZ7R) awarded by the Italian Ministry of University and Research (MUR).

explored, possibly with the exception of computational semantics for natural language [15].

A formal model of hyperintensions for the typed λ-calculus HTLC is introduced in [11]. The system is designed to reason with expressions for extensional, intensional and hyperintensional entities. Under the Curry-Howard isomorphism, expressions of HTLC are interpreted computationally, with a term t a computational process (e.g. the application of the successor function to define a natural number) and its type T its output (type $\mathbb{N}$ or value 4); an hyperintensional term t^* of type $*^T$ can be seen as denoting computational processes t, t' which are identical with respect to their outputs T, but for which structural identity fails (e.g. t might denote the successor function application, while t' might denote composition of the successor function and the summation function). Hence, distinct notations or modes of presentation or execution for the computation of a given natural number, or distinct operations to compute a given value of any other type, would be denoted by the same hyperintensional term.

HTLC offers an alternative, proof-theoretic view to TIL hyperintensions [8], the latter being especially suitable for modelling cases from natural language [10] because of its ability to express partial functions, the former being constrained to total functions. A recent conjecture presented in [7] states that the non-hyperintensional fragment of total functions, without modalities (quantifying over possible worlds and times) of TIL is Henkin complete. For HTLC with only total functions and proper constructions (no modalities quantifying over possible worlds and times are available), we have shown that any consistent set of closed well-formed formulas Λ is satisfiable in a model in which the domain of basic and function types is denumerable, and the domain for hyperintensional types is non-denumerable but strongly reducible to a model with a denumerable domain for extensional and intensional types. There is no form of partiality or non-determinism in HTLC.

While degrees of beliefs have long been interpreted in the form of subjective probabilities [6, 13], the link between probabilities and hyperintensions is little explored. In fact, in the domain of computations with (sharp) probabilistic outputs, hyperintensional terms and associated attitudes (such as belief and trust) become extremely natural. Standardly, types are interpreted as propositions or sets, their terms being respectively proofs and elements of the set. In the case of terms as computational processes, types are normally interpreted as their output types, e.g. the result of an addition process on natural numbers will be a natural number, thereby expressing the signature of the computation $\mathbb{N} \to \mathbb{N}$. An alternative, more characteristic interpretation is to look at input and output values respectively, so that the computational terms are natural numbers and the type is their output value. For simplicity, we focus on the latter. Let us consider an example:

> *With a fair die, the theoretical probability to obtain a* 4 *is* 0.16666666666*.*

One way to express this more precisely is to say: under the probability distribution corresponding to a fair die, throwing the die will produce 4 with a theoretical probability of 0.16666666666. Note that here the process of throwing the die can be seen as a computational process with 4 as its output. It is now quite natural to see computations with probabilistic outputs as possible denotations of hyperintensional terms:

1. the claim that

   the theoretical probability to obtain 4 *is* 0.16666666666

   can be seen as the subjective probability 1 measuring the degree of belief that the die is fair at least with respect to output 4;

2. the claim assigns a probability value valid under a number of distinct initial probability distributions expressing either a fair or a biased die; hence, an assertion of identity valid for two processes observed for a given probabilistic output, may fail for different outputs; in other words, one may believe that "the theoretical probability to obtain a 4 is 0.16666666666", without necessarily believing to be using a fair die.

To see this in more detail, consider rolling two dice, where d_1 is fair, and d_2 is biased (although we might not know that): d_1 (theoretically) shows a probability 0.16666666666 of outputting 4; d_2, may be biased in a way that the probability of outputting 4 is still the same, while the probability of outputting 5 is 0.2 and the probability of outputting any of $1, 2, 3, 6$ is 0.158333333335. As their actual probability distribution is unknown, the two dice represent two systems which are behaviourally identical with respect to a given output, but structurally different otherwise. As we might not know what their actual structure is, biased or fair, they are epistemically opaque in the sense of an explanation of their working not being transparently accessible. An abstract procedure denoting the throw of a die with a probability 0.16666666666 of outputting 4 maybe actually refer to both d_1 and d_2.

In [4], further extended in [5], a natural deduction system for probabilistic computations is offered, with the above sentence formalised as follows:

$$\{x_d{:}1_{1/6}, x_d{:}2_{1/6}, x_d{:}3_{1/6}, x_d{:}4_{1/6}, x_d{:}5_{1/6}, x_d{:}6_{1/6}\} \vdash d{:}4_{0.16666666666}$$

This formula has: on the left-hand side of the derivability sign a probability distribution in which a random variable x associated with the process d receives uniformly probability $1/6$ for each of its possible outputs $1, \ldots, 6$; under such distribution, one derives on the right-hand side the expected probability of 0.16666666666 to get output 4 from process d. A major task in [5] is to allow reasoning under an unknown initial distribution to infer trustworthiness of computations by comparing the frequency of a given observed output with its expected probability. Evaluating trustworthiness clearly seems a natural task for a language like HTLC which includes hyperintensional terms denoting processes which are identical for some output but are structurally distinct, as it is the case for probabilistic computations.

In this paper we sketch the extension of HTLC to probabilistic outputs in a language dubbed HTLC$_p$, identifying appropriate conditions for reasoning from probabilistic computations to their hyperintensional counteparts and back. We offer an overview of meta-theoretical results, and conclude with some comparison with the notion of semantic computation holding for Transparent Intensional Logic.

2 The Logic HTLC$_p$

2.1 Language

The syntax of HTLC$_p$ is a typed λ-calculus extended with probabilistic types and a type for hyperintensional terms denoting them:

Definition 1 (Grammar).

$$T_p ::= T_p \mid (T_p\ T^1_{p_1} \dots T^n_{p_n}) \mid (T + T')_p \mid *^{T_p}$$

$$t ::= x_t \mid [\lambda x_{t_1} \dots x_{t_n}.t] \mid [t\ t_1 \dots t_n] \mid t^*$$

In the present context, we are especially interested in an interpretation of types following the Curry-Howard-De Bruijn isomorphism: terms are computations and types are their (probabilistic) outputs. As mentioned above, probabilitic outputs can be seen both as types of outputs and as output values, and we focus here on the latter interpretation. Hence the first significant difference from HTLC in this grammar is the addition of a probabilistic index p to types.

The type syntax includes extensional entities T_p. Among these we might include probabilistic truth values, individuals with a certain probability to belong to a given universe, but also output values in those domains, e.g. a random number generator and its output 4 in the domain $\mathbb{N}$ with a given probability to be produced. The latter will be the most natural interpretation for HTLC$_p$.

Next, we add intensional entities with a given probability p of outputting type T when the probabilities of their input type is respectively p_1 for T_1 up to p_n for T_n). While in HTLC these are constrained to total functions, by having functions with probabilistic arguments $T^1_{p_1} \dots T^n_{p_n}$ we obtain an output T_p, with $p = \Pi(p_1 \cdot \dots \cdot p_n)$ which implements a form of indeterminacy. We simplify the arrow notation of multi-argument functions $(T_1 \to \cdots \to T_n \to T)$ with the pair notation $(((T\ T_n)\dots)\,T_1\,)$. As in standard typed λ-calculi we use association to the left when dealing with function types, so the curried version can be rewritten as $(T\ T_n\ \dots\ T_1)$. We can build higher-order functions that take functions as arguments and return a function as value. Specific to this calculus is the case where all the input types to a function express the probability distribution of outputs under which a given one can be obtained: $(\lambda x)t : (T_p\ T'_{1-p})$ is thus a specific form of a function type term in which the probability p of t to get output T depends on the probability $1-p$ of t to get output T'. Returning to our example from the previous section, consider the expression $d : (4_{1/6}\ 1_{1/6}\dots 6_{1/6})$, which expresses the function type denoting a fair die such that if there is a probability 1/6 to get any of its possible outputs, then there is a probability 1/6 to get output 4.

Moreover, and precisely to obtain such functional interpretation, we add a disjunction type, through which we can express exclusive probabilistic outputs by the same computation, summing up to probability $p = 1$. Hence, for our fair die we can induce the expression $\vdash d : (1+\cdots+6)_1$ saying that the probability of getting one of outputs $1\dots 6$ from die d is 1.

Finally, we include hyperintensional entities of type $*^{T_p}$, where a term of this type denotes an entity with a given probability p of having an extensional or intensional type T. Hyperintensional types are thus interpreted as procedures denoting computations with a given probabilistic output. We explain this further below referring to a term of such type.

Terms of the language are thus computations. As in HTLC we include variables, abstraction terms denoting functions and application terms denoting values of functions on given arguments. Here another notable difference from HTLC is that variables get indexed with the term (or computation) for which they act as a random variable, asserting the probability of its output. Hence, the list of variables on the left-hand side of an expression provides the probability distribution for a term on its right-hand side. To follow up on the example above,

$$\{x_d : 1_{1/6}, \dots, x_d : 6_{1/6}\} \vdash d : 4_{1/6}$$

denotes the expression that states that under the probability distribution of a fair die d, there is a probability 1/6 to have output 4.

Hyperintensional terms denote abstract procedures for computations of any of the above forms, with a given probability for the corresponding output: the two procedures denoted by the same hyperintensional term have necessarily identical probabilities for some of their value (and in turn necessarily identical beliefs are attributed to them for those values), while they might differ for probabilities assigned to other output values (and in turn different beliefs are attributed to them for those values). This is possible for example because there is no explicit formulation of the internal structure of such computations, or such structure is not accessible (hence, technically they might be black boxes, opaque in the sense used above). For example, the intuition is that an expression $t^* : *^{4_{0.16666666666}}$ denotes any process, including any rolling die (fair or not), which may return 4 with probability 0.16666666666. Currently, we do not distinguish in the syntax the theoretical probability of the fair die from the expected probability or frequency of a die observed rolling (something we explicitly do in TPTND), but will allow ourselves to consider such distinctions informally.

To sum up, a judgement of HTLC$_p$ is a formula of the form

$$\{x_1 : T^1_{p_1}, \dots, x_n : T^n_{p_n}\} \vdash t : T_p$$

saying that a computation t outputs T with probability p, provided that x_1 outputs T^1 with probability p_1, up to x_n outputs T^n with probability p_n.

2.2 Rules

Inference rules for computational terms of arbitrary types are adapted from [5] and further extended with rules to define the meaning of hyperintensional terms denoting probabilistic computations, see Figure 1. Note that in this preliminary formulation we merge what in TPTND [5] is strictly separated, namely reasoning about random variables, single experiments and rules for sampling.

The Assumption rule allows the derivation of a typed random variable from its own assumption. The Abstraction rule allows to construct a λ-term for a function type that takes inputs $T^1, \dots, T^n$ respectively with probabilities $p_1, \dots, p_n$ (possibly with more assumptions encoded in Γ), returns output T with probability p. The Application rule creates a term of type T denoting the value of a function, expressed by a computation t on the n-tuple of arguments expressed by computational terms $t_1, \dots, t_n$, where the probability of T is computed as the product of the probabilities of all its inputs. We further formulate rules for the additivity of exclusive probabilistic outputs. Given a probability distribution encoded in Γ and process t giving output T with probability p and output T' with probability p', process t will give either output with a probability $p + p'$, with such probabilities adding up to 1. Taken the disjunction of all possible outputs $T + T'$ of process t and the probability p of one such output T, the probability of the remaining output T' will be $1 - p$.

Next we add rules for hyperintensional terms. The Trivialization rule defines the process of going from a given computation t which outputs T with probability p to the hyperintensional term t^* which denotes a construction of the computation denoted by t, i.e. t^* denotes any of the computations with the same output as t. When the trivialized computation t is of type T_p, Trivialization allows to shift from a an extensional computation to a hyperintensional one producing it. When the trivialized computation t is of type $(T_p\ T_{p_1} \dots T_{p_n})$, Trivialization allows to shift from a functional computation to a hyperintensional one producing it. We do not consider higher-order triviliased terms here. The criterion for selecting a procedure t^* denoting any computation with output T_p is fidelity, which can be evaluated by the corresponding elimination rule. Execution proceeds from a hyperintensional type $*^{T_p}$ to its intended denoted extensional or intensional computational term with output T with probability p. In order to do so, i.e. to identify a computation t' that satisfies the hyperintensionally denoted output T_p, we exploit the trustworthiness criteria defined for TPTND in [5]. Given a procedure t^* of type $*^{T_p}$, Execution returns a computation t' denoting its output, that is a term of type T with probability p', provided the absolute difference between the intended and the observed probability of the outputs is below a given critical threshold, typically the confidence interval for the intended probability parametrized over the number n of times one has observed t' outputting T over the overall number of occurrences of t' with any other output. Similarly so for a term t^* of type $*^{(T\ T_1 \dots T_n)}$, where Execution returns the functional computation t' that denotes an intensional entity of type $(T_p\ T^1_{p_1} \dots T^n_{p_n})$.

Note that Execution requires canonical terms, i.e. terminating computations in normal form by β-reduction rules, see Figure 2. In these evaluation rules, we state: evaluation by argument substitution in abstraction; evaluation to equivalent abstracted terms; evaluation to equivalent applied terms; evaluation to equivalent arguments in application. We use $\rightarrow_\beta$ for these evaluation steps. Further, we extend reduction rules with one specific instance to account for the reduction of terms with one equivalent among exclusive distinct outputs, see Figure 3. We use $\rightarrow_p$ for this evaluation step. We use simply $\rightarrow$ for the common evaluation relation and $\twoheadrightarrow$ for its transitive and reflexive closure. The latter evaluation rule allows us to establish two important variations from HTLC. First: subject reduction is in a weaker form as it allows some form of subtyping by establishing that if t' is a term obtained by an evaluation rule (including $\rightarrow_p$) from term t, its type T' must be included in the type T of the latter, up to identity, see Figure 4. Second: term identity is defined in a stronger form, under reduction for all possible output types, see Figure 5.

Coming back to our example, consider two expressions: $\Gamma \vdash d_1 : 4_{0.16666666666}$ and $\Delta \vdash d_2 : 4_{0.16666666666}$ denote respectively that the first and the second die show a behaviourally identical probability to output 4 (assuming here the same number of throws), under the (possibly unknown) distribution of their outputs encoded respectively in Γ and Δ. Applying Trivialization to each, will return the same term

$$\frac{}{x_t : T_p \vdash x_t : T_p}\ \text{Assumption}$$

$$\frac{\Gamma, x_1 : T^1_{p_1}, \ldots, x_n : T^n_{p_n} \vdash t : T_p}{\Gamma \vdash [\lambda x_1 \ldots x_n .t] : (T_p\ T^1_{p_1}, \ldots, T^n_{p_n})}\ \text{Abstraction}$$

$$\frac{\Gamma \vdash t : (T_p\ T^1_{p_1}, \ldots, T^n_{p_n}) \qquad \Gamma_1 \vdash t_1 : T^1_{p_1} \ldots \Gamma_n \vdash t_n : T^n_{p_n}}{\Gamma, \Gamma_1, \ldots \Gamma_n \vdash [t\ t_1 \ldots t_n] : T_{p=p_1 \cdot \ldots \cdot p_n}}\ \text{Application}$$

$$\frac{x_t : T_p, x_t : T'_{p'}, \vdash t : T_p \qquad x_t : T_p, x_t : T'_{p'}, \vdash t : T'_{p'}}{x_t : T_p, x_t : T'_{p'}, \vdash t : (T + T')_{p+p' \leq 1}}\ \text{+-I}$$

$$\frac{x_t : T_p, x_t : T'_{p'}, \vdash t : (T + T')_1 \qquad x_t : T_p, x_t : T'_{p'}, \vdash t : T_p}{x_t : T_p, x_t : T'_{p'}, \vdash t : T'_{1-p}}\ \text{+-E}$$

$$\frac{\Gamma \vdash t : T_p}{\Gamma \vdash\ t^* : *^{T_p}}\ \text{Trivialization}$$

$$\frac{\Gamma \vdash t^* : *^{T_p} \qquad \Delta \vdash t' : T_{p'} \qquad \mid p - p' \mid \leq \epsilon(n)}{\Gamma, \Delta \vdash t' : T_{p'}}\ \text{Execution}$$

Figure 1: HTLC$_p$ Rules System

$t^* : *^{4_{0.16666666666}}$. Hence, given a term $t^* : *^{4_{0.16666666666}}$, Execution would return either d_1 or d_2. But assuming d_1 fair and d_2 biased, e.g. such that the probability of outputting 4 is still the same, while the probabilities of outputting 5 is 0.2 and the probability of outputting any of $1, 2, 3, 6$ is 0.158333333335, the two dice are identical with respect to a given output (namely 4), but structurally different otherwise. Hence, Subject Reduction will be satisfied for at least one evaluation (namely of the term type with output $4_{0.16666666666}$), but not for all such outputs, making term identity fail. The desirable property to infer the identity of the two dice with respect to output

$$\Gamma \vdash [[\lambda x_1 \dots x_n .t]\ t_1 \dots t_n] \rightarrow_\beta t[x_1/t_1 \dots x_n/t_n] : T_p$$

$$\frac{\Gamma, x_1 : T^1_{p_1}, \dots, x_n : T^n_{p_n} \vdash t \rightarrow_\beta t' : T_p}{\Gamma \vdash [\lambda x_1 \dots x_n .t] \rightarrow_\beta [\lambda x_1 \dots x_n .t'] : (T_p\ T^1_{p_1} \dots T^n_{p_n})}\ \beta\text{-Abstr}$$

$$\frac{\Gamma \vdash t \rightarrow_\beta t' : (T_p\ T^1_{p_1} \dots T^n_{p_n}) \qquad \Gamma_1 \vdash t_1 : T^1_{p_1} \dots \Gamma_n \vdash t_n : T^n_{p_n}}{\Gamma, \Gamma_1 \dots \Gamma_n \vdash [t\ t_1 \dots t_n] \rightarrow_\beta [t'\ t_1 \dots t_n] : T_{p=p_1 \cdot \dots \cdot p_n}}\ \beta\text{-App}$$

$$\frac{\Gamma \vdash t : (T_p\ T^1_{p_1} \dots T^i_{p_i} \dots T^n_{p_n}) \qquad \Gamma_1 \vdash t_1 : T^1_{p_1} \dots \Gamma_n \vdash t_n : T^n_{p_n} \qquad \Gamma_i \vdash t_i \rightarrow_\beta t'_i : T^i_{p_i}}{\Gamma, \Gamma_1 \dots \Gamma_n \vdash [t\ t_1 \dots t_i \dots t_n] \rightarrow_\beta [t\ t_1 \dots t'_i \dots t_n] : T_{p=p_1 \cdot \dots \cdot p'_i \cdots p_n}}\ \beta\text{-App}$$

Figure 2: HTLC$_p$: β-rules

$$\frac{\Gamma \vdash t \rightarrow_\beta t : (T_p + T'_{p'})_1}{\Gamma \vdash t \rightarrow_p t : T_{p=1-p'}}\ \rightarrow_p$$

Figure 3: HTLC$_p$: Probabilistic reduction

4 is granted by a successful instance of the Execution rule, where the first premise is induced by Trivializion on e.g. d_1 and the second premise is an instance of d_2:

$$\frac{\Gamma \vdash t^* : *^{4_{0.16666666666}} \qquad \Gamma, \Delta \vdash d_2 : 4_{0.16666666666} \qquad | 4_{0.16666666666} - 4_{0.16666666666} | = 0}{\Gamma, \Delta \vdash d_2 : 4_{0.16666666666}}\ \text{Execution}$$

Similarly, distinguishing among the two dice with respect to the outputs $1, 2, 3, 5, 6$ is possible by a failing instance of the Execution rule using any of the other outputs:

$$\frac{\Gamma \vdash t^* : *^{1_{0.16666666666}} \qquad \Gamma, \Delta \vdash d_2 : 5_{0.2} \qquad | 4_{0.16666666666} - 4_{0.16666666666} | > \epsilon(n)}{\Gamma, \Delta \nvdash d_2 : 4_{0.16666666666}}\ \text{Execution}$$

2.3 Structural Properties

Structurally, the behaviour of HTLC$_p$ differs from its deterministic counterpart and it is inspired by TPTND in [5], see Figure 6.

$$\frac{\Gamma \vdash t : T_p \qquad t \twoheadrightarrow t'}{\Gamma \vdash t' : T'_{p'} \subseteq T_p}$$

Figure 4: HTLC: Subject reduction

$$\frac{\Gamma \vdash t : T^i_{p_i}, \forall T^i_{p_i} \in \Gamma \mid \sum^i_{1\ldots n} p_i = 1 \qquad t \twoheadrightarrow t'}{\Gamma \vdash t = t'}$$

Figure 5: HTLC: Term Identity

By the Weakening rule, given the probability of a computation t to have output T with probability p, provided the value assigned to some variable x_1 is T^1 with probability p_1; if the latter assignment is independent ($\perp\!\!\!\perp$) of a distribution Δ from which some process $t_2 : T^2_{p_2}$ follows, then the starting conditional probability can be extended by additional assumptions Δ and the probability p is left unaltered. In other words: any given probability distribution can be extended at will without inferring new probabilities, as long as no new dependencies are introduced.

The Contraction rule expresses a form of plausibility test on hypotheses. Given more than one distinct hypothesis on the theoretical probability p of some process with output T^1, on which computation t with output T with expected probability p depends, a contraction is a function $f[0,1]$ which extracts one value $x : T^1_{p_i}$ from the hypotheses $p_1, \ldots, p_n$. The function f can be chosen at will, e.g. the Maximum Likelihood function.

3 Normalization

In this section we adapt the Normalization strategy shown for HTLC in [11] and provide an appropriate novel interpretation for the role of hyperintensional terms of probabilistic computations.

Applying Execution is possible to derive distinct equivalent com-

$$\frac{\Gamma, x_1 : T^1_{p_1} \vdash t : T_p \qquad \Delta \vdash t_2 : T^2_{p_2} \qquad x_1 : T^1_{p_1} \perp\!\!\!\perp \Delta}{\Gamma, x_1 : T^1_{p_1}, \Delta \vdash t : T_p} \text{ Weakening}$$

$$\frac{\Gamma, x_1 : T^1_{p_1}, \ldots, x_1 : T^1_{p_n} \vdash t : T_p}{\Gamma, x_1 : T^1_{f(p_1,\ldots,p_n)} \vdash t : T_p} \text{ Contraction}$$

Figure 6: Structural Rules

putations (i.e. terms for which the symmetric closure of $\twoheadrightarrow$ holds) of a base type or of a function type, and hence they return the same output value with the same probability when executed. When such reduction includes disjunction types, evaluation may lead to terms that satisfy only some of their subtypes. This is the classical example of failing identity for hyperintensions of probabilistic computations. Normalization grants the identity of processes with outputs with the same probability, when these are obtained by Execution on the common hyperintensional term. In particular, an application of Trivialization on distinct but reducible terms t_1, t_2 (denoting e.g. respectively the fair and the biased die) may induce distinct hyperintensional terms t^*_1, t^*_2 (two distinct representation of the distributions) of the same type. We would like to show that they may be indistinguishable from the point of view of output 4 when Execution is applied, but distinguishable for other outputs, as illustrated above. To show the first case, we need to restrict evaluation to β-reductions:

Lemma 1 (Trivialized Diamond)**.** *Let $\Gamma \vdash t_1 : T_p$, $\Gamma \vdash t_2 : T_p$ and $t_1 \twoheadrightarrow_\beta t_2$. Let, moreover, $t^*_1 : *^{T_p}$ be obtained from $t_1 : T_p$ by Trivialization, and $t^*_2 : *^{T_p}$ be obtained from $t_2 : T_p$ by Trivialization. Then $t^*_1 \twoheadrightarrow_\beta t^*_2$.*

Proof. Identical to the proof for HTLC, see [11]. □

Theorem 1 (Church-Rosser)**.** *If $\Gamma \vdash t : T_p$, $t \twoheadrightarrow_\beta t'$ and $t \twoheadrightarrow_\beta t''$ then there is a term u such that $t' \twoheadrightarrow_\beta u$ and $t'' \twoheadrightarrow_\beta u$ and $\Gamma \vdash u : T_p$.*

Proof. By induction on t, t', t'', and u using Subject Reduction and the Diamond Lemma if the term u is of the form t'^* (and thus t''^*). □

Note that the above results do not hold in general for $\twoheadrightarrow$, i.e. when evaluation includes the reduction from disjunction types to any of their subtypes, as the same premise may induce different conclusions (i.e. differently probabilistically indexed outputs).

To illustrate this further, let us come back to our example. Consider two dice, for which it can be proven that if $d_1 : 4_{0.16666666666}$ and $d_2 : 4_{0.16666666666}$, then the hyperintensional term d_1^* will also reduce to d_2^*, i.e they will have the same type (behaviourally: the same output) by Subject Reduction. Then for d_1, d_2 there is also a term u to which they both reduce and for which $u^* : *^{4_{0.16666666666}}$ can be obtained by Trivialization. Note, however, that such reasoning cannot be performed for all other outputs of d_1, d_2 as requested by Term Identity. Hence, when $\rightarrow_p$ is involved different types may be obtained and identity of terms resulting from Execution fails.

4 On diverging semantic computations

The reading of hyperintensions as terms denoting computational processes with probabilistic outputs allows to maintain a distinction natural in TIL, see [14]:

- syntactic computation (also dubbed ”construction transformation”) corresponds to a computational step obtained by a β-reduction;
- semantic computation (or "v-constructing") correlates to a computational step returning a specific interpretation or evaluation of a λ-term.

Note that for construction transformation it is assumed that β-reduction preserves strict equivalence. From this perspective, in our notation $\twoheadrightarrow_\beta$ is syntactic computation, while typing expresses semantic computation, which includes $\twoheadrightarrow_p$. Then Term Identity and Normalization express syntactic and semantic equivalence. However, probabilistic computations with our weaker notion of Subject reduction introduce diverging semantic computations by non-reducible syntactic transformations:

Proposition 1. *Consider t_1, t_2 such that $t_1 : T_p$ and $t_2 : T_p$, then $t_1 \twoheadrightarrow_\beta t_2$. Consider further $t_1 : U_{1-(p-n)}$ and $t_2 : U'_{1-(p-n')}$ but it is not the case that $t_1 \twoheadrightarrow t_2$ because in particular it is not the case that $t_1 \twoheadrightarrow_p t_2$. Then $t_1 \neq t_2$.*

In the present paper we do not further investigate the relation of HTLC$_p$ hyperintensions with other relevant aspects of TIL, e.g. the different kinds of β-conversions possible when considering the contrast between syntactic and semantic transformations, or the connection to Church-Rosser, see [9, 12]

5 Conclusions

The system HTLC$_p$ introduced in this paper is a typed λ-calculus with hyperintensions for probabilistic computations. Its semantics is defined by rules for expressions in which contexts are representations of probability distributions, terms are computational processes and types are their outputs (values or types). Under this reading, hyperintensional terms are useful to denote computations with some identical probabilistic outputs, but obtained under possibly different probability distributions. The ability to make such distinctions is granted by a strong form of term identity and a Normalization result by appropriate versions of the Diamond Lemma and the Church-Rosser Theorem holding for β-reducible terms in a system that allows further reducibility of probabilistic terms.

A natural domain of application for logics such as HTLC$_p$ is the verification of AI systems for which only certain behavioural properties are accessible, while their inner structure may remain opaque. In such cases, the ability to evaluate identity with respect to transparent models is essential to establish safety and liveness properties, see

[1, 2], or more generally accuracy and trustworthiness, as done e.g. in [16].

Extensions of HTLC$_p$ concern in particular: its relational semantics, further study of its meta-theoretical properties and applications to bias identification for labeling methods in AI.

References

[1] Nicola Angius and Giuseppe Primiero. The logic of identity and copy for computational artefacts. *J. Log. Comput.*, 28(6):1293–1322, 2018.

[2] Nicola Angius and Giuseppe Primiero. Copying safety and liveness properties of computational artefacts. *Journal of Logic and Computation*, 08 2022. exac053.

[3] M. J. Cresswell. Hyperintensional logic. *Studia Logica*, 34(1):25–38, 1975.

[4] Fabio Aurelio D'Asaro and Giuseppe Primiero. Probabilistic typed natural deduction for trustworthy computations. In Dongxia Wang, Rino Falcone, and Jie Zhang, editors, *Proceedings of the 22nd International Workshop on Trust in Agent Societies (TRUST 2021) Co-located with the 20th International Conferences on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3-7, 2021*, volume 3022 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021.

[5] Fabio Aurelio D'Asaro and Giuseppe Primiero. Checking trustworthiness of probabilistic computations in a typed natural deduction system, 2022. `https://arxiv.org/abs/2206.12934`.

[6] Bruno de Finetti. *Theory of Probability: A Critical Introductory Treatment.* New York: John Wiley, 1970.

[7] M. Duží. Hyperintensions as abstract procedures. Presented at Congress on Logic Methodology and Philosophy of Science and Technology 2019, 2019.

[8] M. Duží, B. Jespersen, and P. Materna. *Procedural Semantics for Hyperintensional Logic – Foundations and Applications of Transparent Intensional Logic*, volume 17 of *Logic, Epistemology, and the Unity of Science*. Springer, 2010.

[9] Marie Duží and Miloš Kosterec. A valid rule of β-conversion for the logic of partial functions. *Organon F*, 24(1):10–36, 2017.

[10] Marie Duzí and Marek Mensík. Inferring knowledge from textual data by natural deduction. *Computación y Sistemas*, 24(1), 2020.

[11] Michal Fait and Giuseppe Primiero. HTLC: hyperintensional typed lambda calculus. *FLAP*, 8(2):469–496, 2021.

[12] Milos Kosterec. Substitution contradiction, its resolution and the church-rosser theorem in TIL. *J. Philos. Log.*, 49(1):121–133, 2020.

[13] Henry Ely Kyburg. *Studies in Subjective Probability*. Krieger, 1964.

[14] Ivo Pezlar. On two notions of computation in transparent intensional logic. *Axiomathes*, 29(2):189–205, 2018.

[15] C. Pollard. Hyperintensions. *Journal of Logic and Computation*, 18(2):257–282, 2008.

[16] Alberto Termine, Giuseppe Primiero, and Fabio Aurelio D'Asaro. Modelling accuracy and trustworthiness of explaining agents. In Sujata Ghosh and Thomas Icard, editors, *Logic, Rationality, and Interaction - 8th International Workshop, LORI 2021, Xi'ian, China, October 16-18, 2021, Proceedings*, volume 13039 of *Lecture Notes in Computer Science*, pages 232–245. Springer, 2021.