A Quantitative Doxastic Logic for Probabilistic Processes and Applications to Information-Hiding

Simon Kramer, Catuscia Palamidessi, Roberto Segala, Andrea Turrini, Christelle Braun

0

A Quantitative Doxastic Logic for Probabilistic Processes and Applications to Information-Hiding ⋆

Simon Kramer[**] — **Catuscia Palamidessi**[*] — **Roberto Segala**[***] — **Andrea Turrini** [***]— **Christelle Braun**[*]

[*] *INRIA and LIX, École Polytechnique*
France
{braun,catuscia}@lix.polytechnique.fr

[**] *Laboratory of Cryptography and Information Security*
Tsukuba, Japan
simon.kramer@a3.epfl.ch

[***] *Dipartimento di Informatica, Università degli Studi di Verona*
Italy
{roberto.segala,andrea.turrini}@univr.it

ABSTRACT. *We introduce a novel modal logic, namely the* doxastic μ-calculus with error control *(DμCEC), and propose a formalization of* probabilistic anonymity *and* oblivious transfer *in the logic, and the validation of these formalizations on implementations formalized in probabilistic CCS.*

The distinguishing feature of our logic is to provide a combination of dynamic *operators for belief (whence the attribute "doxastic") with a* control *on the possible error of apprehension of the perceived reality, and for internalized probability. Both operators are dynamic (non-monotonic) thanks to the possibility of combining them with temporal operators, and are parameterized with a lower and upper probability bound (the error control).*

KEYWORDS: *Anonymity and information-hiding, doxastic μ-calculus with error control, oblivious transfer, probabilistic process calculi, multi-agent systems*

DOI:10.3166/JANCL.18.0–27

⋆. This work has been partially supported by the INRIA DREI Équipe Associée PRINTEMPS, and it has been carried over during the visit of Simon Kramer at LIX on a postdoc position.

1. Introduction

1.1. *Motivation*

Information-hiding is a major concern in today's networked world of interacting agents, be they unconcerned Internet bots or caring humans concerned by malicious such bots (spambots, viruses, worms, etc.). *Anonymity* (the absence of information about an agent's identity) is a major *end* of information-hiding. Whereas *oblivious transfer* (implying the absence of information about an agent's receiving action to the corresponding sending agent) is a major and even foundational (Kilian, 1988) *means* thereto. Since protocols for information hiding often involve randomization, it seems natural to consider a *quantitative* approach to the formalization of information-hiding properties.

1.2. *Contribution*

We propose a modal logic, namely the *doxastic μ-calculus with error control* (DμCEC). The formulas of DμCEC are interpreted in terms of the processes of a small protocol specification formalism, namely CCS with probabilistic internal choice (CCS$_p$), which is a variant of the language proposed in (Chatzikokolakis *et al.*, 2007a; Chatzikokolakis *et al.*, 2007b). We show some application examples of our logic to the archetypical dining cryptographers (Chaum, 1988), and to oblivious transfer (Rabin, 1981) for single bits and entire strings. In both cases, we specify the protocol in CCS$_p$. The distinguishing feature of our logic is to provide a combination of *dynamic* operators for belief (whence the attribute "doxastic") with a *control* on the possible error of apprehension of the perceived reality, and for internalized probability. Both operators are dynamic (non-monotonic) thanks to the possibility of combining them with temporal operators, and are parameterized with a lower and upper probability bound (the error control).

Dynamicity is useful for the logical formalization of the original intuition of probabilistic anonymity and oblivious transfer, which is the one of an invariant w.r.t. *a priori* and *a posteriori* stances of apprehension of the perceived reality (cf. Sections 4 and 5.2). The belief operator is used to express that an agent a believes with *confidence* of at least l and at most u that a state of affairs ϕ is the case. The operator for internalized probability is used to express that a state of affairs ϕ is the case with *certainty* of at least l and at most u. Note that confidence is a qualification of an agent's belief (that something is the case), whereas certainty is just a qualification of something being the case: confidence has a subjective (belief) connotation whereas certainty has an objective (truth) connotation. In our framework, both qualifications are also quantitative thanks to the mentioned error control in terms of a lower and upper probability bound. A technical feature of our logic is that operators for belief and internalized probability can be nested, and that nested such operators can be flattened on certain, but different conditions.

The paper continues the previous work by some of the authors. In particular, in (Bhargava *et al.*, 2005) we have developed a probabilistic framework for anonymity. In (Chatzikokolakis *et al.*, 2009) we have refined that framework by using an information-theoretic approach, and in (Beauxis *et al.*, 2008) we have presented an overview of the various frameworks in literature for formalizing the notion of (absence of) leakage in information-hiding protocol. In (Chatzikokolakis *et al.*, 2007b; Chatzikokolakis *et al.*, 2007a) we have developed a probabilistic version of CCS and of the π-calculus respectively. In (Chatzikokolakis *et al.*, 2007a) we have used the latter to model the oblivious transfer, and we have proved its correctness by extending standard process-theoretic techniques.

In this paper we propose a logic for expressing properties based on belief, such as "the execution of the protocol does not increase the belief about the identity of the culprit" (anonymity), and "Alice believes with degree of confidence $1/2$ that Bob has received the bit 0" (a feature of the oblivious transfer). The advantages of having a logic, with respect to the approaches based on expressing the properties directly on the underlying formalism used to model the protocol, are the following: First, the logic is more high-level and it allows to reason about the properties more deeply, by highlighting their subtelties. Second, this way the properties are independent from the formalism used for representing the protocol. Third, the logic is more expressive, in particular thanks to the fact that allows distinguishing between subjective and objective uncertainty, i.e. between belief and probabilistic truth. Consider for instance the property of strong anonymity: In (Bhargava *et al.*, 2005) is is expressed as equality of the conditional probability of getting a certain observable under different culprits. Intuitively we intend such probabilities to represent the subjective uncertainty of the attacker, but, having only one form of probability, in that paper we cannot distinguish between belief or probabilistic truth. Here we are able to make such distinction, and we express strong anonymity in terms of belief (see Section 4). A similar consideration holds for the properties of the Oblivious Transfer analysed in (Chatzikokolakis *et al.*, 2007a), which are represented here by using both belief and probabilistic truth (see Section 5).

1.3. *Related work*

The literature on belief logics is large due to their wide applicability in philosophical logic, artificial intelligence, and information security. However, all belief logics we are aware of are either only about belief without error control, or *static* belief with error control.

In (Halpern *et al.*, 2005a), the authors introduce static belief with error control in the form of a functional symbol (term constructor) $\mathrm{Pr}_i(\phi)$ to be used in atomic formulas $\mathrm{Pr}_i(\phi) \leq \alpha$ that are true in a certain state by definition if and only if the probability that ϕ is true is at most α in that state. The probability value results from a probability measure applied to the set of all those states that are indistinguishable from the current state to agent i and where ϕ is true. The authors then obtain a formalization

of probabilistic anonymity for the dining cryptographers that mixes static knowledge (as a modality) and static belief. The logic is static, i.e. it does not have a temporal fragment. A fortiori, the belief in the logic is static (not possibly evolving). Also, the authors do not explicitly account for the possible presence of a scheduler, whereas we explicitly do.

In (Halpern *et al.*, 2005b), the authors introduce what they call *randomized*, *explicit* (or *algorithmic*) belief. The intuition is that

> a randomized knowledge algorithm returning "Yes" to a query about a fact ϕ provides evidence for ϕ being true.

The algorithm always returns either "Yes", "No", or "Don't know", and the return value "Yes" may depend on the outcome of coin tosses. The authors' motivation for the algorithmic modeling of belief is the resource-boundedness of real agents, which are thus identified with algorithms. The authors define measurable upper and lower *weights* that an answer given the knowledge algorithm, i.e. an observation, lends to a particular hypothesis in the set of hypotheses. Such a weight is not a probability measure, but rather "a prescription for how to update a prior probability on the hypotheses into a posterior probability on those hypotheses, after having considered the observations made". The goal of the authors of (Halpern *et al.*, 2005b) is "to understand what the evidence provided by a knowledge algorithm tells us.", which is quite a different than ours.

In (Lomuscio *et al.*, 2007), the authors present a formalization of non-probabilistic anonymity for the dining cryptographers expressed in a modal logic combining knowledge and time. Hence, their notion of knowledge is dynamic, yet not inflected with probability: it really is knowledge, which necessarily is true, and not belief, which possibly is false.

In (Dechesne *et al.*, 2007), the authors present a formalization of non-probabilistic anonymity for the dining cryptographers expressed in the μ-calculus with knowledge. Hence, the same comments apply as for (Lomuscio *et al.*, 2007). Additionally, their logic is, as ours, closely tied to a process calculus.

Internalized probability in our logic is based on the construct $[\phi]_p$ introduced in (Parma *et al.*, 2007) to represent probabilistic statements. A different probabilistic extension of Hennessy-Milner logic is the one of (Larsen *et al.*, 1991; Desharnais *et al.*, 1998), where they consider a probabilistic variant $\langle\!\langle a \rangle\!\rangle_p$ of the modal operator $\langle\!\langle a \rangle\!\rangle$. Intuitively, $\langle\!\langle a \rangle\!\rangle_p \phi$ means that a process can perform an a-transition and go *with probability at least* p to a state that satisfies ϕ. As showed in (Parma *et al.*, 2007), the operator $[\phi]_p$ is more expressive, because $\langle\!\langle a \rangle\!\rangle_p \phi$ can be represented as $\langle\!\langle a \rangle\!\rangle [\phi]_p$. Furthermore, Parma and Segala have shown that the operator $[\phi]_p$ is necessary for characterizing (probabilistic) bisimulation in systems that allow both probabilistic and non-deterministic branching from the same state, which turns out to be the case for CCS_p.

1.4. *Plan of the paper*

The next section recalls basic theory about probabilities, probabilistic automata, and probabilistic CCS. In Section 3, we introduce the *doxastic μ-calculus with error control* (DμCEC). Sections 4 and 5 are dedicated to the formalization and validation of probabilistic anonymity and oblivious transfer. Section 6 concludes the paper with an assessment of achievements and future work.

2. Preliminaries

In this section, we give a brief introduction to probability spaces, probabilistic automata (Segala, 1995; Segala *et al.*, 1995), and probabilistic CCS (Chatzikokolakis *et al.*, 2007b; Chatzikokolakis *et al.*, 2007a).

2.1. *Probability spaces*

Let Ω designate a set. A *σ-field* over Ω is a collection $\mathcal{F}$ of subsets of Ω closed under complement and countable union and such that $\Omega \in \mathcal{F}$. Ω is also called the *sample space* of the σ-field $\mathcal{F}$. If $\mathcal{B}$ is a collection of subsets of Ω then *the σ-field generated by $\mathcal{B}$* is defined as the smallest σ-field containing $\mathcal{B}$ (its existence is ensured by the fact that the intersection of an arbitrary set of σ-fields containing $\mathcal{B}$ is still a σ-field containing $\mathcal{B}$).

A *probability measure* on $\mathcal{F}$ is a function $\mu\colon \mathcal{F} \to [0, \infty]$ such that

1) $\mu(\emptyset) = 0$,
2) $\mu(\bigcup_i C_i) = \sum_i \mu(C_i)$ if $\{C_i\}_i$ is a countable collection of pairwise disjoint elements of $\mathcal{F}$, and
3) $\mu(\Omega) = 1$.

A *probability space* is a tuple $(\Omega, \mathcal{F}, \mu)$ where Ω is a set, called the *sample space*, $\mathcal{F}$ is a σ-field on Ω and μ is a probability measure on $\mathcal{F}$. The elements of a σ-field $\mathcal{F}$ are also called *events*.

For $x \in \Omega$, we denote by $\delta(x)$ (called the *Dirac measure* on x) the probability measure on $\mathcal{F}$ s.t. $\delta(x)(\{y\}) = 1$ if $y = x$, and $\delta(x)(\{y\}) = 0$ otherwise. If $\{c_i\}_i$ are convex coefficients (namely $c_i \geq 0$ for all i and $\sum_i c_i = 1$), and $\{\mu_i\}_i$ are probability measures, we denote by $\sum_i c_i\mu_i$ the probability measure defined as $(\sum_i c_i\mu_i)(A) \triangleq \sum_i c_i\mu_i(A)$. We denote by $supp(\mu) \triangleq \{x \in \Omega \mid \mu(x) > 0\}$ the *support set* of μ.

If A, B are events then $A \cap B$ is also an event. If $\mu(A) > 0$ then we can define the *conditional probability* $p(B \mid A)$, meaning "the probability of B given A", as

$$p(B \mid A) \triangleq \frac{\mu(A \cap B)}{\mu(A)}$$

Note that $p(\,\cdot \mid A)$ is a new probability measure on $\mathcal{F}$. In continuous probability spaces, where many events have zero probability, it is possible to generalize the concept of conditional probability to allow conditioning on such events. However, this is not necessary for the needs of this paper. Thus we will use the above "traditional" definition of conditional probability and make sure that we never condition on events of zero probability.

A probability space and the corresponding probability measure are called *discrete* if Ω is countable and $\mathcal{F} = 2^{\Omega}$. In this case, we can construct μ from a function $p\colon \Omega \to [0,1]$ satisfying $\sum_{x\in\Omega} p(x) = 1$ by assigning $\mu(\{x\}) = p(x)$. The set of all discrete probability measures with sample space Ω will be denoted by $Disc(\Omega)$.

2.2. *Probabilistic automata*

In this section we introduce the probabilist automata of (Segala *et al.*, 1995) following a notation that is similar to the one used in (Segala, 2006). To be closer to the CCS approach, we resolve the nondeterminism using a restricted class of schedulers: the Dirac non-halting schedulers. These schedulers choose a transition each time a transition is available. This restriction allows us to simplify the definition of the probability measures induced by the scheduler, since it reduces the sample space.

A *probabilistic automaton* M is a tuple $(St, s_{init}, Act, \mathcal{T})$ where St is a set of states, $s_{init} \in St$ is the *initial state*, Act is a set of actions and $\mathcal{T} \subseteq St \times Act \times Disc(St)$ is a *transition relation*. Intuitively, if $(s, a, \mu) \in \mathcal{T}$ then there is a transition from the state s performing the action a and leading to a distribution μ over the states of the automaton. The idea is that the choice of transition among the available ones in $\mathcal{T}$ is performed non-deterministically, and the choice of the target state among the ones allowed by μ (i.e. those states s' such that $\mu(s') > 0$) is performed probabilistically. Note that in general from a state there can be two transitions with the same action leading to two different distributions.

An *execution fragment* h of a probabilistic automaton is a (possibly infinite) alternating sequence $s_0 a_1 s_1 a_2 s_2 \ldots$ of states and actions, such that for each i there is a transition $(s_i, a_{i+1}, \mu_i) \in \mathcal{T}$ and $\mu_i(s_{i+1}) > 0$. The concatenation of a finite execution fragment $h_1 = s_0 \ldots a_n s_n$ and an execution fragment $h_2 = s_n a_{n+1} s_{n+1} \ldots$ is the execution fragment $h_1 \cdot h_2 = s_0 \ldots a_n s_n a_{n+1} s_{n+1} \ldots$. A finite execution fragment h_1 is a prefix of h, written $h_1 \leq h$, if there is an execution fragment h_2 such that $h = h_1 \cdot h_2$. We use $fst(h)$, $lst(h)$ to denote the first and last state of a finite execution fragment h respectively. An *execution* (or *history*) h is an execution fragment such that $fst(h) = s_{init}$. An execution h is maximal if it is infinite or there is no transition from $lst(h)$ in $\mathcal{T}$. We denote by $exec^*(M)$, $exec^{\perp}(M)$, and $exec(M)$ the set of all the finite, all the non maximal, and all the executions of M, respectively.

A *scheduler* for a probabilistic automaton $M = (St, s_{init}, Act, \mathcal{T})$ is a total function

$$\zeta\colon exec^{\perp}(M) \to \mathcal{T}$$

such that $\zeta(h) = (s, a, \mu) \in \mathcal{T}$ implies that $s = lst(h)$. The idea is that when we are in state s the scheduler selects a transition among the ones available in $\mathcal{T}$ for s, and it can base its decision on the history of the execution that has lead to s. As anticipated, we depart here from the traditional definition of probabilistic automata. First, we impose totality to be in line with the CCS concept of scheduler, while in general the scheduler may decide to stop even if a transition is available; second we impose that a scheduler does not use randomization in resolving nondeterminism, while in general a scheduler may be randomized.

The *execution tree* of M under the scheduler ζ, denoted by $etree(M, \zeta)$, is a fully probabilistic automaton $M' = (St', s_{init}, Act, \mathcal{T}')$ such that $St' \subseteq exec^*(M)$, and $(h, a, \mu') \in \mathcal{T}'$ if and only if $\zeta(h) = (lst(h), a, \mu)$ for some μ, and $\mu'(has) = \mu(s)$. Intuitively, $etree(M, \zeta)$ is produced by unfolding the executions of M and resolving all non-deterministic choices using ζ.

Given a probabilistic automaton $M = (St, s_{init}, Act, \mathcal{T})$ and a scheduler ζ we can define the probability space $(\Omega_M, \mathcal{F}_M, p_M)$ on the maximal executions of M induced by ζ as follows:

– $\Omega_M \triangleq exec(M) \setminus exec^{\perp}(M)$ (the set of all the maximal executions of M).

– Given a finite execution h, the cone with prefix h is defined as $C_h \triangleq \{h' \in \Omega_M | h \leq h'\}$. Define $\mathcal{F}$ as the σ-field generated by the set of all cones of M.

– Define the probability of a cone C_h, where $h = s_0 a_1 s_1 \ldots a_n s_n$, as

$$p(C_h) \triangleq \prod_{i=1}^{n} \mu_i(s_i)$$

where, for each i, $\zeta(s_0 a_1 s_1 \ldots a_{i-1} s_{i-1}) = (s_{i-1}, a_i, \mu_i)$. We define $p_{M,\zeta}$ as the measure extending p to $\mathcal{F}$ (see (Segala, 1995) for more details).

REMARK 1. — The σ-field used in (Segala *et al.*, 1995) considers the sample space $\Omega = exec(M)$ to account for the termination at non-maximal executions. Since here we require that the schedulers are total, the support of the measure $p_{M,\zeta}$ does not need to include elements of $exec^{\perp}(M)$. Note that the σ-field defined in this paper coincides with the sub-σ-field not containing $exec^{\perp}(M)$ of the standard σ-field on probabilistic-automata induced by total schedulers. □

Convention Given a probabilistic automaton M and a scheduler ζ, we denote $p_{M,\zeta}$ by p_ζ whenever M is clear from the context.

2.3. *CCS with probabilistic internal choice*

In this section we introduce the probabilistic CCS CCS_p of (Chatzikokolakis *et al.*, 2007b) as well as its operational semantics in terms of probabilistic automata. We assume that the number of channel names is finite[1]. This restriction allows us to

1. Usually it is assumed to be at most countable, so it can be either finite or infinite.

express certain operators as syntactic sugar, notably the operators $\diamondsuit$, $\overleftarrow{\diamondsuit}$, and $\boxdot$ of Table 3, thus simplifying the theory. Note that this is not really a restriction in the context of this paper, because we are interested in analysing properties of programs, that, being finite syntactic entities, can only contain a fixed number of channel names.

Let A be a set of *channel names*, which, for the purposes of this paper, we assume to be finite. Each $a \in A$ represents the action of making an input from the channel a, and the corresponding output action on a is denoted by $\bar{a}$. The actions a and $\bar{a}$ are seen as *complementary*, and we require $\bar{\bar{a}} = a$. Let $\bar{A} = \{\bar{a} \mid a \in A\}$ and let $\tau \notin A \cup \bar{A}$ represent the *invisible action*. The set $Act = A \cup \bar{A} \cup \{\tau\}$ is the set of *actions*, which we represent by $a, b, \ldots$. We assume that τ has no complement, hence the notation $\bar{a}$ implies that $a \neq \tau$.

The syntax of CCS$_p$ is the following:

P, Q	::=		**processes**
		$a \,.\, P$	action prefix
	$\mid$	$\sum_i p_i P_i$	probabilistic internal choice
	$\mid$	$P \mid Q$	parallel
	$\mid$	$P + Q$	non-deterministic internal choice
	$\mid$	$(\nu a)P$	restriction
	$\mid$	$!P$	replication
	$\mid$	0	nil

We also use the notation $P_1 \oplus_p P_2$ to represent a binary probabilistic choice $\sum_i p_i P_i$ with $p_1 = p$ and $p_2 = 1 - p$.

The semantics of a CCS$_p$ term is a probabilistic automaton whose states are CCS$_p$ processes, and whose transitions are defined inductively on the basis of the syntax according to the rules in Table 1. We write $s \xrightarrow{a} \mu$ to indicate that (s, a, μ) is a transition of the probabilistic automaton. We also denote by $\mu \mid Q$ the measure μ' such that $\mu'(P \mid Q) = \mu(P)$ for all processes P and $\mu'(R) = 0$ if R is not of the form $P \mid Q$. Similarly $(\nu a)\mu = \mu'$ such that $\mu'((\nu a)P) = \mu(P)$ and $\mu'(P') = 0$ if P' is not of the form $(\nu a)P$.

The semantics of CCS$_p$ is a conservative extension of the one of CCS, in the sense that there is a one-to-one correspondence between the rules of CCS$_p$, except PROB, and those of CCS. The correspondence is evident after observing that a transition of the form $P \xrightarrow{a} \delta(P')$, i.e. a transition having for target a Dirac measure, corresponds to a transition $P \xrightarrow{a} P'$ in a standard labeled transition system.

We explain briefly the meaning of the rules: ACT represents the execution of the action a in $a.P$. PAR1 and the omitted rule PAR2 represent the fact that in $P \mid Q$, the process P (resp. Q) can execute a step while Q (resp. P) stays idle (interleaving). COM represents a communication step between P and Q in $P \mid Q$, which can take place when P and Q are ready to perform complementary actions. SUM1 and the

Table 1. *The semantics of CCS$_p$. PAR1 and SUM1 have corresponding right rules PAR2 and SUM2, omitted for simplicity.*

ACT	$\dfrac{}{a\,.\,P \xrightarrow{a} \delta(P)}$	PROB	$\dfrac{}{\sum_i p_i P_i \xrightarrow{\tau} \sum_i p_i \delta(P_i)}$
PAR1	$\dfrac{P \xrightarrow{a} \mu}{P \mid Q \xrightarrow{a} \mu \mid Q}$	COM	$\dfrac{P \xrightarrow{a} \delta(P') \quad Q \xrightarrow{\overline{a}} \delta(Q')}{P \mid Q \xrightarrow{\tau} \delta(P' \mid Q')}$
SUM1	$\dfrac{P \xrightarrow{a} \mu}{P + Q \xrightarrow{a} \mu}$	RES	$\dfrac{P \xrightarrow{b} \mu \qquad b \neq a, \overline{a}}{(\nu a)P \xrightarrow{b} (\nu a)\mu}$
REP1	$\dfrac{P \xrightarrow{a} \mu}{!P \xrightarrow{a} \mu \mid !P}$	REP2	$\dfrac{P \xrightarrow{a} \delta(P_1) \quad P \xrightarrow{\overline{a}} \delta(P_2)}{!P \xrightarrow{\tau} \delta(P_1 \mid P_2 \mid !P)}$

omitted rule SUM2 represent the commitment of the nondeterministic choice $P + Q$ to P (resp. Q) branch, provided that it is not suspended, i.e. it can make a step. RES filters out the transitions with label a or $\bar{a}$ from a process restricted on a. REP1 and REP2 express the fact that $!P$ can spawn one (resp two) copies of P and let these copies perform a step.

Finally, PROB models probabilistic internal choice: a silent τ transition is available from the sum to a measure containing all of its operands, with the corresponding probabilities. Note that in the produced probabilistic automaton, all transitions to non-Dirac measures are silent, because they are originated from PROB. Note also that a probabilistic term generates exactly one (probabilistic) transition.

It is interesting to observe that the resulting automaton is consistent with the definition of alternating automaton of (Parma *et al.*, 2007): a probabilistic automaton is *alternating* if the states that enable a non-Dirac transition enable only one transition. We call *probabilistic* those states that enable non-Dirac transitions, and *nondeterministic* all the other states. In other words, a probabilistic state enables at most one transition while a nondeterministic state may enable several transitions with the constrait that the target measure of each of these transitions is a Dirac measure.

3. A quantitative doxastic logic and its interpretation in CCS$_p$

In this section we propose an extension DμCEC of Nielsen's μ*-calculus with past* (Nielsen, 1998) suitable for expressing information-hiding properties and for reasoning about protocols for privacy.

Recall that the μ-calculus has modal operators $\langle a \rangle$ that express the *future capabilities* of a process: the formula $\langle a \rangle \phi$ means that a process can perform the action a and evolve into a new process that satisfies ϕ. In addition to these, Nielsen's calculus contains also their *past* counterparts $\overleftarrow{\langle a \rangle}$: the formula $\overleftarrow{\langle a \rangle} \phi$ means that the process is the outcome of an a-transition from another process which satisfies ϕ.

We extend Nielsen's calculus in two ways:

– We internalize probabilistic truth in the form of probabilistic statements which are based on Parma and Segala's probabilistic extension (Parma *et al.*, 2007) of Hennessy-Milner logic. They consider constructs like $[\phi]_p$, where p is a parameter representing a probability. Formulas are interpreted on probability measures, and the meaning of $[\phi]_p$ is that the set of states that satisfy ϕ has probability *at least* p with respect to the given measure. We actually consider constructs like $\mathcal{P}_p^q(\phi)$, meaning that the set of states that satisfy ϕ has probability *at least* p *and at most* q. The operator $\mathcal{P}_p^q(\phi)$ could be approximated by $[\phi]_p \wedge \neg[\phi]_{q+\epsilon}$, but we prefer to have the former as a primitive because in some example we need exact probabilities.

– We add belief in the form of doxastic operators ${}_i\mathcal{B}_p^q$ which represent the degree of confidence of agents about the truth of formulas in DμCEC. Intuitively the formula ${}_i\mathcal{B}_p^q(\phi)$ means that the agent i estimates that there is a probability *at least* p *and at most* q that the process satisfies ϕ. Note that in general ${}_i\mathcal{B}_p^q(\mathcal{P}_1^1(\phi))$ and ${}_i\mathcal{B}_1^1(\mathcal{P}_p^q(\phi))$ are not equivalent, and neither are $\mathcal{P}_1^1({}_i\mathcal{B}_p^q(\phi))$ and $\mathcal{P}_p^q({}_i\mathcal{B}_1^1(\phi))$.

The syntax of DμCEC is given by the following grammar, where p and q are constant between 0 and 1:

$$\phi ::= \top \mid X \mid \phi \wedge \phi \mid \neg\phi \mid \langle a \rangle \phi \mid \overleftarrow{\langle a \rangle} \phi \mid \mathcal{P}_p^q(\phi) \mid {}_i\mathcal{B}_p^q(\phi) \mid \mathit{lfp}_X \phi(X)$$

where in the *least fixpoint formula* $\mathit{lfp}_X \phi(X)$, the variable X is assumed to occur positively in ϕ. To define formally what it means that X occurs positively in a formula, we use the standard notion of *context* $C[\ ]$ and define the concepts of *positive* and *negative* context as follows.

DEFINITION 2. — *For a context $C[\ ]$, the properties of being positive and being negative are defined inductively as follows:*

$[\ ]$	*is*	*positive*	
$C[\ ] \wedge \phi$	*is*	*positive*	*if $C[\ ]$ is positive*
$\phi \wedge C[\ ]$	*is*	*positive*	*if $C[\ ]$ is positive*
$pop(C[\ ])$	*is*	*positive*	*if $C[\ ]$ is positive with $pop = \langle a \rangle, \overleftarrow{\langle a \rangle}, \mathcal{P}_p^1$, or ${}_i\mathcal{B}_p^1$*
$nop(C[\ ])$	*is*	*positive*	*if $C[\ ]$ is negative with $nop = \neg, \mathcal{P}_0^p$, or ${}_i\mathcal{B}_0^p$*
$\mathit{lfp}_X C[X]$	*is*	*positive*	*($C[\ ]$ must be positive)*

and

$C[\,] \wedge \phi$	*is*	*negative*	*if* $C[\,]$ *is negative*
$\phi \wedge C[\,]$	*is*	*negative*	*if* $C[\,]$ *is negative*
$pop(C[\,])$	*is*	*negative*	*if* $C[\,]$ *is negative with* $pop = \diamondsuit_a, \overleftarrow{\diamondsuit}_a, \mathcal{P}^1_p$, *or* ${}_i\mathcal{B}^1_p$
$nop(C[\,])$	*is*	*negative*	*if* $C[\,]$ *is positive with* $nop = \neg, \mathcal{P}^p_0$, *or* ${}_i\mathcal{B}^p_0$

DEFINITION 3. — *X occurs positively in ϕ if $\phi = C[X]$ for some positive context $C[\,]$.*

REMARK 4. — Informally, a variable X occurs positively if it is not within the scope of an odd numer of negations, that is if we use the actual value of X and not its negations. There exists another implicit kind of negation that occurs when we impose an upper bound to the probability of a positive occurrence of X or when we impose a lower bound to the probability of a negative occurrence of X. This leads to the relative definition of operators $\mathcal{P}^q_p$. □

We want to use this logic to express properties of processes written in CCS$_p$, hence we are now going to provide an interpretation of the logic in CCS$_p$, i.e. we define a *satisfaction relation* $\models$ between CCS$_p$ and DμCEC.

In standard Hennessy-Milner logic, and in μ-calculus, satisfaction is usually defined with respect to processes. Here we need to interpret the doxastic operators ${}_i\mathcal{B}^q_p$, and for this purpose we must consider not just the current process, but the whole (finite) history of the execution, because of the dynamic nature of our notion of belief. Furthermore, as explained before, in order to interpret the formulas $\mathcal{P}^q_p(\phi)$ we need to consider probabilistic measures. In conclusion, we are going to take as domain the set of discrete distributions $Disc(H)$, where H is the set of finite histories generated by CCS$_p$.

Given a history h, an action a, and a probability distribution on states μ, we denote by $ha\mu$ the extension of μ to the histories of the form haP, for every CCS$_p$ process P. Namely:

$$(ha\mu)(h') \triangleq \begin{cases} \mu(P) & \text{if } h' = haP \\ 0 & \text{otherwise} \end{cases}$$

The interpretation of the operators ${}_i\mathcal{B}^q_p$ is based on an epistemic accessibility relation $\equiv_i$ on finite histories. Intuitively $h_1 \equiv_i h_2$ represents the fact that the histories h_1 and h_2 are indistinguishable to i. $\equiv_i$ is usually chosen to be an equivalence relation as induced by the local view of i. We assume that the local view is only restricted to actions, hence we consider the projection of histories on actions (*traces*). Intuitively, the trace of h is the string of the actions in h, i.e. what is left in h after we remove all the states. More formally:

DEFINITION 5. — *Given a finite history h, the trace of h is defined inductively as follows:*

– $trace(P) = \epsilon$ *(the empty trace)*
– $trace(haP) = trace(h)a$

We assume that in general an agent has only a *partial view* on actions. Formally, this can be represented by introducing the following abstraction function:

ASSUMPTION 6. — For every agent i we assume a function $f_i \colon A \to A \cup \{\epsilon\}$ which represents i's *view* on actions. □

We can now define the accessibility relation on traces and histories. We use for simplicity the same symbol $\equiv_i$ to denote both relations.

DEFINITION 7. —

– *For every agent i, the relation $\equiv_i$ on traces is defined inductively as follows:*
  - $\epsilon \equiv_i \epsilon$
  - $ta \equiv_i t'b$ *if either* $f_i(a) = f_i(b)$ *and* $t \equiv_i t'$

    or

    $f_i(a) = \epsilon$ *and* $t \equiv_i t' f_i(b)$

    or

    $f_i(b) = \epsilon$ *and* $t f_i(a) \equiv_i t'$
– *For every agent i, the relation $\equiv_i$ on histories is defined as follows:*

$$h \equiv_i h' \text{ if and only if } trace(h) \equiv_i trace(h')$$

We can now define the interpretation of DμCEC with respect to the process terms of CCS$_p$. We only consider the closed formulas of DμCEC, namely only the formulas in which all variable occurrences are bound.

DEFINITION 8. — *The relation $\models$ on $Disc(H)$ and on the closed formula of DμCEC is defined according to the clauses in Table 2. In the table, $[\![\ ]\!]$ is defined as $[\![\phi]\!] \triangleq \{h \mid \delta(h) \models \phi\}$, while p_ζ represents the probability measure on $etree(P, \zeta)$ (see section 2.2), and $[h]_{\equiv_i}$ is the equivalence class of h with respect to $\equiv_i$. Finally, if H is a set of executions, $\downarrow H$ represents the maximal executions with prefix in H, i.e. $\downarrow H \triangleq \{h \in \Omega_P | \exists h' \in H. h' \leq h\}$*

In the definition of $\mu \models {}_i\mathcal{B}_p^q(\phi)$, the idea is that the probability that the process satisfies ϕ *given any h' indistinguishable from h in i's view* is between p and q. We quantify over all possible schedulers because in general i does not know what is the scheduler, except for the partial view it has on h.

The auxiliary "hybrid formulas" D_X ("auxiliary" because they do not exist in the syntax of the language, and "hybrid" because X represents a set of executions) are introduced to define the semantics of lfp_X.

Table 2. *Definition of satisfaction for the closed formulas in DμCEC.*

$\mu \models \top$		
$\mu \models \phi_1 \wedge \phi_2$	:iff	$\mu \models \phi_1$ and $\mu \models \phi_2$
$\mu \models \neg\phi$	:iff	$\mu \not\models \phi$
$\mu \models \langle\!\langle a \rangle\!\rangle \phi$	:iff	for every $h \in supp(\mu)$ there exists η and a transition $lst(h) \xrightarrow{a} \eta$ such that $ha\eta \models \phi$
$\mu \models \overleftarrow{\langle\!\langle a \rangle\!\rangle} \phi$	:iff	there exists h' such that $lst(h') \xrightarrow{a} \mu$ and $\delta(h') \models \phi$
$\mu \models \mathcal{P}_p^q(\phi)$	:iff	$p \leq \mu([\![\phi]\!]) \leq q$
$\mu \models {}_i\mathcal{B}_p^q(\phi)$	:iff	for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(h)$ we have $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$
$\mu \models lfp_X \phi(X)$	:iff	$\mu \in \bigcap\{D_X \subseteq Disc(H) \mid \forall \eta \in Disc(H)$ if $\eta \models \phi(X := D_X)$ then $\eta \models D_X\}$
$\mu \models D_X$	:iff	$\mu \in D_X$
$P \models \phi$	:iff	$\delta(P) \models \phi$

The semantic correspondent of $lfp_X \phi(X)$ (i.e. the set of distributions that satisfy $lfp_X \phi(X)$) is the *least fixed point* of a transformation $\mathcal{T}_\phi \colon 2^{Disc(H)} \to 2^{Disc(H)}$ defined as follows:

$$\mathcal{T}_\phi(D) \triangleq \{\mu \mid \mu \models \phi(X := D)\}$$

It can be proved that if X occurs positively in $\phi(X)$ then $\mathcal{T}_\phi$ is monotonic on the lattice $(2^{Disc(H)}, \subseteq)$ which, by the Theorem of Knaster-Tarski, implies the existence of the least and greatest fixed points. The core of the proof is Theorem 10 below.

DEFINITION 9 (MONOTONICITY). — *For any formula ϕ in DμCEC, let $\{\!|\phi|\!\}$ denote the set $\{\mu \mid \mu \models \phi\}$. An n-ary operator op in DμCEC is* monotonic[2] *if for all $\phi_1, \ldots, \phi_n$, $\psi_1, \ldots, \psi_n$, we have that $\{\!|\phi_1|\!\} \subseteq \{\!|\psi_1|\!\}, \ldots, \{\!|\phi_n|\!\} \subseteq \{\!|\psi_n|\!\}$ implies $\{\!|op(\phi_1, \ldots, \phi_n)|\!\} \subseteq \{\!|op(\psi_1, \ldots, \psi_n)|\!\}$. It is* antimonotonic *if for all $\phi_1, \ldots, \phi_n$, $\psi_1, \ldots, \psi_n$, we have that $\{\!|\phi_1|\!\} \subseteq \{\!|\psi_1|\!\}, \ldots, \{\!|\phi_n|\!\} \subseteq \{\!|\psi_n|\!\}$ implies $\{\!|op(\psi_1, \ldots, \psi_n)|\!\} \subseteq \{\!|op(\phi_1, \ldots, \phi_n)|\!\}$.*

THEOREM 10. — *The operators $\wedge$, $\langle\!\langle a \rangle\!\rangle$, $\overleftarrow{\langle\!\langle a \rangle\!\rangle}$, $\mathcal{P}_p^1$, ${}_i\mathcal{B}_p^1$ and lfp_X are monotonic. The operators $\neg$, $\mathcal{P}_0^p$ and ${}_i\mathcal{B}_0^p$ are antimonotonic.*

PROOF. — The proof proceeds by case analysis. We consider here only the operators that are used in this paper, i.e. those which appear in the scope of a *lfp* or *gfp* operator

2. Not to be confused with monotonicity and non-monotonicity of belief with respect to time.

in Table 3. In the following, we assume $\{\!\{\phi\}\!\} \subseteq \{\!\{\psi\}\!\}$, $\{\!\{\phi_1\}\!\} \subseteq \{\!\{\psi_1\}\!\}$, and $\{\!\{\phi_2\}\!\} \subseteq \{\!\{\psi_2\}\!\}$.

$\wedge$) Let $\mu \models \phi_1 \wedge \phi_2$. Then $\mu \models \phi_1$ and $\mu \models \phi_2$. Since $\{\!\{\phi_1\}\!\} \subseteq \{\!\{\psi_1\}\!\}$ and $\{\!\{\phi_2\}\!\} \subseteq \{\!\{\psi_2\}\!\}$, we have that $\mu \models \psi_1$ and $\mu \models \psi_2$. Hence $\mu \models \psi_1 \wedge \psi_2$.

$\neg$) Let $\mu \models \neg\psi$. Then $\mu \not\models \psi$ Since $\{\!\{\phi\}\!\} \subseteq \{\!\{\psi\}\!\}$, we have that $\mu \not\models \phi$. Hence $\mu \models \neg\phi 2$.

$\langle\!\langle a \rangle\!\rangle$) Let $\mu \models \langle\!\langle a \rangle\!\rangle \phi$. Then for every $h \in \mathit{supp}(\mu)$ there exists η and a transition $\mathit{lst}(h) \xrightarrow{a} \eta$ such that $ha\eta \models \phi$. Since $\{\!\{\phi\}\!\} \subseteq \{\!\{\psi\}\!\}$, we have that $ha\eta \models \psi$. Hence $\mu \models \langle\!\langle a \rangle\!\rangle \psi$.

$\overleftarrow{\langle\!\langle a \rangle\!\rangle}$) Let $\mu \models \overleftarrow{\langle\!\langle a \rangle\!\rangle} \phi$. Then for every $h \in \mathit{supp}(\mu)$ there exists h' such that $\mathit{lst}(h') \xrightarrow{a} \mu$ and $\delta(h') \models \phi$. Since $\{\!\{\phi\}\!\} \subseteq \{\!\{\psi\}\!\}$, we have $\delta(h') \models \psi$. Hence $\mu \models \overleftarrow{\langle\!\langle a \rangle\!\rangle} \psi$.

${}_i\mathcal{B}^1_p$) Let $\mu \models {}_i\mathcal{B}^1_p(\phi)$. Then, for every $h \in \mathit{supp}(\mu)$ and for every scheduler ζ for $\mathit{fst}(\mu)$ we have $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq 1$. Since $\{\!\{\phi\}\!\} \subseteq \{\!\{\psi\}\!\}$, we have $p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i})$. Hence $p \leq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) \leq 1$, and therefore $\mu \models {}_i\mathcal{B}^1_p(\psi)$.

■

COROLLARY 11. — *If X occurs positively in $\phi(X)$ then $\mathcal{T}_\phi$ is monotonic on the lattice $(2^{Disc(H)}, \subseteq)$.*

COROLLARY 12. — *If X occurs positively in $\phi(X)$ then the set of fixed points of $\mathcal{T}_\phi$ forms a sublattice of $(2^{Disc(H)}, \subseteq)$. In particular, there exists a least and a greatest fixed point.*

3.1. *Relation with standard (KD45) belief*

In this section we discuss the relation of DμCEC with standard (KD45) belief.

Our operators ${}_i\mathcal{B}^q_p$ and $\mathcal{P}^q_p$ satisfy probabilistic extensions of the axioms of standard belief and truth, in the sense expressed by Theorems 13 and 15 below. In the following, the operator $\rightarrow$ stands for Boolean (material) implication, see Table 3, and $\models \phi$ means that $\mu \models \phi$ holds for all μ.

THEOREM 13. — *For any $p, q, r, s \in [0, 1]$, any formulas ϕ, ψ in DμCEC, and any agent i, the following hold.*

K) $\models {}_i\mathcal{B}^q_p(\phi \rightarrow \psi) \rightarrow ({}_i\mathcal{B}^s_r(\phi) \rightarrow {}_i\mathcal{B}^q_t(\psi))$ *where* $t = \max\{0, p + r - 1\}$

D) $\models {}_i\mathcal{B}^0_0(\bot)$ *("i does not believe false")*

4) $\models {}_i\mathcal{B}^q_p(\phi) \rightarrow {}_i\mathcal{B}^1_r({}_i\mathcal{B}^q_p(\phi))$

5) $\models \neg_i \mathcal{B}^q_p(\phi) \to {}_i\mathcal{B}^1_r(\neg_i \mathcal{B}^q_p(\phi))$

PROOF. —

K) Assume $\mu \models {}_i\mathcal{B}^q_p(\phi \to \psi)$ and $\mu \models {}_i\mathcal{B}^s_r(\phi)$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(\mu)$ we have $p \leq p_\zeta(\downarrow [\![\phi \to \psi]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$ and $r \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq s$. Let $v = p_\zeta(\downarrow [\![\phi \to \psi]\!] \mid \downarrow [h]_{\equiv_i})$ and $w = p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i})$. Observe that $p_\zeta(\downarrow [\![\phi \to \psi]\!] \mid \downarrow [h]_{\equiv_i}) = p_\zeta(\downarrow [\![\phi \wedge \psi]\!] \mid \downarrow [h]_{\equiv_i}) + p_\zeta(\downarrow [\![\neg\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) + p_\zeta(\downarrow [\![\neg\phi]\!] \mid \downarrow [h]_{\equiv_i})$. From this we obtain $p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) \geq v - (1 - w) \geq p + r - 1$. Hence $p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) \geq \max\{0, p + r - 1\}$.

On the other side, observe that we have $p_\zeta(\downarrow [\![\phi \to \psi]\!] \mid \downarrow [h]_{\equiv_i}) = p_\zeta(\downarrow [\![\phi \wedge \psi]\!] \mid \downarrow [h]_{\equiv_i}) + p_\zeta(\downarrow [\![\neg\phi]\!] \mid \downarrow [h]_{\equiv_i}) = p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) + p_\zeta(\downarrow [\![\neg\phi]\!] \mid \downarrow [h]_{\equiv_i}) - p_\zeta(\downarrow [\![\psi \wedge \neg\phi]\!] \mid \downarrow [h]_{\equiv_i}) \geq p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i})$. Hence we obtain $p_\zeta(\downarrow [\![\psi]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$.

D) This statement follows immedialy from the observation that for every scheduler ζ and every history h we have $p_\zeta(\downarrow [\![\bot]\!] \mid \downarrow [h]_{\equiv_i}) = p_\zeta(\downarrow [\![\bot]\!]) = 0$.

4) Assume $\mu \models {}_i\mathcal{B}^q_p(\phi)$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(\mu)$ we have $p \leq p_\zeta(\downarrow [\![\phi]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$. Hence, for every $h \in supp(\mu)$ we have $\delta(h) \models {}_i\mathcal{B}^q_p(\phi)$, from which we derive that, for every scheduler ζ for $fst(\mu)$, $p_\zeta(\downarrow [\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow [h]_{\equiv_i}) = 1$ holds. Therefore $\mu \models {}_i\mathcal{B}^1_r({}_i\mathcal{B}^q_p(\phi))$.

5) Similar to the proof of **(4)**.

■

For **(4)** and **(5)**, when $r = 1$ the implication holds also in the other direction, which means that belief can be "flattened" for certain probabilities.

PROPOSITION 14. — *For every agent i and every formula ϕ, the following hold*

- $\models {}_i\mathcal{B}^1_1({}_i\mathcal{B}^q_p(\phi)) \to {}_i\mathcal{B}^q_p(\phi)$
- $\models {}_i\mathcal{B}^1_1(\neg_i\mathcal{B}^q_p(\phi)) \to \neg_i\mathcal{B}^q_p(\phi)$

PROOF. — Let us consider the first statement. Assume $\mu \models {}_i\mathcal{B}^1_1({}_i\mathcal{B}^q_p(\phi))$. Then, for every $h \in supp(\mu)$ and for every scheduler ζ for $fst(\mu)$ we have $p_\zeta(\downarrow [\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow [h]_{\equiv_i}) = 1$. Hence, for every $h \in supp(\mu)$ we have $\delta(h) \models {}_i\mathcal{B}^q_p(\phi)$, from which we derive that, for every scheduler ζ for $fst(\mu)$, $p \leq p_\zeta(\downarrow [\![{}_i\mathcal{B}^q_p(\phi)]\!] \mid \downarrow [h]_{\equiv_i}) \leq q$ holds. Therefore $\mu \models {}_i\mathcal{B}^q_p(\phi)$.

The proof of the second statement is similar. ■

For probabilistic truth we have the following

THEOREM 15. — *For any $p, q, r, s \in [0, 1]$, any formulas ϕ, ψ in DμCEC, and any agent i, the following hold.*

K) $\models \mathcal{P}_p^q(\phi \to \psi) \to (\mathcal{P}_r^s(\phi) \to \mathcal{P}_t^q(\psi))$ *where* $t = \max\{0, p + r - 1\}$

D) $\models \mathcal{P}_0^0(\bot)$

4) $\models \mathcal{P}_p^q(\phi) \to \mathcal{P}_p^q(\mathcal{P}_r^1(\phi))$ *if* $r > 0$

5) $\models \neg\mathcal{P}_p^q(\phi) \to \mathcal{P}_p^q(\neg\mathcal{P}_0^r(\phi))$ *if* $r < 1$

PROOF. —

K) Similar to the proof of **(K)** in Theorem 13.

D) Similar to the proof of **(D)** in Theorem 13.

4) Assume $\mu \models \mathcal{P}_p^q(\phi)$. Then $p \leq \mu([\![\phi]\!]) \leq q$. Observe that $h \in [\![\phi]\!]$ if and only if $\delta(h)([\![\phi]\!]) = 1$, or equivalently, for $r > 0$, $r \leq \delta(h)([\![\phi]\!]) \leq 1$. By definition this is equivalent to $\delta(h) \models \mathcal{P}_r^1(\phi)$, which holds if and only if $h \in [\![\mathcal{P}_r^1(\phi)]\!]$. Therefore $p \leq \mu([\![\mathcal{P}_r^1(\phi)]\!]) \leq q$.

5) Similar to the proof of **(4)**.

■

For **(4)** and **(5)**, the implication holds also in the other direction, meaning that also the probabilistic truth can be "flattened" for certain probabilities.

PROPOSITION 16. — *For every agent i and every formula ϕ, the following hold*

- $\models \mathcal{P}_p^q(\mathcal{P}_r^1(\phi)) \to \mathcal{P}_p^q(\phi)$ *if* $r > 0$.
- $\models \mathcal{P}_p^q(\neg\mathcal{P}_0^r(\phi)) \to \neg\mathcal{P}_p^q(\phi)$ *if* $r < 1$.

PROOF. — Let us consider the first statement. Assume $\mu \models \mathcal{P}_p^q(\mathcal{P}_r^1(\phi))$. Then $p \leq \mu([\![\mathcal{P}_r^1(\phi)]\!]) \leq q$. Following the same reasoning as in the proof of Theorem 15 **(4)**, we have that (for $r > 0$) $h \in [\![\mathcal{P}_r^1(\phi)]\!]$ if and only if $h \in [\![\phi]\!]$. Therefore $\mu \models \mathcal{P}_p^q(\phi)$.

The proof of the second statement is similar. ■

Finally, we want to point out that the following formulas hold, meaning that our belief operators behave well with respect to probability measures. The proof is immediate.

PROPOSITION 17. — *For every agent i and every formula ϕ, the following hold*

- $\models {}_i\mathcal{B}_p^q(\phi) \leftrightarrow {}_i\mathcal{B}_{1-q}^{1-p}(\neg\phi)$
- $\models \mathcal{P}_p^q(\phi) \leftrightarrow \mathcal{P}_{1-q}^{1-p}(\neg\phi)$

We conclude this section by giving the definition of some derived operators in DμCEC. They are illustrated in Table 3.

Table 3. *Some derived operators in DμCEC.*

$\bot$	$\triangleq \neg\top$	false
$\phi_1 \vee \phi_2$	$\triangleq \neg(\neg\phi_1 \wedge \neg\phi_2)$	Boolean disjunction
$\phi_1 \rightarrow \phi_2$	$\triangleq \neg\phi_1 \vee \phi_2$	Boolean (material) implication
$\boxed{a}\phi$	$\triangleq \neg\langle a\rangle\neg\phi$	after every a-transitions ϕ holds
$gfp_X\phi(X)$	$\triangleq \neg lfp\neg\phi(\neg X)$	greatest fixed point of $\lambda X.\phi(X)$. The variable X is assumed to occur positively in ϕ. Note that this implies that also $\neg X$ occurs positively in $\neg\phi$
$\langle a\rangle_p\phi$	$\triangleq \langle a\rangle\mathcal{P}^1_p(\phi)$	there is an a-transition after which ϕ holds with probability at least p
$\langle\cdot\rangle\phi$	$\triangleq \bigvee_{a\in Act}\langle a\rangle\phi$	there is a transition after which ϕ holds
$\overleftarrow{\langle\cdot\rangle}\phi$	$\triangleq \bigvee_{a\in Act}\overleftarrow{\langle a\rangle}\phi$	there is a transition before which ϕ holds
$\boxdot\phi$	$\triangleq \bigwedge_{a\in Act}\boxed{a}\phi$	after all transitions ϕ holds
$\langle a\rangle^*$	$\triangleq lfp_X.\langle a\rangle\top \vee \langle\cdot\rangle X$	it is possible to reach a state which has an a-transition
$\overleftarrow{\langle a\rangle}^*$	$\triangleq lfp_X.\overleftarrow{\langle a\rangle}\top \vee \overleftarrow{\langle\cdot\rangle}X$	there has been an a-transition in the past
$\boxdot^*\phi$	$\triangleq lfp_X.\phi \wedge \boxdot X$	ϕ holds now and at all points in all the possible futures
${}_I\mathcal{CB}^1_p\phi$	$\triangleq gfp_X(\bigwedge_{i\in I}{}_i\mathcal{B}^1_p(X \wedge \phi))$	ϕ is common belief among the agents in I

4. Application: Dining Cryptographers

This problem, described by Chaum in (Chaum, 1988), involves a situation in which three cryptographers are dining together. At the end of the dinner, each of them is secretly informed by the master whether he should pay the bill or not. So, either the master will pay, or he will ask one of the cryptographers to pay. The cryptographers, or some external observer, would like to find out whether the payer is one of them or the master. However, if the payer is one of them, the cryptographers wish to maintain anonymity over the identity of the payer. Of course, we assume that the master himself will not reveal this information, and also do we want the solution to be distributed, i.e. communication can be achieved only via message passing, and there is no central memory or central 'coordinator' which can be used to find out this information.

A possible solution to this problem, described in (Chaum, 1988), is the following: Each cryptographer tosses a coin, which is visible to himself and to his neighbor to the right. Each cryptographer then observes the two coins that he can see, and announces *agree* or *disagree*. If a cryptographer is not paying, he will announce *agree* if the two sides are the same and *disagree* if they are not. However, if he is paying then he will say the opposite. As shown in (Chaum, 1988), if the number of *disagrees* is even, then the master is paying; otherwise, one of the cryptographers is paying. Furthermore, if one of the cryptographers is paying, then neither an external observer nor the other two cryptographers can identify, from their individual information, who exactly is paying.

In order to specify formally the protocol, we use the probabilistic version of CCS, CCS_p.

It will be convenient to use a notation for value-passing in CCS_p, which, following standard lines, can be expressed in the following way (assuming a finite set of values).

$$\begin{array}{rrcl} \mathit{Input} & c(x)\,.\,P & = & \sum_v c_v\,.\,P[v/x] \\ \mathit{Output} & \overline{c}\langle v\rangle & = & \overline{c}_v \end{array}$$

The protocol can be described as the parallel composition of the coin processes $Coin_h$, the cryptographer processes $Crypt_i$, the master process $Master$, and a process $Collect$ whose purpose is to collect all the declarations of the cryptographers, and output them in the form of a tuple. The reason for performing this collection is to avoid the so-called *problem of the omniscient scheduler*: since technically a scheduler depends on the history of the computation, it also depends on the choice of the payer. Hence we could define a scheduler that reveals the identity of the payer via the order of the declarations (for instance by always scheduling the declaration of the payer as the last one). See (Chatzikokolakis *et al.*, 2007b) for more details. Collecting the declarations into one single tuple allows us to avoid this kind of information leak.

The CCS_p terms expressing the protocol are given in Table 4. In this representation, the secret actions are $\overline{pay}_i\langle x\rangle$, and the observable actions are $\overline{outall}\langle y_0, y_1, y_2\rangle$.

Table 4. *The dining cryptographers protocol formalized in CCS_p.*

$$
\begin{array}{rcl}
Master & \triangleq & (\overline{m}_0\langle 0\rangle \,.\, \overline{m}_1\langle 0\rangle \,.\, \overline{m}_2\langle 0\rangle) \oplus_p (\sum_0^2 p_i \overline{m}_{0+i}\langle 1\rangle \,.\, \overline{m}_{1+i}\langle 0\rangle \,.\, \overline{m}_{2+i}\langle 0\rangle) \\
Crypt_i & \triangleq & c_{i,i}(x_0) \,.\, c_{i,i+1}(x_1) \,.\, m_i(x) \,.\, \overline{pay}_i\langle x\rangle \,.\, \overline{out}_i\langle x_0 + x_1 + x\rangle \\
Coin_h & \triangleq & (\overline{c}_{h-1,h}\langle 0\rangle \,.\, \overline{c}_{h,h}\langle 0\rangle) \oplus_{p_h} (\overline{c}_{h-1,h}\langle 1\rangle \,.\, \overline{c}_{h,h}\langle 1\rangle) \\
Collect & \triangleq & out_0(y_0) \,.\, out_1(y_1) \,.\, out_2(y_2) \,.\, \overline{outall}\langle y_0, y_1, y_2\rangle \\
DC & \triangleq & (\nu \vec{c})(\nu \vec{m})(\nu \vec{out})(Master \mid \prod_i Crypt_i \mid \prod_h Coin_h \mid Collect)
\end{array}
$$

The constants p and p_i's represent the probability that the master pays, and the probability that cryptographer i pays, respectively.

In the following we model the property of strong anonymity with respect to *external agents*.[3] We assume that, for every external agent i, the actions $\overline{pay}_j\langle b\rangle$ and $\overline{pay}_{j'}\langle b'\rangle$ are indistinguishable for i, namely for each agent j, j' and bit b, b'

$$f_i(\overline{pay}_j\langle b\rangle) = f_i(\overline{pay}_{j'}\langle b'\rangle)$$

i.e. the *view* that i has of $\overline{pay}_j\langle 0\rangle$ is the same as of $\overline{pay}_j\langle 1\rangle$, $\overline{pay}_{j'}\langle 0\rangle$ and $\overline{pay}_{j'}\langle 1\rangle$.

Strong anonymity can be expressed by the following class of formulas, where p is an arbitrary number in $[0, 1]$, j stands for $\overline{pay}_j\langle 1\rangle$, and the conjunction is taken over all external agents i:

$$\bigwedge_i \ \Box^*({}_i\mathcal{B}_p^p(\overleftarrow{\langle j\rangle}^*) \rightarrow \Box^*{}_i\mathcal{B}_p^p(\overleftarrow{\langle j\rangle}^*))$$

Intuitively, this formula means that at every point of the execution, if Agent i attributes probability p to j (i.e. to Cryptographer j being the payer), then at every point in the future he will attribute to j the same probability. In other words, the observable events of the protocol do not help the agent to refine his estimation of the probability distribution on the secrets. This definition of strong anonymity corresponds to the one given originally by Chaum (Chaum, 1988), requiring the *a priori* probability of the secret event to be equal to its *a posteriori* one.

It is possible to show that, if the coins are fair, the program illustrated in Table 4 satisfies the formula above.

PROPOSITION 18. —

$$DC \models \bigwedge_i \ \Box^*({}_i\mathcal{B}_p^p(\overleftarrow{\langle j\rangle}^*) \rightarrow \Box^*{}_i\mathcal{B}_p^p(\overleftarrow{\langle j\rangle}^*))$$

3. In order to model anonymity also w.r.t. internal agents we need quantification over probabilities. This is left as future work.

PROOF. — The proof proceeds by induction, by proving that, for every i, ${}_i\mathcal{B}^p_p(\overleftarrow{\langle j \rangle^*}) \rightarrow \Box^* {}_i\mathcal{B}^p_p(\overleftarrow{\langle j \rangle^*})$ is an invariant which holds at every step of the execution. ■

The strong anonymity of the Dining Cryptographers with fair coins was also proved in (Bhargava *et al.*, 2005). One major difference with respect to that work is that here we use belief operators, which allow us to express the belief of a given agent. As a consequence, we can distinguish between the belief of internal agents and external ones. In (Bhargava *et al.*, 2005) strong anonymity is expressed in terms of equality of the conditional probabilities of an observable given different culprits, but the relation between agents and observables is not formalized. An internal agent, for instance, observes more than an external one because he can see also the results of the adjacent coins. In the case of a complete ring this is not a problem, but if the ring were incomplete (i.e. missing one arc) ten there would be a difference between external and internal agents, in the sense that strong anonymity would hold only for external agents, not for internal ones. With the approach in (Bhargava *et al.*, 2005) we would not be able to express this difference formally. This is also related to the fact that an approach based simply on probabilities cannot distinguish between subjective uncertainty (belief) and objective uncertainty (truth), as already mentioned in the introduction.

5. Application: Oblivious Transfer

An oblivious transfer is a protocol by which an initiator sends some information to a responder, but remains oblivious (ignorant) as to what was recovered by the responder.

In this section, two variations of the oblivious transfer protocol are considered and specified in DμCEC. For each of them, we give the expression of the agents' post-belief holding after the execution of the protocol, and we give a specification in CCS$_p$ of an implementation for the second one.

5.1. *Oblivious Transfer of one bit only*

5.1.1. *Description*

The *Oblivious-Transfer-of-one-bit-only* protocol, OT_b, was first described in (Kilian, 1988). In this protocol, a single secret bit b is transferred between the initiator (e.g. Alice) and the responder (e.g. Bob). At the end of the protocol, one of the following two events will have occurred, each with a probability $\frac{1}{2}$:

1) the responder Bob learns the value of b, or
2) the responder Bob gains no information about the value of b.

In both cases, at the end of the protocol, Bob knows which of these two events has occurred, while the initiator Alice learns nothing about that.

5.1.2. *Specification*

We express the communication between the agents with two actions s and r defined as follows:

$s \triangleq Send(Alice, b, Bob)$: Alice sends bit b to Bob

$r \triangleq Receive(Bob, b)$: Bob receives bit b

The OT_b protocol can be specified as follows:

$$OT_b \triangleq \langle s \rangle \mathcal{P}^{1/2}_{1/2}(\langle r \rangle^*)$$

Intuitively, this formula means that after the bit was sent by Alice, there is a probability of $\frac{1}{2}$ that Bob eventually receives it.

The post-belief of the agents after the execution of the protocol can be expressed as:

$$PostBelief_b \triangleq \Box^*(\rho(r, s))$$

where $\rho(\alpha, \beta) \triangleq {}_{Alice}\mathcal{B}^{1/2}_{1/2}\overleftarrow{\langle \alpha \rangle}^* \wedge \mathcal{P}^{1/2}_{1/2}({}_{Bob}\mathcal{K}\overleftarrow{\langle \beta \rangle}^*)$ and ${}_a\mathcal{K}\phi \triangleq {}_a\mathcal{B}^1_1\phi$.

This formula can be read as follows: Alice believes with degree of confidence $\frac{1}{2}$ that Bob has received the bit (subjective probability), while, with probability $\frac{1}{2}$, Bob knows the bit that Alice has sent (objective probability).

Note that the fact that Bob knows that a formula ϕ holds (${}_{Bob}\mathcal{K}\phi$) is expressed as the limit of belief, i.e. ${}_{Bob}\mathcal{B}^1_1\phi$.

5.2. *The 1-out-of-2 Oblivious Transfer*

5.2.1. *Description*

In the *1-out-of-2-Oblivious-Transfer* protocol, OT^1_2 (Kilian, 1988), the initiator Alice sends two secret strings u and v, of which the responder Bob receives exactly one. At the end of the protocol, the following three states of affairs hold:

1) Bob learns one of the two strings,
2) Bob gains no information about the other string, and
3) Alice does not know which one of the two strings Bob knows.

5.2.2. *Specification*

In the following, with a slight abuse of notation we use the symbols u and v to represent the actions of *sending* the messages u and v respectively. Analogously we represent by $\underline{u}$ and $\underline{v}$ the complementary actions of *retrieving* u and v.

We now express the OT_2^1 protocol and the agents' post-beliefs.

The first requirement is that, after Alice sends the two strings, there is a probability $\frac{1}{2}$ that Bob retrieves u, and a probability $\frac{1}{2}$ that Bob retrieves v.

$$OT_2^1 \triangleq \langle\!\langle u \rangle\!\rangle^* \langle\!\langle v \rangle\!\rangle^* (\mathcal{P}_{1/2}^{1/2}(\underline{u}) \wedge \mathcal{P}_{1/2}^{1/2}(\underline{v}))$$

Secondly, we require that, after the execution of the protocol, Alice believes with degree of confidence $\frac{1}{2}$ that Bob has received the message u and with degree of confidence $\frac{1}{2}$ that Bob has received the message v (subjective probability), while Bob with probability $\frac{1}{2}$ knows the message that Alice has sent (objective probability).

$$PostBelief_1 \triangleq \Box^* (\rho(\underline{u}, v) \wedge \rho(\underline{v}, u))$$

Finally, we require that if Bob receives u, then he gains no further information about v, and viceversa if he receives v, then he gains no further information about u. This can be expressed with an invariant, like for the Dining Cryptographers:

$$PostBelief_2 \triangleq (\forall p \varrho_p(v, u)) \wedge (\forall q \varrho_q(u, v))$$

where

$$\varrho_p(\alpha, \beta) \triangleq {}_{Bob}\mathcal{B}_p^p(\langle\!\langle \overleftarrow{\alpha} \rangle\!\rangle^*) \rightarrow (\Box^* {}_{Bob}\mathcal{K}(\langle\!\langle \overleftarrow{\beta} \rangle\!\rangle^*) \rightarrow {}_{Bob}\mathcal{B}_p^p(\langle\!\langle \overleftarrow{\alpha} \rangle\!\rangle^*))$$

5.2.3. *Implementation of the OT_2^1 protocol using a public-key cryptosystem*

We consider here the implementation of the oblivious transfer OT_2^1 described in (Even *et al.*, 1985). In the following, $\mathcal{M}$ represents the message space and we assume that all the random choices of messages or bits are made with a uniform probability.

Let $\boxplus, \boxminus : \mathcal{M} \times \mathcal{M} \rightarrow \mathcal{M}$ denote two binary operators which satisfy the following:

1) For every $x \in \mathcal{M}$, the mapping $y \mapsto x \boxplus y$ is a permutation on $\mathcal{M}$.
2) For every $y \in \mathcal{M}$, the mapping $x \mapsto x \boxplus y$ is a permutation on $\mathcal{M}$.
3) For every $x, y \in \mathcal{M}$, $(x \boxplus y) \boxminus y = x$.

Furthermore, we assume that these operators are known by both agents. For instance, when using RSA as public-key cryptosystem, $x \boxplus y$ can be defined as the reduction modulo N (the RSA's modulus) of $x + y$ while $x \boxminus y$ can be defined as the reduction modulo N of $x - y$.

In our process calculus, the OT_2^1 protocol can be specified as the parallel composition of the initiator process $Init$ and of the responder process $Resp$. The initiator Alice wants to send one of the two strings u and v. She starts the communication by generating a public key/private key pair (e, d) and sending her public key along with two random messages m_0 and m_1 to the responder Bob. Bob chooses a random message m and a random bit r and sends back to Alice $z = E(m, e) \boxplus m_r$, where $E(m, e)$ denotes the encryption of the message m with the public key e. Similarly, $D(c, d)$ denotes the decryption of a string c with the private key d and we have $D(E(m, e), d) = m$.

Alice (who does not know r) computes both $e_0 = z \boxminus m_0$ and $e_1 = z \boxminus m_1$. Then, Alice decrypts with her private key d both e_0 and e_1, obtaining respectively d_0 and d_1. Only one of these two values, namely $d_r = D(E(m, e), d)$, is identical to the initial message m. This however cannot be determined by Alice since she does not know the value of r and m.

Alice chooses then a random bit s and transmits to Bob the tuple $(u \boxplus d_s, v \boxplus d_{1-s}, s)$. Depending on the choice of s, two independent situations may occur: either $s = r$, and thus $d_s = d_r = m$ and Bob can read u without learning anything about v, or $s = 1 - r$ and Bob can read v without learning anything about u. Both events have equal probability to occur (due to the uniform probability on the random choice of r and s), which ensures that the first and second intended properties of the protocol are satisfied.

Moreover, since Alice only gets the information $z = E(m, e) \boxplus m_r$ and m is randomly chosen by Bob, z does not give Alice any information about r, which ensures that the third intended property of the protocol is satisfied as well.

The protocol narration of OT_2^1 is as follows:

$$\begin{array}{lccc} 1. & Alice & \xrightarrow{e,m_0,m_1} & Bob \\ 2. & Bob & \xrightarrow{E(m,e) \boxplus m_r} & Alice \\ 3. & Alice & \xrightarrow{u \boxplus d_s, v \boxplus d_{1-s}, s} & Bob \end{array}$$

We describe the implementation of the protocol OT_2^1 in CCS$_p$ in Table 5.

The unfolding of the CCS$_p$ terms representing the protocol OT_2^1 is illustrated in Table 6.

5.2.4. *Verification*

In this section we show that our protocol satisfies OT_2^1, $PostBelief_1$ and $PostBelief_2$.

The initial prefix in the formula for OT_2^1 specifies that eventually, the actions u and v occur (i.e. the messages u and v are sent): $POT_2^{1(5,6)} \models OT_2^1$. This is indeed achieved in our protocol by the (synchronous) action out_2 performed in step

Table 5. *Implementation of the protocol OT_2^1 in CCS$_p$. The probabilities p and q represent the uniform probabilities over the space of message pairs (m_0, m_1) and the space of the messages m, respectively.*

$$
\begin{array}{lcl}
\mathit{Init} & \triangleq & \sum\!\!\!\!\!\!\sum_{m_0,m_1} p\ \overline{out}_1\langle e, m_0, m_1\rangle\,.\,in(z)\,. \\
 & & \quad \sum\!\!\!\!\!\!\sum_{s\in\{0,1\}} 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
\mathit{Resp} & \triangleq & out_1(e, x_0, x_1)\,.\sum\!\!\!\!\!\!\sum_{r\in\{0,1\}} 1/2 \sum\!\!\!\!\!\!\sum_{m} q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
 & & \quad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
POT_2^1 & \triangleq & \nu(\mathit{Init}|\mathit{Resp})
\end{array}
$$

$POT_2^{1(4,4)}$. The remaining part of the formula OT_2^1 is true if u and v are received each with a probability of exactly one half, which holds as explained beforehand in the protocol description.

On the contrary to OT_2^1, the prefixes of $PostBelief_1$ and $PostBelief_2$ are used to describe invariant properties that have therefore to hold at every step of the protocol: $\forall (i,j) \in \{(0,0),(1,0),(2,1),(2,2),(2,3),(3,4),(4,4),(5,5),(5,6)\}$, $POT_2^{1(i,j)} \models PostBelief_1 \wedge PostBelief_2$.

The first part of $PostBelief_1$, ${}_{Alice}\mathcal{B}_{1/2}^{1/2}(\overleftarrow{\langle u \rangle}^*) \wedge {}_{Alice}\mathcal{B}_{1/2}^{1/2}(\overleftarrow{\langle v \rangle}^*)$, which describes the subjective knowledge of Alice, is the transcription of the third axiom, while the remaining of the formula, which specifies an objective knowledge of Bob, corresponds to the first axiom. Similarly, $PostBelief_2$ is the transcription of the second axiom. We already saw that these axioms, and thus $PostBelief_1$ and $PostBelief_2$ hold at the end of the protocol. They also hold at the beginning of the protocol. From the description of the protocol, one can finally see that no step leads to a change of these beliefs. Therefore, $PostBelief_1$ and $PostBelief_2$ hold at each step of the protocol.

Note that for the sake of simplicity, several aspects of our description which were not directly necessary for our purposes, such as the cryptographic primitives or the fixpoint operators, have been left informal.

6. Conclusion

6.1. *Achievements*

We have achieved novel formalizations of probabilistic anonymity and oblivious transfer in a promising modal logic, namely the doxastic μ-calculus with error control (DμCEC). Our formalizations can be validated on the protocol of the dining cryptographers, and on the protocols of 1-bit and 1-out-of-2-strings oblivious transfer. The

Table 6. *Unfolding of the protocol* OT_2^1.

Initiator

$$
\begin{array}{lcl}
Init^0 & := & \sum_{m_0,m_1} p\ \overline{out}_1\langle e, m_0, m_1\rangle\,.\,in(z)\,. \\
 & & \quad \sum_{s\in\{0,1\}} 1/2\ \ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
Init^1 & := & \overline{out}_1\langle e, m_0, m_1\rangle\,.\,in(z)\,. \\
 & & \quad \sum_{s\in\{0,1\}} 1/2\ \ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
Init^2 & := & in(z)\,. \\
 & & \quad \sum_{s\in\{0,1\}} 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
Init^3 & := & \sum_{s\in\{0,1\}} 1/2\ \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
Init^4 & := & \overline{out}_2\langle u \boxplus D(z \boxminus m_s, d), v \boxplus D(z \boxminus m_{1-s}, d), s\rangle\,.\,0 \\
Init^5 & := & 0
\end{array}
$$

Responder

$$
\begin{array}{lcl}
Resp^0 & := & out_1(e, x_0, x_1)\,.\sum_{r\in\{0,1\}} 1/2 \sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
 & & \quad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^1 & := & \sum_{r\in\{0,1\}} 1/2 \sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,. \\
 & & \quad out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^2 & := & \sum_m q\ \overline{in}\langle E(m, e) \boxplus x_r\rangle\,.\,out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^3 & := & \overline{in}\langle E(m, e) \boxplus x_r\rangle\,.\,out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^4 & := & out_2(y_0, y_1, s)\,.\,\overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^5 & := & \overline{out}\langle y_{s+r} \boxminus m\rangle\,.\,0 \\
Resp^6 & := & 0
\end{array}
$$

Protocol

$$POT_2^{1(i,j)} \ \triangleq\ \nu(Init^i | Resp^j)$$

Unfolding:

$$POT_2^{1(0,0)} \xrightarrow{m_0,m_1} POT_2^{1(1,0)} \xrightarrow{\tau(out_1)} POT_2^{1(2,1)} \xrightarrow{r} POT_2^{1(2,2)} \xrightarrow{q} POT_2^{1(2,3)} \xrightarrow{\tau(in)} POT_2^{1(3,4)} \xrightarrow{s} POT_2^{1(4,4)} \xrightarrow{\tau(out_2)} POT_2^{1(5,5)} \xrightarrow{out} POT_2^{1(5,6)}$$

intuitiveness of our formalizations is due to (1) our distinction between belief and internalized probabilistic truth, (2) the dynamicity of our notion of belief and internalized probabilistic truth, and (3) the introduction of lower and upper bounds (error control) therefor.

We have also shown that belief and internalized probabilistic truth satisfy a probabilistic analogue of standard KD45-belief, and that belief and internalized probabilistic truth can be flattened on certain, but different conditions.

6.2. *Future work*

As future work for DμCEC, we envisage the development of *tool-support*, its *axiomatization*, and the introduction of *cryptographic data types* and restricted *logical quantification* (over messages, including probability values).

Further, the combination of operators for belief with error control and time allows for the expression of probabilistic statements as arbitrary compound *sentences* in our logic, rather than as compound *terms* as in the language of traditional probability theory. It is natural to study the advantages and disadvantages between these two styles of expression.

Finally, given the expressibility of oblivious transfer in DμCEC and the foundational power of oblivious transfer for modern cryptography (Kilian, 1988), we believe that DμCEC can serve as a framework for comparing abstract cryptography based on Dolev-Yao message-passing and concrete cryptography based on bit-string message-passing, thus bringing a new approach to a problem that has received a lot of attention recently, see for intance the work of (Cortier *et al.*, 2007).

7. References

Beauxis R., Chatzikokolakis K., Palamidessi C., Panangaden P., "Formal Approaches to Information-Hiding (Tutorial)", *in* G. Barthe, C. Fournet (eds), *Proceedings of the Third Symposium on Trustworthy Global Computing (TGC 2007)*, vol. 4912 of *Lecture Notes in Computer Science*, Springer, pp. 347-362, 2008.

Bhargava M., Palamidessi C., "Probabilistic Anonymity", *in* M. Abadi, L. de Alfaro (eds), *Proceedings of CONCUR*, vol. 3653 of *Lecture Notes in Computer Science*, Springer, pp. 171–185, 2005. `http://www.lix.polytechnique.fr/~catuscia/papers/Anonymity/concur.pdf`.

Chatzikokolakis K., Palamidessi C., "A Framework for Analyzing Probabilistic Protocols and its Application to the Partial Secrets Exchange", *Theoretical Computer Science*, vol. 389, num. 3, pp. 512-527, 2007a. A short version of this paper appeared in the *Proceedings of the Symposium on Trustworthy Global Computing (TGC)*, volume 3705 of LNCS, pages 146-162. Springer. `http://www.lix.polytechnique.fr/~catuscia/papers/PartialSecrets/TCSreport.pdf`.

Chatzikokolakis K., Palamidessi C., "Making Random Choices Invisible to the Scheduler", *in* L. Caires, V. T. Vasconcelos (eds), *Proceedings of CONCUR'07*, vol. 4703 of *Lecture Notes in Computer Science*, Springer, pp. 42–58, 2007b. `http://www.lix.polytechnique.fr/~catuscia/papers/Scheduler/report.pdf`.

Chatzikokolakis K., Palamidessi C., "Making Random Choices Invisible to the Scheduler", *Information and Computation*, 2009. Submitted.

Chaum D., "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability", *Journal of Cryptology*, vol. 1, pp. 65–75, 1988.

Cortier V., Rusinowitch M., Zalinescu E., "Relating two standard notions of secrecy", *Logical Methods in Computer Science*, 2007.

Dechesne F., Mousavi M., Orzan S., "Operational and Epistemic Approaches to Protocol Analysis: Bridging the Gap", *in* N. Dershowitz, A. Voronkov (eds), *Proceedings of the 14th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR'07)*, vol. 4790 of *Lecture Notes in Computer Science*, Springer, pp. 226–241, 2007.

Desharnais J., Edalat A., Panangaden P., "A Logical Characterization of Bisimulation for Labeled Markov Processes", *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, pp. 478–487, 1998.

Even S., Goldreich O., Lempel A., "A randomized protocol for signing contracts", *Commun. ACM*, vol. 28, num. 6, pp. 637–647, 1985.

Halpern J. Y., O'Neill K. R., "Anonymity and information hiding in multiagent systems", *Journal of Computer Security*, vol. 13, num. 3, pp. 483–512, 2005a.

Halpern J. Y., Pucella R., "Probabilistic algorithmic knowledge", *Journal of Logical Methods in Computer Science*, 2005b.

Kilian J., "Founding Cryptography on Oblivious Transfer", *Proceedings of the 20th ACM Annual Symposium on the Theory of Computing*, pp. 20-31, 1988.

Larsen K. G., Skou A., "Bisimulation through Probabilistic Testing", *Information and Computation*, vol. 94, num. 1, pp. 1–28, September, 1991.

Lomuscio A., Penczek W., "Symbolic Model Checking for Temporal-Epistemic Logics", *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, 2007.

Nielsen M., "Reasoning About the Past", *in* L. Brim, J. Gruska, J. Zlatuska (eds), *Proceedings of MFCS'98*, vol. 1450 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 117–128, 1998.

Parma A., Segala R., "Logical Characterizations of Bisimulations for Discrete Probabilistic Systems", *in* H. Seidl (ed.), *10th International Conference on the Foundations of Software Science and Computational Structures FOSSACS'07 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007, Braga, Portugal, March 24-April 1, 2007, Proceedings*, vol. 4423 of *Lecture Notes in Computer Science*, Springer, pp. 287–301, 2007.

Rabin M. O., "How to exchange secrets by oblivious transfer", *Technical Memo TR-81, Aiken Computation Laboratory, Harvard University*, 1981.

Segala R., Modeling and Verification of Randomized Distributed Real-Time Systems, PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June, 1995. Available as Technical Report MIT/LCS/TR-676.

Segala R., "Probability and Nondeterminism in Operational Models of Concurrency", *in* C. Baier, H. Hermanns (eds), *Proceedings of the 17th International Conference on Concurrency Theory (CONCUR)*, vol. 4137 of *Lecture Notes in Computer Science*, Springer, pp. 64-78, 2006.

Segala R., Lynch N., "Probabilistic simulations for probabilistic processes", *Nordic Journal of Computing*, vol. 2, num. 2, pp. 250–273, 1995. An extended abstract appeared in *Proceedings of CONCUR '94*, LNCS 836: 481-496.