Probabilistic Typed Natural Deduction for Trustworthy Computations

Fabio A. D'Asaro
Department of Philosophy
"Piero Martinetti",
University of Milan, Italy
fabio.dasaro@unimi.it

Giuseppe Primiero
Department of Philosophy
"Piero Martinetti",
University of Milan, Italy
giuseppe.primiero@unimi.it

Abstract

In this paper we present a typed natural deduction calculus to analyze the formal design of probabilistic computational processes, e.g. in use in machine learning techniques, to model their trustworthiness as a safety property.

1 Introduction

In this paper we introduce a natural deduction derivation system dubbed *Trustworthy Probabilistic Typed Natural Deduction* (*TPTND* for short). The aim of this system is to automatize the task of reasoning about computational processes with probabilistic outputs. In particular, we model computational processes as (generalized) Bernoulli random variables from which we draw a number of independent samples. We interpret the latter as independent runs of the process, and we refer to them as experiments. This allows us to derive properties of these processes, and in particular to verify their trustworthiness. The latter is modelled as a form of statistical hypothesis testing, and it is intended as the acceptable distance of the probabilistic value of the output of an experiment from the value intended by the relevant distribution.

Our framework uses typed sequent calculus-style judgements, where the context of derivation can be intuitively interpreted as representing a probability distribution, and the derived formula as a series of samples extracted from such distribution. Types and terms of our calculus are decorated with an empirical probability and a sample size respectively. For example, the TPTND judgement

$$\Gamma \vdash toss_{100} : Heads_{0.35}$$

may be interpreted as the statement that, under distribution Γ, tossing a coin 100 times produced output *Heads* 35% of the times. Γ will specify the value assigned to the output of each random variable of interest to the experiment of tossing a coin, and in particular whether one is assuming that the coin is fair or biased. It is under such assumption that one wants to judge the trustworthiness of the process at hand, e.g. we may ask whether the *toss* process follows a fair distribution over the outcomes $Heads$ and $Tails$. In particular, assuming that Γ expresses the distribution corresponding to a fair coin as the possible outputs declaration of one random variable $\Gamma = \{x : Heads_{0.5}, x : Tails_{0.5}\}$, the process *toss* should be considered trustworthy if increasing the number n of its runs, the probability of output $Heads$ gets arbitrarily closer to 0.5. An experiment in TPTND corresponds

In: R. Falcone, J. Zhang, and D. Wang (eds.): Proceedings of the 22nd International Workshop on Trust in Agent Societies, London, UK on May 3-7, 2021, published at http://ceur-ws.org

to the basic syntactic transformation of a term increasing its sample size from n to $n+1$. Generalizing from this toy example, any probabilistic computation can be understood as trustworthy in this sense, and therefore our system can be applied to verify trust in e.g. classifiers to produce the correct output.

As standard for type systems, a soundness result is formulated in terms of type safety, which for TPTND establishes that the syntactic transformations of computational processes allowed in the system

1. do not permit to output unintended results (read: side-effects); and
2. provide the means to assess whether a computation is trustworthy, namely if after a (sufficient) number of executions the probabilistic value of the current output approximates the one intended by the corresponding distribution.

In our example, if the distance of the current output $Heads_{0.35}$ from the value $Heads_{0.50}$ assigned in the distribution decreases as a function of n execution of the process *toss*, we say that this process is trustworthy. We consider this system a useful step towards the formal design and automatization of reasoning about probabilistic computations like those in use in machine learning techniques, as well as for the development of proof-checking protocols on such computational systems. In the following, we show the internal workings of TPTND by extending a simple die rolling example. However, we also hint at concrete applications using a more realistic example inspired by gender-bias verification.

The remainder of this paper is organised as follows. In Section 2 we briefly overview some of the most relevant related works in the area of probabilistic type and natural deduction systems, as well as in the area of computational trust using such logical tools. In Section 3 we introduce TPTND through its language, its rules and its structural properties. In Section 4 we provide a series of examples to show the functioning of the system. In Section 5 we offer meta-theoretical results leading to the formulation of safety illustrated above. Finally in Section 6 we sumamrize and sketch the next steps of this research.

2 Related Work

The extension of type systems with probabilities has been recently explored in the literature. Similarly to our work, [Pie20], [Bor19], [GIK$^+$18] introduce some form of probability in calculi with types or natural deduction systems.

[Pie20] introduces a λ-calculus augmented with special "probabilistic choice" constructs, i.e. terms of the form $M = \{p_1 M_1, \ldots, p_n M_n\}$, meaning that term M has probability $p_1, \ldots, p_n$ of reducing to one of the terms $M_1, \ldots, M_n$ respectively. Unlike TPTND, [Pie20] deals with judgements that do not have a context and uses a subtyping relation for the term reduction.

[Bor19] introduces "probabilistic sequents" of the form $\Gamma \vdash_a^b \Delta$ that are interpreted as empirical statements of the form "the probability of $\Gamma \vdash \Delta$ lies in the interval $[a, b]$". Differently from this work, TPTND does not express explicitly probabilistic intervals, but only sharp probability values, while at the same time expressing such probability on output types, rather than on the derivability relation.

Finally, [GIK$^+$18] introduces the logic P$\Lambda_{\rightarrow}$ where it is possible to mix "basic" and "probabilistic" formulas, the former being similar to standard Boolean formulas, and the latter being formed starting from the concept of a "probabilsitic operator" $P_{\geqslant s} M : \sigma$ stating that the probability of $M : \sigma$ is equal to or greater than s. The semantics for P$\Lambda_{\rightarrow}$ is Kripke-like and its axiomatisation is infinitary.

TPTND offers an integrated way to deal with probability intervals (similarly to [GIK$^+$18]) and probabilistic choices (similarly to [Pie20]). However, unlike these languages, TPTND was mainly designed to reason about and derive trustworthiness properties of computational processes. In fact, differently from all these other systems, TPTND has an explicit syntax both to deal with the number of experiments, and to prove trust in a process whenever the empirically verified probability is close enough to the theoretical one.

The application of TPTND is specifically targeted to the evaluation of trustworthiness of probabilistic computational processes. The literature offers several logical frameworks providing a computational notion of trust which deals with one of its many aspects, from manipulation to verification to assessment [Dem04, Sin11, DBS17, Ald18, AT19, DQBS20]. Modal logics, and in particular knowledge and belief logics, have been extensively investigated for trust, see e.g. [Lia03, HLHV10, LL17]. Much less investigated are proof systems for trust. An exception is the family of logic `(un)SecureND` whose most complete formulation, including also a relational semantics, is offered in [Pri20]. The proof-theoretic fragment of the logic has been applied to: software management [BPR15, PB17, PB18] including a Coq implementation (`https:`

$$\frac{}{\{\} :: \mathit{distribution}}\ \text{base} \qquad \frac{\Gamma \vdash t_n : \alpha_a \qquad a + \sum_{x:\beta_b \in \Gamma} b \leqslant 1}{\Gamma, x : \alpha_a :: \mathit{distribution}}\ \text{extend}$$

Figure 1: Distribution Formation Rules

$$\frac{}{\Gamma, x : \alpha_a \vdash t_n : \alpha_{\tilde{a}}}\ \text{exp (where } \tilde{a} \sim B(a, n))$$

$$\frac{\Gamma \vdash t_n : \alpha_{\tilde{a}} \qquad \Gamma \vdash t_m : \alpha_{\tilde{a}'}}{\Gamma \vdash t_{n+m} : \alpha_{\tilde{a} * (n/(n+m)) + \tilde{a}' * (m/(n+m))}}\ \text{upd} \qquad \frac{}{\Gamma \vdash t_n : \bot_0}\ \text{I}\bot$$

Figure 2: Rules of Calculus

`//github.com/gprimiero/SecureNDC`); modelling a trust and reputation protocol for VANET [PRCN17]; and the evaluation of the trustworthiness of online sources [CP19]. The system TPTND presented in this paper can be seen as the probabilistic extension of the fragment of `(un)SecureND` including only the closure of trust under negation which corresponds to distrust in the latter system.

3 TPTND

3.1 Syntax

The syntax of TPTND is as follows:

Definition 1 (Syntax)**.**

$$\begin{aligned}
\mathtt{T} &:= x \mid t_n \mid \langle t, u \rangle_n \mid \mathit{fst}(t)_n \mid \mathit{snd}(t)_n \mid [x]t_n \mid t.u_n \\
\mathtt{O} &:= \alpha_r \mid (\alpha \times \beta)_r \mid (\alpha + \beta)_r \mid (\alpha \to \beta)_{[r']_r} \mid \bot_r \\
\mathtt{R} &:= \Gamma; \cdot \mid \Gamma; x : \alpha_a \\
\mathtt{S} &:= \Gamma :: \mathit{distribution}
\end{aligned}$$

Terms `T` are computational processes, including: variables, constants indexed by a natural number value, pairs of constants, a first and second projection function, an abstraction term and an application. In TPTND, terms with different names are regarded as *independent* processes. In the remainder of this paper, we assume that each process has its associated variable, e.g., t has variable x_t (and vice versa). This is useful in particular to guarantee that distinct processes (possibly with the same outputs) are used correctly in the construction of a distribution. However, to simplify notation, we do not decorate variables with subscripts unless necessary. Thus, throughout we implicitly assume that, whenever a process and a variable occur in an expression, they correspond to each other. Types `O`, deviating from the standard interpretation, are output values with a probability $r \in [0, 1]$ attached: α is a meta-variable for the output of a computational process; $\alpha \times \beta$ is the independent occurrence of outputs α and β; $\alpha + \beta$ is the disjunction of outputs α and β; $\alpha \to \beta$ expresses output type β as a function of α: in this case the probability value is itself the probability r of β given the probability r' of α, and its meaning is explained by substituting to the latter an empirically determined probability value for a process whose output is α; we use $\bot$ to denote "no output" or a non-termination condition. *Contexts* `R` are coherent probability distributions, composed by expressions of the form $x : \alpha_a$. For instance, the context $\{x : \mathit{Heads}_{1/2}, x : \mathit{Tails}_{1/2}, y : \mathit{Heads}_{1/3}, y : \mathit{Heads}_{2/3}\}$ defines two independent random variables x and y which may be intuitively interpreted as a fair and biased coin respectively. *Type declarations* `S` include statements of the form $\Gamma :: \mathit{distribution}$ expressing that that Γ is a validly defined distribution. *Judgements* in TPTND have the form

$$\Gamma \vdash t_n : \alpha_{\tilde{a}}$$

saying that under a coherent probability distribution, encoded in Γ, a given process t has empirical probability $\tilde{a}$ to produce an output of type α after a sample of n of experiments. Intuitively, we can interpret $\vdash$ as the process of extracting n samples from the categorical distribution Γ and getting $n \times \tilde{a}$ times output α. Note that we use the notation $\tilde{a}$ to denote an empirical probability, as opposed to its theoretical probability counterpart which we simply denote as a. Table 1 summarizes the main notational conventions used throughout the paper.

Symbol(s)	Meaning/Used for:
t, u, t', u' ...	Processes
x, y, x', y', ...	Variables
α, β, α', β', ...	Output types
a, b, a', b', ...	Probabilities (Real numbers in $[0,1]$)
$\tilde{a}$, $\tilde{b}$, $\tilde{a'}$, $\tilde{b'}$, ...	Empirical probabilities (Real numbers in $[0,1]$)
n, n', ...	Natural numbers (usually representing sample size)
$x : \alpha_a$	Variable x gives output α with probability a
$t_n : \alpha_{\tilde{a}}$	n executions of process t give output α with empirical probability $\tilde{a}$
Γ, Δ, Γ', Δ', ...	Contexts defining probability distributions, i.e., sets of the form $\{x : \alpha_a, x' : \alpha'_{a'}, \dots\}$
$\Gamma \vdash t_n : \alpha_{\tilde{a}}$	Sampling from probability distribution in Γ results in $t_n : \alpha_{\tilde{a}}$
$[a]b$	Value b depends on a
$[a'/a]b$	Substitution of a' to a in b
$\langle\cdot,\cdot\rangle$, fst, snd, λ	Symbols to compose processes
$\times$, $+$, $\to$	Symbols to compose types

Table 1: Notation

$$\frac{\Gamma \vdash t_n : \alpha_{\tilde{a}} \qquad \Delta \vdash u_m : \beta_{\tilde{b}} \qquad t \neq u}{\Gamma, \Delta \vdash \langle t, u\rangle_{n\cdot m} : (\alpha \times \beta)_{\tilde{a} * \tilde{b}}}\ \mathrm{I}\times \qquad \frac{\Gamma \vdash t_n : \alpha_{\tilde{a}} \qquad \Gamma \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{a}+\tilde{b}}}\ \mathrm{I}+$$

$$\frac{\Gamma \vdash t_n : (\alpha \times \beta)_{\tilde{c}} \qquad \Gamma \vdash fst(t)_m : \alpha_{\tilde{a}}}{\Gamma \vdash snd(t)_{n/m} : \beta_{\tilde{c}/\tilde{a}}}\ \mathrm{E}\times_L \qquad \frac{\Gamma \vdash t_n : (\alpha \times \beta)_{\tilde{c}} \qquad \Gamma \vdash snd(t)_m : \beta_{\tilde{b}}}{\Gamma \vdash fst(t)_{n/m} : \alpha_{\tilde{c}/\tilde{b}}}\ \mathrm{E}\times_R$$

$$\frac{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{c}} \qquad \Gamma \vdash t_n : \alpha_{\tilde{a}}}{\Gamma \vdash t_n : \beta_{\tilde{c}-\tilde{a}}}\ \mathrm{E}+_L \qquad \frac{\Gamma \vdash t_n : (\alpha + \beta)_{\tilde{c}} \qquad \Gamma \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash t_n : \beta_{\tilde{c}-\tilde{b}}}\ \mathrm{E}+_R$$

$$\frac{\Gamma, x : \alpha_a \vdash t_n : \beta_{\tilde{b}}}{\Gamma \vdash [x]t_n : (\alpha \to \beta)_{[a]\tilde{b}}}\ \mathrm{I}\to \qquad \frac{\Gamma \vdash [x]t_n : (\alpha \to \beta)_{[a]\tilde{b}} \qquad \Delta \vdash u_m : \alpha_{\tilde{a}}}{\Gamma, \Delta \vdash (t.u)_n : \beta_{[\tilde{a}/a]\tilde{b}}}\ \mathrm{E}\to$$

Figure 3: Rules for Connectives

3.2 Typing Rules

In this section we present the rules of TPTND. In Figure 1 we show the formation rules for the probability distribution. Contexts are built inductively. We start from an empty context (rule "base"), and can extend it with $x : \alpha_a$ whenever we are provided with an estimate a for the probability of output α (rule "extend"), provided the sum of probabilities over outputs for variable x does not exceed 1. In view of the association of processes and variables, it holds that, for any context Γ, a variable cannot yield a single output with two (or more) distinct probabilities. For instance, $\{x : Heads_{1/2}, x : Heads_{1/3}\}$ is not a valid context. In particular, in the rule "extend" the process t_n is meant to add the output α to the family of variables $x : \beta$, as indicated in the side condition of the rule. In addition, for any given variable x, the sum of the probabilities over its output types is always less or equal than 1, therefore e.g. $\{x : Heads_{0.6}, x : Tails_{0.7}\}$ is not a valid context.

In Figure 2 we present the basic derivation rules of the calculus. Given a categorical distribution, including a random variable which is assigned a probability a to output α, a term t has empirical probability $\tilde{a}$ to produce α after a sample of n of experiments (rule "exp") where $\tilde{a}$ follows a Binomial distribution with parameters a

$$\frac{\Gamma, x : \alpha_a :: distribution \qquad \Delta \vdash u_n : \alpha_{\tilde{a}'} \qquad \mid a - \tilde{a}' \mid \leqslant \epsilon(n)}{\Gamma, \Delta \vdash\ Trust(u_n : \alpha_{\tilde{a}'})}\ \mathrm{IT}$$

$$\frac{\Gamma, x : \alpha_a :: distribution \qquad \Delta \vdash u_n : \alpha_{\tilde{a}'} \qquad \mid a - \tilde{a}' \mid > \epsilon(n)}{\Gamma, \Delta \vdash \neg\, Trust(u_n : \alpha_{\tilde{a}})}\ \mathrm{I}\neg\mathrm{T}$$

Figure 4: Trust Fragment

and n (written $B(a,n)$). Rule "upd" allows one to combine the empirical probabilities for two series of runs of a fixed process t. In fact, if n runs of process t produced output α with empirical probability $\tilde{a}$ and m runs produced output α with empirical probability $\tilde{a'}$, then it holds that $n+m$ runs produce output α with empirical probability given by the weighted sum of n and m over the total number of experiments. Recall that we regard the special type $\perp$ as a the value associated to programs which do not terminate with a valid output. Rule "I$\perp$" formalizes the intuition which states that no process in the system of interest can produce output $\perp$, or in other words that we always assume termination.

In Figure 3 we define rules for connectives. The typing rule I$\times$ says that if two distinct (and thus independent) processes t and u produced empirical probabilities $\tilde{a}$ and $\tilde{b}$ for outputs α and β and sample sizes n and m respectively, the empirical probability of jointly getting output α and β from t and u is given by $\tilde{a} * \tilde{b}$. Note that in this rule, if we have n samples extracted from t and m from u, then the sample size increases to $n \cdot m$ in the conclusion as we need to consider all possible pairs of outputs. By E$\times$, given a process for two independent outputs with associated probability $\tilde{c}$, and knowing the probability of the first composing process $\tilde{a}$, we infer the probability of the second process to produce the second output as $\tilde{c}/\tilde{a}$; the dual rule has *snd* in the second premise and fst in the conclusion.

Example 1. *Let d and g be two dice associated with distributions Δ and Γ respectively. Rolling die d 4 times produces the series 1, 3, 5, 1. We also roll die g 6 times, producing 2, 3, 6, 1, 4, 4. In our syntax, this translates to $\Delta \vdash d_4 : 1_{1/2}$ and $\Gamma \vdash g_6 : 2_{1/6}$. The empirical probability of getting a 1 from die d and a 2 from g can be estimated by considering all the pairs of rolls, i.e. the 24 pairs $(1,2)$, $(1,3)$, $(1,6)$, $(1,1)$, $(1,4)$, $(1,4)$, $(3,2)$, $(3,3)$, $(3,6)$, $(3,1)$, $(3,4)$, $(3,4)$, $(5,2)$, $(5,3)$, $(5,6)$, $(5,1)$, $(5,4)$, $(5,4)$, $(1,2)$, $(1,3)$, $(1,6)$, $(1,1)$, $(1,4)$ and $(1,4)$. The pair $(1,2)$ occurs twice in the list – this is mirrored by our I$\times$ rule:*

$$\frac{\Delta \vdash d_4 : 1_{1/2} \qquad \Gamma \vdash g_6 : 2_{1/6}}{\Delta, \Gamma \vdash \langle d, g \rangle_{24} : (1 \times 2)_{1/12}}\, I\times$$

Conversely, if we are told that the pair $(1,4)$ occurs 4 times in the list of 24 pairs of independent rolls of d and g, and that die g gave 4 on 2 rolls out of 5, then we can conclude that die d gave 1 on 2 rolls out of 4. In our calculus, this is formalized as:

$$\frac{\Delta, \Gamma \vdash \langle d, g \rangle_{24} : (1 \times 4)_{4/24} \qquad \Delta, \Gamma \vdash snd\langle d, g \rangle_6 : 4_{2/6}}{\Delta, \Gamma \vdash fst\langle d, g \rangle_4 : 1_{1/2}}\, E\times_R$$

Rule I+ introduces disjunction: intuitively, if we have that process t produces outputs α and β with frequencies $\tilde{a}$ and $\tilde{b}$ respectively, then the frequency of output α or output β is $\tilde{a} + \tilde{b}$. Rule I$\rightarrow$ introduces the implication by constructing the empirical probability of output β for process t_n as a function of the theoretical probability a of α under context Γ. The empirical probability $\tilde{b}$ of β given the theoretical probability a of α is denoted $[a]\tilde{b}$. E$\rightarrow$ eliminates such dependency substituting the theoretical probability a with the empirical probability measured under context Δ over m experiments. We denote such substitution by $[a/\tilde{a}]\tilde{b}$. Note that the estimate of a does not need to be based on the same number of experiments as those run for $[x]t_n$. The conclusion provides an estimate of the empirical probability of process t outputting β when it is run n times, according to an estimate $\tilde{a}$ obtained from m runs of process u outputting α.

3.3 Trustworthy computations

In Figure 4 we provide typing rules for trustworthy computations. A probabilistically valid output is labeled as trustworthy under a given distribution if and only if given the probability $\tilde{a}$ of producing a given output based on the known probability distribution, the computational process t after a sample of n experiments has probability $\tilde{a}'$ to produce an output of type α such that the difference between $\tilde{a}$ and $\tilde{a}'$ remains below a critical threshold, parametric with respect to the number of experiments performed (rule IT). If such a value is surpassed, the process is labelled untrustworthy (rule $\neg$T). The parametric threshold $\epsilon(n)$ is domain-specific and depends on the application. For instance, one may take $\epsilon(n)$ to correspond to the 95% confidence interval under the Normal approximation to the Binomial Distribution. Note that, according to these rules, it is possible to derive the trustworthiness of some untrustworthy process (and vice versa) with some (very small) probability. The probability of this happening can be made arbitrarily small by repeating the experiment associated with the process.

3.4 Structural Rules

Structural Rules are admissible. The Cut rule can be obtained as a detour of I→ and E→ as follows:

$$\dfrac{\dfrac{\Delta, x : \alpha_a \vdash u_n : \beta_{\widetilde{b}}}{\Delta \vdash [x]u_n : (\alpha \to \beta)_{[\tilde{a}]\widetilde{b}}}\,\mathrm{I}\to \qquad \Gamma \vdash t_n : \alpha_{\widetilde{a}}}{\Gamma, \Delta \vdash (u.t)_n : \beta_{[\tilde{a}/\tilde{a}]\widetilde{b}}}\,\mathrm{E}\to$$

Contraction would say that, given the empirical probability $\widetilde{a}$ that process t outputs α determined over m experiments; and given the probability $\widetilde{a'}$ of an independent instance t' to output α given $n > m$ experiments; the context in which the two processes are considered both can be contracted with a probability assigned to obtain α updated by rule "upd". The rule can be obtained as a detour of E*xtend*, E*xp* and *upd*:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash t_n : \alpha_{\tilde{a}}}{\Gamma, x : \alpha_a :: \mathit{distribution}}\,\text{extend} \qquad \dfrac{}{\Gamma, x : \alpha_a \vdash t_m : \alpha_{\tilde{a}'}}\,\text{exp}}{\Gamma \vdash t_{n+m} : \alpha_{\tilde{a}*(n/(n+m))+\tilde{a}'*(m/(n+m))}}\,\text{upd}}{\Gamma, y : \alpha_{\tilde{a}*(n/(n+m))+\tilde{a}'*(m/(n+m))} :: \mathit{distribution}}\,\text{ext}$$

Finally, consider that the rule *upd* is used to extend a distribution $\Gamma, x : \alpha_a$ by the result of a new experiment $\Delta \vdash t_n : \alpha_{\widetilde{a'}}$, i.e. where there is a confounder $x : \alpha_a \vdash t_n : \alpha_{\widetilde{a'}}$. For any new process without such a condition, the rule E*xtend* will act as Weakening.

4 Examples

In this section, we illustrate the working of TPTND by commenting on a toy die-rolling example.

Example 2. *We first start by modeling a fair die. Let* $\Gamma = \{x : 1_{1/6}, \ldots, x : 2_{1/6}, \ldots, x : 6_{1/6}\}$. *Then, for example, we can construct a process that outputs type* 5 *or* 6 *with probability* 2/6 *as follows:*

$$\dfrac{\dfrac{}{\Gamma \vdash t_n : 5_{\widetilde{1/6}}}\,exp \qquad \dfrac{}{\Gamma \vdash t_n : 6_{\widetilde{1/6}}}\,exp}{\Gamma \vdash t_n : (5+6)_{\widetilde{2/6}}}\,I+$$

Now let Δ be an "opaque" context, i.e. an unknown distribution. One of the use of our trust fragment is to allow reducing "opaque" contexts to "transparent" ones by verifying whether an empirical probability is sufficiently close to one we expect. In this sense, a trustworthy computation is one whose underlying distribution has been explained. The following example demonstrates this use of trust:

Example 3. *Let* Δ *be an opaque context, i.e. a "die system" which we have been provided with; and let* Γ *be the distribution defined in Example 2, representing a fair die. In this example, we focus on output* 5. *Experimenting under context* Δ *gives:*

$$\dfrac{}{\Delta \vdash u_{10} : 5_{1/2}}\,exp$$

Our aim is to decide whether Δ *describes a fair distribution over outputs* $\{1, 2, \ldots, 6\}$, *i.e. whether* Δ *approximates correctly to* Γ *and can be therefore considered safe to use. To this aim, we use the Introduction of (Negative) Trust, where we use the (exact) Binomial* 95% *confidence interval to set our threshold:*

$$\dfrac{\Gamma \vdash t_n : 5_{\widetilde{1/6}} \qquad \Delta \vdash u_{10} : 5_{1/2} \qquad 1/6 \notin [0.19, 0.81]}{\Gamma, \Delta \vdash \neg\, \mathit{Trust}(u_{10} : 5_{1/2})}\,I\neg T$$

that is, we do not trust process u *to be extracted from the distribution* Γ.

Obviously, things may change as new evidence is collected, as we show in the following example:

Example 4. *Let* Γ, Δ *be as in Examples 2 and 3, and suppose we are provided with some more experimental evidence about process* u, *i.e.* $\Delta \vdash u_{20} : 5_{3/20}$. *We can use rule "upd" to update the result from Example 3:*

$$\dfrac{\dfrac{\Delta \vdash u_{10} : 5_{1/2} \qquad \Delta \vdash u_{20} : 5_{3/20}}{\Delta \vdash u_{30} : 5_{4/15}}\ upd \qquad \Gamma, x : 5_{1/6} :: distribution \qquad 1/6 \in [0.12, 0.46]}{\Gamma, \Delta \vdash Trust(u_{30} : 5_{4/15})}$$

i.e., based on a total of 30 *runs of process* u*, now we trust it to be extracted from the transparent and fair distribution* Γ*.*

We now turn to a gender-bias inspired example, where we test whether we trust a given classifier not to have an inherent bias.

Example 5. *Suppose we are given a commercial, closed-source software to automatically shortlist CVs according to a set of criteria. To this aim, we consider the output of the classification algorithm to fall into one of the following four categories: (1) male, shortlisted, (2) male, not shortlisted, (3) female, shortlisted, (4) female, not shortlisted. To verify that the underlying algorithm does not have an inherent gender bias, we test whether we trust the distribution over these four classes to reflect the actual gender distribution in the current Italian population, as encoded in a distribution* Γ *"extended" from an official dataset. A proof that this is indeed the case would produce a certificate in the form of a natural deduction proof, that would go as follows:*

$$\dfrac{\dfrac{\vdash population_n : female_{0.52}}{\{x : female_{0.52}\} :: distribution}\ extend \qquad \Delta \vdash c_{10} : female_{3/10} \qquad 0.52 \in [0.07, 0.65]}{\Gamma, \Delta \vdash Trust(c_{10} : female_3)}\ IT$$

where we assumed a 95% *confidence interval and* $n = 62246674$ *is the Italian population in July 2018. Note that a shortlist selection which would provide strictly less than* 2/10 *or more than* 9/10 *of female candidates would be considered untrustworthy according to this model. Note that if we sampled* 30 *CVs instead, shortlisting* 7 *female CVs, we would derive untrustworthiness:*

$$\dfrac{\dfrac{\vdash population_n : female_{0.52}}{\{x : female_{0.52}\} :: distribution}\ extend \qquad \Delta \vdash c_{30} : female_{7/30} \qquad 0.52 \notin [0.1, 0.42]}{\Gamma, \Delta \vdash \neg\, Trust(c_{30} : female_{7/10})}\ I\neg T$$

We would in fact consider untrustworthy any number of shortlisted female CVs strictly smaller than 10 *or bigger than* 21*.*

We now show how our syntactic machinery allows for the definition of dependent processes:

Example 6. *Let* $\Gamma = \{x : 1_{x_1}, \ldots, x : 6_{x_6}, y : 1_{y_1}, \ldots, y : 6_{y_6}\}$ *be a distribution. Note that, according to axiom "extend",* $x_1, \ldots, x_6, y_1, \ldots, y_6$ *are variables in* $[0, 1]$ *such that* $x_1 + \cdots + x_6 \leqslant 1$ *and* $y_1 + \cdots + y_6 \leqslant 1$*. Then, for any* $n \in \mathbb{N}$*, rule "exp" gives:*

$$\dfrac{}{\Gamma \vdash t_n : 5_{\widetilde{x_5}}}\ exp \qquad \dfrac{}{\Gamma \vdash u_n : 5_{\widetilde{y_5}}}\ exp$$

and

$$\dfrac{}{\Gamma \vdash t_n : 6_{\widetilde{x_6}}}\ exp \qquad \dfrac{}{\Gamma \vdash u_n : 6_{\widetilde{y_6}}}\ exp$$

Applying the introduction of conjunction yields:

$$\dfrac{\Gamma \vdash t_n : 5_{\widetilde{x_5}} \qquad \Gamma \vdash u_n : 6_{\widetilde{y_6}}}{\Gamma \vdash \langle t, u \rangle_{n^2} : (5 \times 6)_{\widetilde{x_5} \cdot \widetilde{y_6}}}\ I\times$$

and

$$\dfrac{\Gamma \vdash t_n : 6_{\widetilde{x_6}} \qquad \Gamma \vdash u_n : 5_{\widetilde{y_5}}}{\Gamma \vdash \langle t, u \rangle_{n^2} : (6 \times 5)_{\widetilde{x_6} \cdot \widetilde{y_5}}}\ I\times$$

Finally, we can introduce disjunction:

$$\frac{\Gamma \vdash \langle t,u\rangle_{n^2} : (5\times 6)_{\widetilde{x_6}\cdot\widetilde{y_5}} \qquad \Gamma \vdash \langle t,u\rangle_{n^2} : (6\times 5)_{\widetilde{x_5}\cdot\widetilde{y_6}}}{\Gamma \vdash \langle t,u\rangle_{n^2} : ((5\times 6)+(6\times 5))_{\widetilde{x_5}\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5}}}\, I+$$

For readability, we rename the type $((5\times 6)+(6\times 5))$ *into* 11*, since the type* $((5\times 6)+(6\times 5))$ *corresponds to the only way we can get the two dice to sum to* 11*. Under this convention, the latter conclusion reads:*

$$\Gamma \vdash \langle t,u\rangle_{n^2} : 11_{\widetilde{x_5}\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5}}$$

Now we introduce $\rightarrow$*:*

$$\frac{\Gamma, x : 5_{x_5} \vdash \langle t,u\rangle_{n^2} : 11_{\widetilde{x_5}\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5}}}{\Gamma \vdash [x]\langle t,u\rangle_{n^2} : (5\rightarrow 11)_{[x_5](\widetilde{x_5}\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5})}}\, I\rightarrow$$

Finally we illustrate elimination:

$$\frac{\Gamma \vdash [x]\langle t,u\rangle_{n^2} : (5\rightarrow 11)_{[x_5](\widetilde{x_5}\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5})} \qquad \Delta \vdash v_m : 5_{0.99}}{\Gamma,\Delta \vdash (\langle t,u\rangle.v)_{n^2} : 11_{0.99\cdot\widetilde{y_6}+\widetilde{x_6}\cdot\widetilde{y_5}}}\, E\rightarrow$$

5 Metatheory

The meta-theoretical analysis of TPTND aims at proving type safety. Such result, in line with the traditional approach defined first in [WF94], is obtained by formulating type preservation in Theorem 3 and progress in Theorem 4. In TPTND these results are interpreted as follows:

- preservation: any computation implementing certain steps preserves the probability of its intended output;
- progress: any such computation is trustworthy, and viceversa.

This interpretation of progress deviates from the standard one, which requires proving that for any term either a reduction step can be formulated, or the term is in normal form. In TPTND there is no restriction on the number of term transformation which can be performed by experiment execution, hence one can repeatedly apply the *upd* rule. To reach safety, we first formulate substitution (both deterministic and non-deterministic) and rules for term evaluation.

Theorem 1 (Deterministic Substitution)**.** *If* $x{:}\alpha_a \vdash t_n : \beta_{\tilde{b'}}$ *and* $\vdash u_n{:}\alpha_1$*, then* $\vdash (t.u)_n : \beta_{\tilde{b'}}$*.*

Proof. By structural induction on the premise of the first derivation. To show the reduction to b', for each case of $x{:}\alpha_a \vdash t_n : \beta_{\tilde{b'}}$ in the induction, take the second derivation $\vdash u_n{:}\alpha_1$, where the output α does not depend on any distribution, as additional premise in the elimination from $\vdash [x/u]t_n : (\alpha\rightarrow\beta)_{[a]b}$ to derive the term $(t.u)_n : \beta_{[1/a]\tilde{b'}}$:

- For α atomic, the case is immediate;
- For $\alpha \equiv \gamma\times\delta$, there are independent terms $\langle v,z\rangle_n$ such that:
  $\vdash v_n : \gamma_1$, $x{:}\gamma_1 \vdash t_n : \beta_{b'}$ and the base case applies;
  $\vdash z_n : \delta_1$, $x{:}\delta_1 \vdash t_n : \beta_{b'}$ and the base case applies;
- For $\alpha \equiv \gamma+\delta$, the term u_n is induced from $u_n : \gamma_{\tilde{b''}}$ and $u_n : \delta_{\tilde{b'''}}$ with $b''+b''''=1$; i.e., α is a metavariable for the distribution of possible outputs of the term u. Then either one of them is such that $u_n : \gamma_1$ or $u_n : \delta_1$ (and respectively the other term has probability $= 0$), then use distributivity of $+$ over $\rightarrow$ and commutativity of $+$ to select either one, and the base case applies.
- For $\alpha \equiv \gamma\rightarrow\gamma$ (as an instance of the rule *upd*), the term u_n is of the form $[x]v_n$ with output having $b=1$; hence, $x : \gamma_1 \vdash v_n : \gamma_1$ and $y : (\gamma\rightarrow\gamma)_1 \vdash t_n : \beta_{b'}$ allows to derive $\vdash t.u_n : \beta_{b'}$;

□

In most interesting cases, substitution will make use of non-deterministic processes:

$$\frac{}{t_n \mapsto_p t_{n+1}} \quad \frac{t_n \mapsto_p t'_n}{fst(t_n) \mapsto_p fst(t'_n)} \quad \frac{t_n \mapsto_p t'_n}{snd(t_n) \mapsto_p snd(t'_n)} \quad \frac{t_n \mapsto_p t'_n}{\langle t, u\rangle_n \mapsto_p \langle t', u\rangle_n} \quad \frac{u_n \mapsto_p u'_n}{\langle t, u\rangle_n \mapsto_p \langle t, u'\rangle_n}$$

$$\frac{}{fst(\langle t, u\rangle_n) \mapsto_p t_n} \quad \frac{}{snd(\langle t, u\rangle_n) \mapsto_p u_n} \quad \frac{t_n \mapsto_p t'_n}{[x]t_n \mapsto_p [x]t'_n} \quad \frac{t_n \mapsto_p t'_n}{t.u_n \mapsto_p t'.u_n} \quad \frac{t_n \mapsto_p t'_n}{[x/t]u_n \mapsto_p [x/t']u_n}$$

Figure 5: Term Evaluation Rules

Theorem 2 (Non-deterministic Substitution)**.** *If* $x{:}\alpha_a \vdash t_n : \beta_{\tilde{b}}$*, and* $\Gamma \vdash u_n : \alpha_{\tilde{a}<1}$*, then* $\Gamma \vdash (t.u)_n : \beta_{\tilde{b'}<1}$.

Proof. By structural induction on the premise of the first derivation, as above to derive the term $(t.u)_n : \beta_{[\tilde{a}/a]\tilde{b'}}$:

- For α atomic, the case is immediate by substitution in $\Gamma \vdash (t.u)_n : \beta_{[b/a]\tilde{b'}}$, and provided $b < 1$, then $b' < 1$;
- For $\alpha \equiv \gamma \times \delta$, there are independent terms $\langle v, z\rangle_n$ such that, with possibly $\Gamma \equiv \Delta$:

  $\Gamma \vdash v_n : \gamma_b$, $\Gamma, x{:}\gamma_b \vdash t_n : \beta_{b'}$ and the base case applies;

  $\Delta \vdash z_n : \delta_b$, $\Delta, x{:}\delta_b \vdash t_n : \beta_{b'}$ and the base case applies;
- For $\alpha \equiv \gamma + \delta$, the term u_n is induced from $u_n : \gamma_{\tilde{b''}}$ and $u_n : \delta_{\tilde{b'''}}$ with $b'' + b'''' < 1$; i.e., α is a metavariable for the distribution of possible outputs of the term u and also neither $b'' = 1$ nor $b''' = 1$. Then $\Gamma, u_n : \gamma_{b''} \vdash t_n : \beta_{\tilde{b}}$ and $\Gamma, u_n : \delta_{b'''} \vdash t_n : \beta_{\tilde{b}}$, and for each the base case applies;
- For $\alpha \equiv \gamma \to \delta$, the term u_n is of the form $[x]v_n : (\gamma \to \delta)_{[a]\tilde{b}<1}$; hence, $x : \gamma_{b''} \vdash v_n : \delta_{b'''}$ and $y : (\gamma \to \delta)_{\tilde{b}<1} \vdash t_n : \beta_{\tilde{b'}}$ allows to derive $\vdash t.u_n : \beta_{[b<1]\tilde{b'}<1}$.

□

5.1 Term Evaluation

A term $t : \alpha_a$ is said to probabilistically evaluate to a term $t' : \alpha'_{a'}$ if t can be transformed into t' according to the rewriting rules in Figure 5. These term reduction rules correspond each to one typing rule from Figure 3 where some term transformation occur. Notably, they do not include rules where the term is not transformed and in particular the rules for + which express the possible outputs of a given process. Note moreover that term transformation for expression including → should be considered for their computational content, but do not affect output probability. Term reduction is abbreviated as $t_n \mapsto_p t'_n$ and $\mapsto_{p*}$ denotes its transitive and reflexive closure.

Proposition 1. *Given an aleathoric variable* $x : \alpha_a$*, the non-deterministic evaluation of atomic terms* $t_n \mapsto_{p*} t'_{n+m}$ *with output* $\alpha_{\tilde{a}}$ *has as fix point the term whose output has probabilistic value* a*:*

$$t_n \mapsto_{p*} t'_{n+m} \equiv lim_{n\to\infty}\tilde{a} = a. \tag{1}$$

Hence, by increasing the number of instances of the rule *upd*, we approximate better and better from $\tilde{a}$ the ideal value a of output α. The smaller this difference, the more trustworthy the process at hand.

Proposition 2. *If* $t \mapsto_{p*} t'$

- $t : \alpha_a$ *and* $t' : \alpha_{a'}$ *with* $a' \geqslant a$ *if and only* $\mapsto_{p*}$ *results from applying upd;*
- $t : \alpha_a$ *and* $t' : \beta_{a'}$ *with* $\beta \subset \alpha$ *and* $a' \geqslant a$ *if and only if* $\mapsto_{p*}$ *results from applying E-*$\times$*.*

Hence, term evaluation of processes with an increasing output probability has to include at least as many instances of these rules such that the difference of $\tilde{a}$ from the probability a of the associated aleathoric variable in the appropriate distribution remains within the confidence interval or it reduces.

5.2 Safety

A type safety result for TPTND expresses the fact that under trustworthiness conditions, the evaluation of a probabilistic process will lead to the intended output within allowed margins of certainty. This is shown by considering that any computation has bounded limits under which its output is probabilistically determined. This result is granted by preservation and progress.

Theorem 3 (Output preservation)**.** *If* $\Gamma, x : \alpha_a \vdash t_n : \beta_{\tilde{b}}$ *and* $t \mapsto_p^* t'$ *according to the cases of Proposition 2, then* $\Gamma \vdash t'_n : \beta_{\tilde{b'}}$ *and* $b' \geqslant b$.

Proof. The proof is by induction on the term t and the appropriate reduction step which increases the probability of the output.

$\rightarrow$
- if t_n has output $\beta \equiv \alpha$, the only reduction step which applies is in the number n of executions of the rule *upd* which preserves α while increasing for t' the value of $\tilde{b'}$ up to a.
- if t_n is a pair $\langle t, u\rangle$ with output $(\alpha \times \beta)_{\tilde{c}}$, the reduction step corresponds to an instance of E$\times$ from $\Gamma, x : \alpha_a, y : \beta_b \vdash t'_n : \alpha_{\tilde{a}}$ (respectively $t'_n : \beta_{\tilde{b}}$), which increases for t' the value of $\tilde{b'}$ compared to $\tilde{b}$.

□

Hence, any of the above reduction steps preserves the output of the reduced term.

Lemma 1. $\Gamma, x : \alpha_a \vdash t_n : \beta_{\tilde{b}}$ *and* $t \mapsto_p^* t'$ *as by Theorem 3 if and only if* $\Gamma \vdash Trust(t'_n : \beta_{\tilde{b'}})$.

Proof. The left-to-right direction follows from Theorem 3 in that in each case the reduction step decreases the difference from the probability value of the corresponding aleathoric variable; the right-to-left direction follows from the conditions for the Trust operator.

$\rightarrow$
- If t_n has output $\beta \equiv \alpha$, $\mid a - \tilde{b'} \mid = 0$ for a sufficiently large number n, which satisfies the condition for $Trust(t_n : \beta_{\tilde{b}})$.
- If $t_n : (\alpha \times \beta)_{\tilde{c}}$ and the reduction step leads to $\Gamma, x : \alpha_a, y : \beta_b \vdash t'_n : \alpha_{\tilde{a}}$ (respectively $t'_n : \beta_{\tilde{b}}$) then $\tilde{a} > \tilde{c}$ (respectively $\tilde{b} > \tilde{c}$). Then the argument reduces to the base case;

$\leftarrow$
- For the base case, if $x : \beta_b \vdash Trust(t' : \beta_{\tilde{b'}})$, then by the IT rule $\mid b - b' \mid \leqslant \epsilon(n)$. Hence the reduction $x \mapsto_{p*} t'$ must have been applied so that (for n sufficiently big) $\mid b - b' \mid = 0$.
- If $\Gamma, x : \alpha_a, y : \beta_b \vdash Trust(\langle t', u'\rangle_{m \cdot n} : ((\alpha \times \beta)_{\tilde{a} * \tilde{b}}))$, then by the IT rule $\mid (a * b) - (\tilde{a} * \tilde{b}) \mid \leqslant \epsilon(m \cdot n)$. Hence there are reduction steps for $\Gamma, x : \alpha_a \vdash t'_m : \alpha_{\tilde{a}}$ and $\Delta, y : \beta_b \vdash u'_n : \beta_{\tilde{b}}$, with $t' \neq u'$ such that for both the base case applies.

□

Theorem 4 (Progress)**.** *If* $\Gamma, x : \alpha_b \vdash t_n : \alpha_{\tilde{b}}$*, then either it exists* t' *such that* $t \rightarrow_{p*} t'$ *according to the reduction steps of Theorem 3 and* $\Gamma \vdash Trust(t'_{n+m} : \alpha_{\tilde{b'}})$*, or* t_n *is untrustworthy.*

Proof. Progress from t_n to a trustworthy computation t_{n+m} is guaranteed by Lemma 1 after a sufficiently large number of reduction steps which includes at least m new instances of the process t by the rule step *upd* (and possibly more, if t is not atomic). If any further reduction $t \rightarrow_{p*} t'$ corresponding to *upd* does not make $b - \tilde{b'} \leqslant \epsilon(n + m)$, we infer $\Gamma \vdash \neg Trust(t'_{n+m} : \alpha_{\tilde{b'}})$. The result generalizes for non-atomic terms in the case of reduction steps obtained by E-$\times$. □

Type safety for TPTND as obtained by preservation and progress tells us that any probabilistic computation can be evaluated in steps including experimental instances which increase its trustworthiness, intended as closeness to the probability of its output as by the distribution under which it is formulated. Failing to reach such margin within the appropriate number of steps, determines an untrustworthy computation. Progress for TPTND does not require termination of the reduction process for the term t_n, as usually required: this is due to the fact that the base case of such reduction corresponds to just a new sample extracted from a relevant distribution, and unless one assumes fixed resources, such execution is always possible.

6 Conclusion

We have presented a type natural deduction system with terms decorated by a sample size and types by a probability value. The intended interpretation of judgements of the system TPTND is to assert the validity of the probabilistic output of a process, given a certain probability distribution. The main use of such a calculus is the determination of trustworthy probabilistic computations, expressed as the admissible distance from their theoretical probabilities. Such notion of trustworthiness can be used also to evaluate opaque distributions, e.g. in the context of Machine Learning systems, and it is guaranteed in TPTND by a safety result under specific conditions.

Next steps of this work include: the development of a Coq verification protocol for probabilistic trustworthy computations, by extending the existing protocol for trust presented in [PB18] with one of the available Coq libraries for probabilistic reasoning, e.g. `https://github.com/jtassarotti/polaris`; the extension of TPTND with probabilistic intervals and imprecise probabilities; the use of TPTND in the verification of biased computations. Another variation of TPTND can be foreseen which takes into account a finite number of resources, especially for experiments: in this format, safety can rely on a normal form for terms which is reached after a fixed number of possible experiments. Finally, TPTND falls within current investigations in uncertain reasoning and the development of sound relational and state transition semantics for this system are also planned.

Acknowledgments

This research was funded by the Department of Philosophy "Piero Martinetti" of the University of Milan under the Project "Departments of Excellence 2018-2022" awarded by the Ministry of Education, University and Research (MIUR). The authors also thankfully acknowledge the support of the Italian Ministry of Education, University and Research (PRIN 2017 project n. 20173YP4N3). Giuseppe Primiero acknowledges support of the NIAS-Lorentz Programme for funding the Interdisciplinary Theme Group on "Accountability in Medical Autonomous Expert Systems".

References

[Ald18] Alessandro Aldini. Design and verification of trusted collective adaptive systems. *ACM Trans. Model. Comput. Simul.*, 28(2):9:1–9:27, 2018.

[AT19] Alessandro Aldini and Mirko Tagliaferri. Logics to reason formally about trust computation and manipulation. In Andrea Saracino and Paolo Mori, editors, *Emerging Technologies for Authorization and Authentication - Second International Workshop, ETAA 2019, Luxembourg City, Luxembourg, September 27, 2019, Proceedings*, volume 11967 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2019.

[Bor19] Marija Boričić. Sequent calculus for classical logic probabilized. *Archive for Mathematical Logic*, 58(1-2):119–136, 2019.

[BPR15] Jaap Boender, Giuseppe Primiero, and Franco Raimondi. Minimizing transitive trust threats in software management systems. In Ali A. Ghorbani, Vicenç Torra, Hüseyin Hisil, Ali Miri, Ahmet Koltuksuz, Jie Zhang, Murat Sensoy, Joaquín García-Alfaro, and Ibrahim Zincir, editors, *13th Annual Conference on Privacy, Security and Trust, PST 2015, Izmir, Turkey, July 21-23, 2015*, pages 191–198. IEEE Computer Society, 2015.

[CP19] Davide Ceolin and Giuseppe Primiero. A granular approach to source trustworthiness for negative trust assessment. In Weizhi Meng, Piotr Cofta, Christian Damsgaard Jensen, and Tyrone Grandison, editors, *Trust Management XIII - 13th IFIP WG 11.11 International Conference, IFIPTM 2019, Copenhagen, Denmark, July 17-19, 2019, Proceedings*, volume 563 of *IFIP Advances in Information and Communication Technology*, pages 108–121. Springer, 2019.

[DBS17] Nagat Drawel, Jamal Bentahar, and Elhadi M. Shakshuki. Reasoning about trust and time in a system of agents. In Elhadi M. Shakshuki, editor, *The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017) / The 7th International Conference on Sustainable Energy Information Technology (SEIT 2017), 16-19 May 2017, Madeira, Portugal*, volume 109 of *Procedia Computer Science*, pages 632–639. Elsevier, 2017.

[Dem04] Robert Demolombe. Reasoning about trust: A formal logical framework. In Christian Damsgaard Jensen, Stefan Poslad, and Theodosis Dimitrakos, editors, *Trust Management, Second International Conference, iTrust 2004, Oxford, UK, March 29 - April 1, 2004, Proceedings*, volume 2995 of *Lecture Notes in Computer Science*, pages 291–303. Springer, 2004.

[DQBS20] Nagat Drawel, Hongyang Qu, Jamal Bentahar, and Elhadi M. Shakshuki. Specification and automatic verification of trust-based multi-agent systems. *Future Gener. Comput. Syst.*, 107:1047–1060, 2020.

[GIK+18] Silvia Ghilezan, Jelena Ivetić, Simona Kašterović, Zoran Ognjanović, and Nenad Savić. Probabilistic reasoning about simply typed lambda terms. In *International Symposium on Logical Foundations of Computer Science*, pages 170–189. Springer, 2018.

[HLHV10] Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, and Laurent Vercouter. A logic of trust and reputation. *Log. J. IGPL*, 18(1):214–244, 2010.

[Lia03] Churn-Jung Liau. Belief, information acquisition, and trust in multi-agent systems–a modal logic formulation. *Artif. Intell.*, 149(1):31–60, 2003.

[LL17] Fenrong Liu and Emiliano Lorini. Reasoning about belief, evidence and trust in a multi-agent setting. In Bo An, Ana L. C. Bazzan, João Leite, Serena Villata, and Leendert W. N. van der Torre, editors, *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings*, volume 10621 of *Lecture Notes in Computer Science*, pages 71–89. Springer, 2017.

[PB17] Giuseppe Primiero and Jaap Boender. Managing software uninstall with negative trust. In Jan-Philipp Steghöfer and Babak Esfandiari, editors, *Trust Management XI - 11th IFIP WG 11.11 International Conference, IFIPTM 2017, Gothenburg, Sweden, June 12-16, 2017, Proceedings*, volume 505 of *IFIP Advances in Information and Communication Technology*, pages 79–93. Springer, 2017.

[PB18] Giuseppe Primiero and Jaap Boender. Negative trust for conflict resolution in software management. *Web Intell.*, 16(4):251–271, 2018.

[Pie20] A. D. Pierro. A type theory for probabilistic λ-calculus. In *From Lambda Calculus to Cybersecurity Through Program Analysis*, 2020.

[PRCN17] Giuseppe Primiero, Franco Raimondi, Taolue Chen, and Rajagopal Nagarajan. A proof-theoretic trust and reputation model for VANET. In *2017 IEEE European Symposium on Security and Privacy Workshops, EuroS&P Workshops 2017, Paris, France, April 26-28, 2017*, pages 146–152. IEEE, 2017.

[Pri20] Giuseppe Primiero. A logic of negative trust. *J. Appl. Non Class. Logics*, 30(3):193–222, 2020.

[Sin11] Munindar P. Singh. Trust as dependence: a logical approach. In Liz Sonenberg, Peter Stone, Kagan Tumer, and Pinar Yolum, editors, *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*, pages 863–870. IFAAMAS, 2011.

[WF94] Andrew K. Wright and Matthias Felleisen. A syntactic approach to type soundness. *Inf. Comput.*, 115(1):38–94, 1994.