


1 The Sub-Additives: A Proof Theory for 2 Probabilistic Choice extending Linear Logic

3 **Ross Horne**

4 Computer Science and Communications, University of Luxembourg,

5 Esch-sur-Alzette, Luxembourg

6 ross.horne@uni.lu

7  <https://orcid.org/0000-0003-0162-1901>

8 — Abstract —

9 Probabilistic choice, where each branch of a choice is weighted according to a probability distribution, is an established approach for modelling processes, quantifying uncertainty in the environment and other sources of randomness. This paper uncovers new insight showing probabilistic choice has a purely logical interpretation as an operator in an extension of linear logic. By forbidding projection and injection, we reveal additive operators between the standard *with* and *plus* operators of linear logic. We call these operators the *sub-additives*. The attention of the reader is drawn to two sub-additive operators: the first being sound with respect to probabilistic choice; while the second arises due to the fact that probabilistic choice cannot be self-dual, hence has a de Morgan dual counterpart. The proof theoretic justification for the sub-additives is a cut elimination result, employing a technique called *decomposition*. The justification from the perspective of modelling probabilistic concurrent processes is that implication is sound with respect to established notions of probabilistic refinement, and is fully compositional.

21 **2012 ACM Subject Classification** Theory of computation → Proof theory; Theory of computation → Process calculi; Theory of computation → Linear logic

23 **Keywords and phrases** calculus of structures, probabilistic choice, probabilistic refinement

24 **Digital Object Identifier** 10.4230/LIPIcs.FSCD.2019.23

25 **Acknowledgements** I thank Bogdan Aman and Gabriel Ciobanu for their enjoyable discussions.

26 **1 Introduction**

27 This paper lays down a novel foundation for a proof theory of formulae modelling concurrent processes with mixed probabilistic and non-deterministic choice. Probabilistic choices refine non-deterministic choices by indicating the probability with which one action or another occurs, and have been introduced in game theory and process calculi to model measurable uncertainty in the environment, such as a decision made by tossing a coin.

32 It is already well known that, in various *processes-as-formulae* approaches to modelling processes using extensions of linear logic [15], the additive operators can be used to model non-deterministic choices. The key novelty of this work is the observation that probabilistic choices can also be handled using additive operators, of a more restrictive kind, which we call the *sub-additives*.

37 In what follows we clarify the *processes-as-formulae* approach to modelling processes directly as formulae in extensions of linear logic. We highlight key observations leading to probabilistic sub-additive operators, and explain why their proof theory is non-trivial. Furthermore, for readers for whom the discovery of a novel proof theory is insufficient motivation, we highlight that, unlike most semantics previously proposed for probabilistic concurrent processes, our model is exceptionally compositional, admitting *action refinement*.



© Ross Horne;

licensed under Creative Commons License CC-BY

4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019).

Editor: Herman Geuvers; Article No. 23; pp. 23:1–23:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

43 1.1 The processes-as-formulae paradigm

44 Various approaches to modelling processes by directly embedding them as formulae in an
 45 extension of linear logic have been floated since the discovery of linear logic (see [22] for a
 46 comparison). Progress in this processes-as-formulae approach has been accelerated by an
 47 advance in proof theory — the *calculus of structures* [17] — a generalisation of the sequent
 48 calculus. Process models not limited to CCS [3], session types [5], attack trees [21] and the
 49 π -calculus [23, 24] have been tackled using the processes-as-formulae approach.

50 An advantage of the processes-as-formulae paradigm is that formulae modelling processes
 51 can be directly compared using *implication* in the logical system. Furthermore, there are
 52 *no design decisions*, since the semantics are determined by the principles of *cut elimina-*
 53 *tion*. In every process model this approach always leads us to a preorder over processes
 54 with appealing properties. The preorder obtained enjoys the following properties: it is a
 55 congruence; is sound with respect to most commonly-used process preorders, including weak
 56 simulation [22], and pomset ideals [21]; and respects *action refinement* — the ability to refine
 57 atomic actions with larger sub-processes. This makes implication highly *compositional*.

58 In this work, by introducing an operator modelling probabilistic choice, the above prop-
 59 erties can also be achieved in the probabilistic setting, where preorders are defined with
 60 respect to probability distributions. To emphasise this point we prove that implication in
 61 this work is sound with respect to a notion of refinement called weak *probabilistic simu-*
 62 *lation* [37, 2]. A famous result in the theory of probabilistic processes [10], means that,
 63 equivalently, implication is sound with respect to *probabilistic may testing* [27, 30]. An
 64 advantage implication has over simulation/testing semantics is that, as mentioned above,
 65 implication guarantees a greater degree of compositionality.

66 1.2 Motivation: uncovering the probabilistic sub-additive operators

67 We explain key observations that uncover the probabilistic sub-additive operators. Sub-
 68 additive operators are restricted forms of additive conjunction or disjunction, found in linear
 69 logic. Sub-additives forbid projection and injection, while permitting other properties of the
 70 additives, notably idempotency.

71 Firstly, consider how the standard additives can be used to model non-deterministic
 72 choice. To be specific, in linear logic, we have *with* $\&$, which enjoys the following *projection*
 73 *laws*, where \multimap is linear implication: $P \& Q \multimap P$ and $P \& Q \multimap Q$. For example, *heads & tails*
 74 can be used to model a process that does not toss a coin but instead chooses on which side
 75 to lay the coin. This can be refined by process *heads* that always chooses to lay down heads.
 76 This does not model tossing a coin, instead modelling a decision the process can make.

77 The key observation is, by restricting additives such that **projection and injection**
 78 **are forbidden**, we are able to model probabilistic choice. For example, *heads $\oplus_{1/2}$ tails*
 79 models a fair coin, where heads or tails occurs with probability $1/2$. Notice the process
 80 cannot influence the outcome of the coin toss, therefore such a fair coin cannot be refined to
 81 *heads*. The absence of this refinement corresponds to forbidding projection. Furthermore,
 82 it is standard for probabilistic processes, that a fair coin **cannot** be refined to an unfair
 83 coin where the balance of probabilities are different from $1/2$ each. Notions of probabilistic
 84 refinement preserve the balance of probabilities.

85 Although projection/injection are forbidden, non-deterministic choice and probabilistic
 86 choice are related. For example, non-deterministic choice *heads & tails* can be refined to
 87 probabilistic choice *heads $\oplus_{1/2}$ tails*. This refinement can be established by proving the

88 following using the logical system in the body of this work.

89 $heads \& tails \multimap heads \oplus_{1/2} tails$

90 Such a refinement, introducing probabilities, is standard for *probabilistic simulation* or,
91 equivalently, *probabilistic may testing* [27, 30, 9].

92 Note, there are many other modelling capabilities of the logic in this work. For example,
93 we can capture probabilistic choice with margins of error, and probabilistic model checking,
94 within a bound of probability. Application wise, such models have been used for a wide
95 range of problems, e.g., quantifying the degree of anonymity offered by privacy protocols,
96 or quantifying risk in attacker models. This work focusses on introducing our new logical
97 system ΔMAV and providing clear and simple examples.

98 The interplay between the sub-additives and both sequential and parallel composition
99 can be non-trivial. For example, we discover, for subtle reasons explained later, in the
100 presence of parallel composition, operator \oplus_p cannot be self-dual. Thereby we obtain also
101 a de Morgan dual operator $\&_p$, essential for completing the symmetry demanded by a logic
102 satisfying cut elimination. The central result of this paper, cut elimination (Theorem 2),
103 ensures these new sub-additive operators co-exist happily with other operators of linear
104 logic — a prerequisite for using implication with confidence. Furthermore, the soundness
105 of linear implication as a notion of probabilistic refinement (Theorem 4) is verified and
106 the merits of this notion of refinement discussed. In particular, we claim that this logical
107 approach to modelling processes helps us discover the coarsest notion of refinement, in the
108 literature, that can: firstly, handle probabilistic processes; secondly, accommodate parallel
109 composition; and, thirdly, permit *action refinement* [41].

110 **Outline of the paper.** Section 2, provides established background material on probabilistic
111 processes. Section 3, recalls MALL in the calculus of structures, and introduces the extended
112 system ΔMAV featuring a pair of sub-additive operators. Section 4 provides a series of
113 examples illustrating how we can construct, more traditional, probabilistic simulations from
114 proofs in ΔMAV . Section 5, outlines the proof of cut elimination, necessary to justify the
115 logical system proposed. Section 6 highlights the existence of further sub-additive operators
116 between the standard operators of linear logic.

117 **2 Background: an established notion of probabilistic simulation**

118 We begin with background on probabilistic simulation. We select a minimal probabilistic
119 process calculus and standard notion of probabilistic simulation.

120 Note there are numerous probabilistic calculi in the literature mixing non-deterministic
121 and probabilistic choice, not limited to probabilistic extensions of CCS [28], CSP [11], and
122 the π -calculus [33]. Due to the rich proof calculi developed [23], expressive process models
123 can be handled by techniques in this work. For scientific clarity, we select here a minimal
124 calculus in order to make a clear comparison with the new logical approach to probabilistic
125 refinement introduced in subsequent sections.

126 The syntax of our minimal process calculus is drawn from terms in the following grammar,
127 where ‘ a ’ represents actions.

128
$$t ::= \text{ok (successful completion)} \mid a.t \text{ (action prefix)} \mid t \parallel t \text{ (parallel composition)}$$

$$\mid t \sqcap t \text{ (non-deterministic choice)} \mid t +_p t \text{ (probabilistic choice)}$$

129 *Discrete probability distributions* are uniquely determined by a *probability mass function*
130 $\Delta : S \rightarrow [0, 1]$ over a set S of process terms such that $\sum_{t \in S} \Delta(t) = 1$. A Dirac distribution for

131 process term s , written $\mathbf{1}_s$, is defined by the probability mass function such that $\Delta(s) = 1$.
 132 For probability p and distributions, Δ_1 and Δ_2 *linear combination* $p\Delta_1 + (1-p)\Delta_2$, defined
 133 as $(p\Delta_1 + (1-p)\Delta_2)(t) = p\Delta_1(t) + (1-p)\Delta_2(t)$, is a distribution and *dot product* $\Delta_1 \cdot \Delta_2$ is
 134 defined such that $(\Delta_1 \cdot \Delta_2)(t \parallel u) = \Delta_1(t)\Delta_2(u)$ and 0 elsewhere.

135 Process terms are mapped to distributions using the following function δ .

$$\begin{aligned}
 136 \quad & \delta(\text{ok}) = \mathbf{1}_{\text{ok}} & \delta(a.t) = \mathbf{1}_{a.t} & \delta(t \sqcap t) = \mathbf{1}_{t \sqcap t} \\
 137 \quad & \delta(t +_p t) = p\delta(t) + (1-p)\delta(t) & \delta(t \parallel t) = \delta(t) \cdot \delta(t)
 \end{aligned}$$

139 Labelled transitions from process terms to distributions are defined by the following rules,
 140 where label α ranges over any action a or τ .

$$\begin{array}{c}
 141 \quad \frac{}{a.t \xrightarrow{a} \delta(t)} \quad \frac{i \in \{1, 2\}}{t_1 \sqcap t_2 \xrightarrow{\tau} \delta(t_i)} \quad \frac{t_1 \xrightarrow{\alpha} \Delta}{t_1 \parallel t_2 \xrightarrow{\alpha} \Delta \cdot \delta(t_2)} \quad \frac{t_2 \xrightarrow{\alpha} \Delta}{t_1 \parallel t_2 \xrightarrow{\alpha} \delta(t_1) \cdot \Delta} \\
 142
 \end{array}$$

143 Labelled transitions lift to weak transitions over distributions, as according to the following
 144 four clauses, which allow zero or more τ -transitions. Firstly, $\Delta \xrightarrow{\tau} \Delta$; secondly, if for
 145 all i , $s_i \xrightarrow{\alpha} \Delta_i$ and $\sum_{i \in I} p_i = 1$ then $\sum_{i \in I} p_i \mathbf{1}_{s_i} \xrightarrow{\alpha} \sum_{i \in I} p_i \Delta_i$; thirdly, if $\Delta_1 \xrightarrow{\tau} \Delta_2$ then
 146 $p\Delta_1 + (1-p)\mathcal{E} \xrightarrow{\tau} p\Delta_2 + (1-p)\mathcal{E}$, fourthly, if $\Delta_1 \xrightarrow{\tau} \Delta_2$ and $\Delta_2 \xrightarrow{\alpha} \Delta_3$, then $\Delta_1 \xrightarrow{\alpha} \Delta_3$.

147 For tighter results, we also employ the predicate \checkmark indicating successful *termination*,
 148 defined such that $\text{ok}\checkmark$ and if $t_1\checkmark$ and $t_2\checkmark$ then $(t_1 \parallel t_2)\checkmark$. Termination extends to distribu-
 149 tions in the obvious way such that if $t\checkmark$ then $\mathbf{1}_t\checkmark$ and if $\Delta\checkmark$ and $\mathcal{E}\checkmark$ then $(p\Delta + (1-p)\mathcal{E})\checkmark$.

150 The above labelled transitions and termination predicate are employed in the following
 151 definition of a *weak complete probabilistic simulation*. The definition also employs a standard
 152 *lifting* of relations from processes to distributions.

153 ► **Definition 1.** For a relation \mathcal{R} between processes and distributions, its *lifting* $\hat{\mathcal{R}}$ is such
 154 that: if, for all i , $t_i \mathcal{R} \Delta_i$ and $\sum_{i \in I} p_i = 1$, then $\sum_{i \in I} p_i \mathbf{1}_{t_i} \hat{\mathcal{R}} \sum_{i \in I} p_i \Delta_i$. A relation between
 155 processes and distributions \mathcal{R} is a weak complete probabilistic simulation whenever:

- 156 ■ If $s \mathcal{R} \Delta$ and $s \xrightarrow{\alpha} \mathcal{E}$, there exists \mathcal{E}' such that $\Delta \xrightarrow{\alpha} \mathcal{E}'$ and $\mathcal{E} \hat{\mathcal{R}} \mathcal{E}'$.
- 157 ■ If $t \mathcal{R} \Delta$ and $t\checkmark$ then there exists \mathcal{E} such that $\Delta \xrightarrow{\tau} \mathcal{E}$ and $\mathcal{E}\checkmark$.

158 If there exists weak complete probabilistic simulation \mathcal{R} such that $\delta(t_1) \hat{\mathcal{R}} \delta(t_2)$, then we
 159 say t_2 *simulates* t_1 .

160 We refer to the above notion simply as *probabilistic simulation* throughout this work.
 161 Recall this definition is used only as a reference to show the logic we develop is sound with
 162 respect to such a standard notion of probabilistic refinement, and contains no new concepts.
 163 We provide examples later in subsequent sections when making such a comparison.

164 3 Extending linear logic with probabilistic sub-additive operators

165 In this section, we introduce a proof system featuring the probabilistic sub-additives. The
 166 system is a conservative extension of multiplicative-additive linear logic (MALL). Therefore,
 167 first we recall a presentation of MALL in the calculus of structures, a generalisation of
 168 the sequent calculus. We employ the calculus of structures, since it provides additional
 169 expressive power demanded by our target logic ΔMAV .

170 3.1 An established presentation of MALL in the calculus of structures

171 The fragment of linear logic MALL was one of the first proof systems studied in the calculus
 172 of structures [38]. Fig 1 recalls a proof system for multiplicative-additive linear logic MALL

173 in the calculus of structures. Inference rules apply in any context. We assume formulae
 174 are always in negation-normal-form, where negation is always pushed to atoms, a , by the
 175 following function, inducing De Morgan dualities.

$$176 \quad \overline{P \oplus Q} = \overline{P} \& \overline{Q} \quad \overline{P \& Q} = \overline{P} \oplus \overline{Q} \quad \overline{\overline{a}} = a \quad \overline{P \otimes Q} = \overline{P} \wp \overline{Q} \quad \overline{P \wp Q} = \overline{P} \otimes \overline{Q} \quad \overline{\circ} = \circ$$

178 The formulation of MALL in Fig. 1 was employed to prove cut elimination for a non-
 179 commutative extension of MALL called MAV [20]. The rules are also similar to a version
 180 used to study focussing in the calculus of structures [4].

structural congruence:

$$\begin{array}{lll} P \wp Q \equiv Q \wp P & (P \wp Q) \wp R \equiv P \wp (Q \wp R) & \circ \wp P \equiv P \\ P \otimes Q \equiv Q \otimes P & (P \otimes Q) \otimes R \equiv P \otimes (Q \otimes R) & \circ \otimes P \equiv P \end{array}$$

inference rules:

$$\begin{array}{lll} \frac{\mathcal{C}\{\circ\}}{\mathcal{C}\{\overline{a} \wp a\}} \text{interact} & \frac{\mathcal{C}\{(P \wp Q) \otimes R\}}{\mathcal{C}\{P \wp (Q \otimes R)\}} \text{switch} & \frac{\mathcal{C}\{\circ\}}{\mathcal{C}\{\circ \& \circ\}} \text{tidy} \\ \frac{\mathcal{C}\{P_1\}}{\mathcal{C}\{P_1 \oplus P_2\}} \text{choose left} & \frac{\mathcal{C}\{P_2\}}{\mathcal{C}\{P_1 \oplus P_2\}} \text{choose right} & \frac{\mathcal{C}\{(P \wp R) \& (Q \wp R)\}}{\mathcal{C}\{(P \& Q) \wp R\}} \text{external} \end{array}$$

■ **Figure 1** Structural congruence and inference rules for MALL in the calculus of structures.

181 The structural congruence ensures the multiplicatives *par* \wp and *times* \otimes are commutative
 182 monoids with a common unit. The *switch* rule and *interact* rule form multiplicative linear
 183 logic. Regarding the inference rules, there is one rule, *choose*, for additive *plus* \oplus , which
 184 chooses either the left or right branch during proof search. The rule *external* distributes the
 185 additive *with* $\&$ over *par*, forcing both branches to be explored. The *tidy* rule ensures proof
 186 search is successful only if both branches are successful.

187 A *derivation* is a sequence of zero or more rule instances, where the structural congruence
 188 can be applied at any step. The bottommost formula is the *conclusion* and the topmost is
 189 the *premiss*. A proposition P is *provable*, written $\vdash P$, whenever there exists a derivation
 190 with conclusion P and premise \circ . *Linear implication* $P \multimap Q$ is defined as $\overline{P} \wp Q$; hence a
 191 provable linear implication is written $\vdash P \multimap Q$.

192 This presentation of MALL has a common unit for the multiplicatives, consequently
 193 implication $\vdash P \otimes Q \multimap P \wp Q$ holds. The reader familiar with linear logic will observe this
 194 means the *mix* rule is admissible. Note the results in this paper also hold for a formulation
 195 of MALL that does not admit *mix*, but *mix* is included so as the logic extends immediately
 196 to non-commutative logic.

197 3.2 Extending with the probabilistic sub-additives (and sequentiality)

198 The calculus of structures provides a setting in which the sub-additives can be expressed and
 199 evaluated. We explain briefly the new rules of the structural congruence and the inference
 200 rules in Fig. 2. Note we assume a probability p is always such that $0 < p < 1$, thus any
 201 sub-formula that appears in a probabilistic choice occurs with non-zero probability.

202 The rule of the structural congruence for the probabilistic sub-additives, Fig. 2, ensures
 203 the balance of probabilities is maintained when applying idempotency, associativity and

204 commutativity. By maintaining the balance of probabilities, structural congruence preserves
 205 underlying probability distributions. For example $p\Delta + (1-p)\Delta = \Delta$, hence we have a
 206 weighted form of idempotency $P \oplus_p P = P$.

207 For associativity, observe if Δ_0, Δ_1 and Δ_2 are distributions corresponding to P, Q and
 208 R respectively, then $q(p\Delta_0 + (1-p)\Delta_1) + (1-q)\Delta_2 = r\Delta_0 + (1-r)(s\Delta_1 + (1-s)\Delta_2)$ only
 209 if $r = pq$ and $(1-r)s = q(1-p)$. Furthermore, commuting formulae inverts probabilities
 210 $(p\Delta_1 + (1-p)\Delta_2) = (1-p)\Delta_2 + p\Delta_1$.

structural congruence:

$$\begin{array}{lll}
 P \&_r Q \equiv Q \&_{1-r} P & P \&_r P \equiv P & (P \&_p Q) \&_q R \equiv P \&_{pq} \left(Q \&_{\frac{q(1-p)}{1-pq}} R \right) \\
 P \oplus_r Q \equiv Q \oplus_{1-r} P & P \oplus_r P \equiv P & (P \oplus_p Q) \oplus_q R \equiv P \oplus_{pq} \left(Q \oplus_{\frac{q(1-p)}{1-pq}} R \right) \\
 P \triangleleft P \equiv P & P \equiv P \triangleleft P & (P \triangleleft Q) \triangleleft R \equiv P \triangleleft (Q \triangleleft R)
 \end{array}$$

inference rules:

$$\begin{array}{c}
 \frac{\mathcal{C}\{ (P \wp R) \&_p (Q \wp S) \}}{\mathcal{C}\{ (P \oplus_p Q) \wp (R \&_p S) \}} \text{confine} \\
 \\
 \frac{\mathcal{C}\{ (P \wp R) \oplus_q (Q \wp S) \}}{\mathcal{C}\{ (P \oplus_q Q) \wp (R \oplus_q S) \}} \text{medial} & \frac{\mathcal{C}\{ (P \&_p R) \oplus_q (Q \&_p S) \}}{\mathcal{C}\{ (P \oplus_q Q) \&_p (R \oplus_q S) \}} \text{medial} \\
 \\
 \frac{\mathcal{C}\{ (P \& R) \oplus_q (Q \& S) \}}{\mathcal{C}\{ (P \oplus_q Q) \& (R \oplus_q S) \}} \text{medial} & \frac{\mathcal{C}\{ (P \& R) \&_p (Q \& S) \}}{\mathcal{C}\{ (P \&_p Q) \& (R \&_p S) \}} \text{medial} \\
 \\
 \frac{\mathcal{C}\{ (P \wp R) \triangleleft (Q \wp S) \}}{\mathcal{C}\{ (P \triangleleft Q) \wp (R \triangleleft S) \}} \text{medial} & \frac{\mathcal{C}\{ (P \& R) \triangleleft (Q \& S) \}}{\mathcal{C}\{ (P \triangleleft Q) \& (R \triangleleft S) \}} \text{medial} \\
 \\
 \frac{\mathcal{C}\{ (P \&_p R) \triangleleft (Q \&_p S) \}}{\mathcal{C}\{ (P \triangleleft Q) \&_p (R \triangleleft S) \}} \text{medial} & \frac{\mathcal{C}\{ (P \triangleleft R) \oplus_p (Q \triangleleft S) \}}{\mathcal{C}\{ (P \oplus_p Q) \triangleleft (R \oplus_p S) \}} \text{medial}
 \end{array}$$

linear negation:

$$\overline{P \triangleleft Q} = \overline{P} \triangleleft \overline{Q} \quad \overline{P \oplus_p Q} = \overline{P} \&_p \overline{Q} \quad \overline{P \&_p Q} = \overline{P} \oplus_p \overline{Q}$$

■ **Figure 2** Rules for the probabilistic sub-additive operators and *seq* in ΔMAV , extending Fig. 1.

211 A self-dual non-commutative operator *seq*, notated \triangleleft , is introduced in order to model
 212 processes with action prefixes or sequential composition. Seq was first introduced in system
 213 BV [17], which was subsequently extended with the additives to obtain system MAV [20]. The
 214 operator *seq* lies between multiplicative operators *times* \otimes and *par* \wp from linear logic [15].

215 Inference rule *confine* and the *medial* rules are best explained in the context of examples
 216 throughout the remainder of this paper. Notice all medials have a standard form.

$$\frac{(P \sqcap R) \sqcup (Q \sqcap S)}{(P \sqcup Q) \sqcap (R \sqcup S)} \text{medial} \quad \text{where } (\sqcap, \sqcup) \in \left\{ (\wp, \oplus_q), (\&_p, \oplus_q), (\&, \oplus_q), (\&, \&_p), \right. \\
 \left. (\wp, \triangleleft), (\&, \triangleleft), (\&_p, \triangleleft), (\triangleleft, \oplus_q) \right\}$$

218 Cut elimination in the calculus of structures is equivalent to the following statement.

219 ► **Theorem 2** (cut elimination). *In ΔMAV , if $\vdash \mathcal{C}\{ P \otimes \bar{P} \}$, then $\vdash \mathcal{C}\{ \circ \}$.*

220 The above theorem is the main technical justification for the correctness of ΔMAV . A proof
 221 sketch is delayed until Section 5. As with MALL, linear implication $P \multimap Q$ is defined in
 222 terms of negation and *par* such that $\bar{P} \wp Q$. A useful but straightforward property is linear
 223 implication is reflexive. Amongst the immediate consequences of cut elimination is linear
 224 implication in ΔMAV is transitive. Furthermore, also as a corollary of cut elimination, linear
 225 implication holds in every context (note negation and implication are derived operators,
 226 hence are not part of the syntax of contexts).

227 ► **Corollary 3.** *Linear implication is a preorder that holds in every context (a precongruence).*

228 This corollary establishes a key criteria for using linear implication as a notion of refinement.

229 Note, in this paper, operator $\&_p$ is treated as a synthetic dual to \oplus_p necessary for complet-
 230 ing the proof system, and used when proving linear implications. This operator likely has
 231 applications, for modelling probabilistic communicating systems; but we avoid controversy
 232 by sticking to the indisputable established probabilistic choice modelled by \oplus_p .

233 3.3 Embedding of Probabilistic Processes in ΔMAV

234 While cut elimination proves we have made the correct choices of rules for the logic to work,
 235 it says little about its relationship to probabilistic refinement. Here we state the main result
 236 showing that implication is sound with respect to the key established notions of refinement
 237 for probabilistic processes.

238 We employ the following embedding, mapping processes to formulae.¹

Name of operator	Process term	Logical operator
success	$\llbracket \text{ok} \rrbracket$	\circ
prefix	$\llbracket \alpha.t \rrbracket$	$\alpha \triangleleft \llbracket t \rrbracket$
239 parallel composition	$\llbracket t_1 \parallel t_2 \rrbracket$	$\llbracket t_1 \rrbracket \otimes \llbracket t_2 \rrbracket$
external choice	$\llbracket t_1 \sqcap t_2 \rrbracket$	$\llbracket t_1 \rrbracket \& \llbracket t_2 \rrbracket$
probabilistic choice	$\llbracket t_1 +_p t_2 \rrbracket$	$\llbracket t_1 \rrbracket \oplus_p \llbracket t_2 \rrbracket$

240 The mapping extends to discrete probability distributions over process terms such that
 241 $\llbracket \mathbf{1}_t \rrbracket = \llbracket t \rrbracket$ and if $\Delta = p\Delta_1 + (1-p)\Delta_2$, where $0 < p < 1$ then $\llbracket \Delta \rrbracket = \llbracket \Delta_1 \rrbracket \oplus_p \llbracket \Delta_2 \rrbracket$.

242 Using the above embedding of processes as formulae we can compare processes using
 243 linear implication. All linear implications between processes can also be established using
 244 weak complete probabilistic simulation. Each approach is quite different, since the former
 245 involves unfolding logical rules while the latter involves defining a simulation relation wit-
 246 nessing the refinement. Here these two approaches to probabilistic refinement are formally
 247 connected as follows.

248 ► **Theorem 4.** *If $\vdash \llbracket t_1 \rrbracket \multimap \llbracket t_2 \rrbracket$, in ΔMAV , then t_1 simulates t_2 (Def. 1).*

249 The proof provides a procedure that constructs a weak complete probabilistic simulation
 250 from any linear implications between embeddings of processes. It adapts proof techniques
 251 devised for establishing a similar results for the π -calculus [22] (without probabilities).

252 The converse of Theorem 4 does not hold. As reinforced by related work [21], linear
 253 implication has non-interleaving properties. For example $a \wp a \multimap a \triangleleft a$ does **not** hold,

¹ Note the system is completely symmetric so the dual operators could be used, inverting implication.

254 but these processes are equivalent in any interleaving semantics, including probabilistic
 255 simulation in Def. 1. This can be regarded as a strength of linear implication, since such
 256 non-interleaving semantics are preserved under *action refinement* [41] — the substitution of
 257 an atomic action with any process. For the minimal process language in this this work, we
 258 consider only refinement of an action with a sequence of actions.

259 ► **Corollary 5.** *For process terms t_1 and t_2 , and substitution σ mapping actions, say a , to a*
 260 *sequence of actions, say $b_1 \dots b_n$, if $\vdash \llbracket t_1 \rrbracket \multimap \llbracket t_2 \rrbracket$ then $\vdash \llbracket t_1 \sigma \rrbracket \multimap \llbracket t_2 \sigma \rrbracket$.*

261 For example, since $\vdash \llbracket a \parallel a \rrbracket \multimap \llbracket a.a \rrbracket$ holds, by applying the action refinement $\sigma = \{b.c/a\}$, the
 262 following holds: $\vdash \llbracket b.c \parallel b.c \rrbracket \multimap \llbracket b.c.b.c \rrbracket$.

263 Action refinement is not respected by any interleaving semantics, including weak com-
 264 plete probabilistic simulation (previous work on action refinement in the probabilistic set-
 265 ting [8] avoids parallel composition). Furthermore, although there is work on probabilistic
 266 event structures [1, 42], linear implication in ΔMAV appears to be the first non-interleaving
 267 notion of refinement accommodating probabilistic choice.

268 4 Examples of properties established using linear implication

269 Having introduced definitions and stated the main results, we illustrate the theory with
 270 examples. This section covers examples of refinements that are permitted or forbidden
 271 between processes. There are also some examples justifying the medial rules.

272 4.1 Refinements also provable using probabilistic simulation

273 As noted in the introduction, projection and injection are forbidden for probabilistic simu-
 274 lation, hence should be forbidden for the sub-additives. Indeed, the following processes are
 275 unrelated by linear implication.

276 $heads +_{1/2} tails$ is unrelated to $heads$ and also is unrelated to $tails$

277 Hence, as a consequence of Theorem 4, **none** of the following hold in general: $P \multimap P \oplus_p Q$,
 278 $P \oplus_p Q \multimap P$, $Q \multimap P \oplus_p Q$ and $P \oplus_p Q \multimap Q$.

279 Now, using the rules of ΔMAV , we can verify the following chain of implications, proving
 280 that the probabilistic sub-additives lie between the standard additives.

$$281 \quad P \& Q \multimap P \&_p Q \qquad P \&_p Q \multimap P \oplus_p Q \qquad P \oplus_p Q \multimap P \oplus Q$$

282 The first implication has a proof of the following form.

$$283 \quad \frac{\frac{\frac{\frac{\circ}{\circ \&_p \circ} \text{idempotency}}{(\overline{P} \wp P) \&_p (\overline{Q} \wp Q)} \text{Proposition 3}}{((\overline{P} \oplus \overline{Q}) \wp P) \&_p ((\overline{P} \oplus \overline{Q}) \wp Q)} \text{choose}}{((\overline{P} \oplus \overline{Q}) \oplus_p (\overline{P} \oplus \overline{Q})) \wp (P \&_p Q)} \text{confine}}{(\overline{P} \oplus \overline{Q}) \wp (P \&_p Q)} \text{idempotency}$$

284 Also, due to de Morgan dualities, the third implication in the chain above has a proof of
 285 the same form (by setting P as \overline{P} and Q as \overline{Q}). The second implication in the chain of

286 implications above has the following proof.

$$\begin{array}{c}
\frac{\circ}{\circ \&_p \circ} \text{idempotency} \\
\frac{\frac{\frac{\circ}{\circ \&_p \circ} \text{idempotency}}{(\overline{P} \wp P) \&_p (\overline{Q} \wp Q)} \text{Proposition 3}}{(\overline{P} \&_p \overline{Q}) \wp (P \oplus_p Q)} \text{confine} \\
\frac{\frac{(\overline{P} \&_p \overline{Q}) \wp (P \oplus_p Q)}{(\overline{P} \oplus_p \overline{Q}) \wp (\circ \&_p \circ) \wp (P \oplus_p Q)} \text{confine}}{(\overline{P} \oplus_p \overline{Q}) \wp (P \oplus_p Q)} \text{idempotency}
\end{array}$$

288 Notice, by instantiating the above with process embeddings, $\vdash \llbracket t_1 \sqcap t_2 \rrbracket \multimap \llbracket t_1 \rrbracket \&_p \llbracket t_2 \rrbracket$ and
289 $\vdash \llbracket t_1 \rrbracket \&_p \llbracket t_2 \rrbracket \multimap \llbracket t_1 +_p t_2 \rrbracket$ hold. Hence, by Theorem 2, there is also a proof of the following.

$$290 \quad \vdash \llbracket t_1 \sqcap t_2 \rrbracket \multimap \llbracket t_1 +_p t_2 \rrbracket$$

291 As guaranteed by Theorem 4, the above linear implication can also be established by prob-
292 abilistic simulation. For example, process $a \sqcap b$ simulates $a +_p b$. This holds since \mathcal{R} such
293 that $a \mathcal{R} \mathbf{1}_{a \sqcap b}$, $b \mathcal{R} \mathbf{1}_{a \sqcap b}$, and $\text{ok} \mathcal{R} \mathbf{1}_{\text{ok}}$ defines a weak probabilistic simulation such that
294 $\llbracket a \&_p b \rrbracket \hat{\mathcal{R}} \llbracket a \sqcap b \rrbracket$. The converse does not hold since $a \sqcap b \xrightarrow{a} \mathbf{1}_{\text{ok}}$, which is a transition
295 that cannot be matched by distribution $p\mathbf{1}_a + (1-p)\mathbf{1}_b$. Hence, by Theorem 4, the converse
296 implication $P \oplus_p Q \multimap P \& Q$ also does **not** hold in general.

297 4.2 Distributivity properties, some forbidden others permitted

298 We highlight, quite subtly, that we must also forbid certain distributivity properties over
299 parallel composition. Operator \oplus_p forbids refinements that undesirably leak information.
300 For example, processes $(a \parallel c) +_p (b \parallel d)$ and $(a +_p b) \parallel (c +_p d)$ are unrelated by probabilistic
301 simulation. Therefore, by Theorem 4, the following are unrelated by linear implication.

$$302 \quad (a \otimes c) \oplus_p (b \otimes d) \quad \text{is unrelated to} \quad (a \oplus_p b) \otimes (c \oplus_p d)$$

303 However we should allow other refinements. For example, the semantics of ΔMAV , does
304 admit the following partial distributivity property, preserving all four possible combinations
305 of parallel actions.

$$306 \quad \vdash (a \oplus_p b) \otimes (c \oplus_q d) \multimap ((a \otimes c) \oplus_q (a \otimes d)) \oplus_p ((b \otimes c) \oplus_q (b \otimes d))$$

307 The above distributivity property is also respected by probabilistic simulation introduced
308 in Sec. 2. Observe, both $\delta(((a \parallel c) +_q (a \parallel d)) +_p ((b \parallel c) +_q (b \parallel d)))$ and $\delta((a +_p b) \parallel (c +_q d))$
309 map to the same underlying probability distribution, hence have the same behaviours.

$$310 \quad pq\mathbf{1}_{a \parallel c} + p(1-q)\mathbf{1}_{a \parallel d} + (1-p)q\mathbf{1}_{b \parallel c} + (1-p)(1-q)\mathbf{1}_{b \parallel d}$$

311 Indeed, in general, the following implication holds in ΔMAV , establishing how probabilistic
312 choice distributes over parallel composition.

$$313 \quad \vdash P \otimes (Q \oplus_p R) \multimap (P \otimes Q) \oplus_p (P \otimes R)$$

314 There are also distributivity properties relating non-deterministic and probabilistic choice [43].

315 For example we have that $\vdash (P \& Q) \oplus_p (P \& R) \multimap P \& (Q \oplus_p R)$ holds, as established by

344 provable without using any medial rules.

$$345 \quad (a_1 \wp a_2) \&_p ((b_1 \&_q (c \& d)) \wp (b_2 \oplus_q (c \& d))) \multimap (a_1 \&_p (b_1 \&_q (c \& d))) \wp (a_2 \oplus_p (b_2 \oplus_q (c \& d)))$$

346 Now, assuming $r = (1 - p)q$ and $p = s(1 - r)$, observe the following are equivalent by associativity and commutativity of the sub-additives.

$$348 \quad (a_1 \&_p (b_1 \&_q (c \& d))) \wp (a_2 \oplus_p (b_2 \oplus_q (c \& d))) \equiv (b_1 \&_r (a_1 \&_s (c \& d))) \wp (b_2 \oplus_r (a_2 \oplus_s (c \& d)))$$

349 Thirdly, observe the following implication is provable, without any medial rules.

$$350 \quad (b_1 \&_r (a_1 \&_s (c \& d))) \wp (b_2 \oplus_r (a_2 \oplus_s (c \& d))) \\ \multimap (b_1 \&_r ((a_1 \&_s c) \& (a_1 \&_s d))) \wp (b_2 \oplus_r ((a_2 \oplus_s c) \& (a_2 \oplus_s d)))$$

351 Now, assuming cut elimination holds, combining the above three observations, necessarily,
352 we can construct a cut-free proof of the following implication.

$$353 \quad (a_1 \wp a_2) \&_p ((b_1 \&_q (c \& d)) \wp (b_2 \oplus_q (c \& d))) \\ \multimap (b_1 \&_r ((a_1 \&_s c) \& (a_1 \&_s d))) \wp (b_2 \oplus_r ((a_2 \oplus_s c) \& (a_2 \oplus_s d)))$$

354 Unfortunately, the above implication is not provable without medial rules. Specifically, we
355 require medial rules commuting the sub-additives over *with* in order to establish the proof
356 of the above implication. This example is extracted from exactly where the cut elimination
357 would fail if the medial rules are omitted. Thus the medial rules are not a design decision,
358 but necessary in order for cut-elimination to hold.

359 5 On the proof of cut-elimination (Theorem 2)

360 Proving proof normalisation results involves extensive case analysis; hence we provide only
361 a sketch proof of cut elimination proof for ΔMAV . The interesting point is that the idem-
362 potency of sub-additives is problematic, giving rise to infinite derivations. For example,
363 formula $a \oplus b$ has infinitely many premises, including those of the form $a \&_{1/2-1/2^n} (a \oplus b)$.

364 To handle such problems caused by idempotency in the cut elimination proof we move to
365 a semantically equivalent but more controlled version of ΔMAV , turning *idempotency*, from
366 an equivalence into the following inference rules.

$$367 \quad \frac{\mathcal{C}\{R \oplus_p R\}}{\mathcal{C}\{R\}} \text{ contract} \quad \frac{\mathcal{C}\{\circ\}}{\mathcal{C}\{\circ \&_p \circ\}} \text{ tidy distribution} \quad \frac{\mathcal{C}\{P \&_r Q\}}{\mathcal{C}\{P \oplus_r Q\}} \text{ special case of confine}$$

368 The proof of cut-elimination (Theorem 2) proceeds by, firstly, observing rule $\frac{P \otimes \bar{P}}{\circ} \text{ cut}$
369 can be broken down to its atomic form *co-interact* using the following *co-rules*.

$$370 \quad \frac{\mathcal{C}\{(P \oplus R) \otimes (Q \& S)\}}{\mathcal{C}\{(P \otimes Q) \oplus (R \otimes S)\}} \text{ co-additives} \quad \frac{\mathcal{C}\{(P \oplus_p Q) \otimes (R \&_p S)\}}{\mathcal{C}\{(P \otimes R) \oplus_p (Q \otimes S)\}} \text{ co-confine}$$

$$371 \quad \frac{\mathcal{C}\{\circ \oplus \circ\}}{\mathcal{C}\{\circ\}} \text{ co-tidy} \quad \frac{\mathcal{C}\{a \otimes \bar{a}\}}{\mathcal{C}\{\circ\}} \text{ co-interact} \quad \frac{\mathcal{C}\{P\}}{\mathcal{C}\{P \&_p P\}} \text{ co-contract}$$

$$372 \quad \frac{\mathcal{C}\{(P \sqcap R) \sqcup (Q \sqcap S)\}}{\mathcal{C}\{(P \sqcup Q) \sqcap (R \sqcup S)\}} \text{ medial} \quad \text{where } (\sqcap, \sqcup) \in \{(\&_q, \otimes), (\&_q, \oplus), (\oplus_p, \oplus), (\triangleleft, \otimes)\}$$

373

374 We then proceed by the following strategy to show all such co-rules can be eliminated.
 375 We firstly apply a technique called decomposition [18, 39, 40], showing instances of the
 376 problematic *contract* rule can be pushed to the bottom of a proof. This involves introducing
 377 further *co-rules*, notably the rule *co-contract*, which is pushed to the top of the proof. The
 378 technical challenge with decomposition is devising a measure controlling explosions in the
 379 size of the proof, based on the topology of the proof, caused by permuting contractions with
 380 co-contractions.

381 ► **Lemma 6** (decomposition). *For any derivation $\frac{S}{P}$, including co-rules, there exists Q and*

$$\frac{S}{R}$$
using co-contract only
 382
$$\frac{Q}{P}$$
using contract only

383 Notice, when decomposition is applied to a proof, which must have premise \circ , the *co-contract*
 384 rules disappear, becoming instances of *tidy distribution*. This way, we transform a proof of
 385 P into a proof of some formula Q which does not use *contract* or *co-contract* rules, such
 386 that Q is reachable from P using only the *contract* rule. For the proof of Q , that does not
 387 use *contract* or *co-contract* rules, we can apply a technique called *splitting* [19]. Splitting
 388 generalises the effect of applying rules in sequent-like contexts.

389 ► **Lemma 7** (splitting). *In the following, killing contexts are multi-hole contexts defined by*
 390 *grammar $\mathcal{T}\{ \} ::= \{ \cdot \} \mid \mathcal{T}\{ \} \& \mathcal{T}\{ \}$. The following hold in ΔMAV without contract, but*
 391 *with tidy distribution and the special case of confine:*

392 ■ *If $\vdash (P \& Q) \wp R$ then $\vdash P \wp R$ and $\vdash Q \wp R$.*

393 ■ *If $\vdash (P \&_p Q) \wp R$, there exist U, V such that $\frac{U \oplus_p V}{R}$ and both $\vdash P \wp U$ and $\vdash Q \wp V$ hold.*

394 ■ *If $\vdash (P \oplus_p Q) \wp R$, there exist U, V such that $\frac{U \&_p V}{R}$ and both $\vdash P \wp U$ and $\vdash Q \wp V$ hold.*

395 ■ *If $\vdash (P \triangleleft Q) \wp R$, there exist $\mathcal{T}\{ \}$, U_i and V_i such that $\frac{\mathcal{T}\{ U_i \triangleleft V_i \}}{R}$ and, for all i , both*
 396 *$\vdash P \wp U_i$ and $\vdash Q \wp V_i$ hold.*

397 ■ *If $\vdash (P \otimes Q) \wp R$, there exist $\mathcal{T}\{ \}$, U_i and V_i such that $\frac{\mathcal{T}\{ U_i \wp V_i \}}{R}$ and, for all i ,*
 398 *$\vdash P \wp U_i$ and $\vdash Q \wp V_i$.*

399 ■ *If $\vdash (P \oplus Q) \wp R$ then, there exist W_i such that $\frac{\mathcal{T}\{ W_i \}}{R}$ and, for all i , either $\vdash P \wp W_i$*
 400 *or $\vdash Q \wp W_i$ hold.*

401 ■ *If $\vdash a \wp R$ then $\frac{\mathcal{T}\{ \bar{a} \}}{R}$.*

402 ■ *If $\vdash \bar{a} \wp R$ then $\frac{\mathcal{T}\{ a \}}{R}$.*

403 Splitting is then used to extended sequent-like contexts to any context.

404 ► **Lemma 8** (context reduction). *If, for all R , $\vdash P \wp R$ yields $\vdash Q \wp R$ then, for all contexts*
 405 *$\mathcal{C}\{ \}$, $\vdash \mathcal{C}\{ P \}$ yields $\vdash \mathcal{C}\{ Q \}$.*

406 By using splitting and context reduction, the co-rules previously introduced in this sec-
 407 tion are shown to be admissible, which together show cut is admissible in the fragment
 408 without contraction. The first three co-rule elimination lemmas concern only connectives of
 409 MALL [20].

410 ▶ **Lemma 9.** *If $\vdash \mathcal{C}\{ \circ \oplus \circ \}$ then $\vdash \mathcal{C}\{ \circ \}$.*

411 ▶ **Lemma 10.** *If $\vdash \mathcal{C}\{ (P \oplus Q) \otimes (R \& S) \}$ holds, then it holds that $\vdash \mathcal{C}\{ (P \otimes R) \oplus (Q \otimes S) \}$.*

412 ▶ **Lemma 11.** *If $\vdash \mathcal{C}\{ a \otimes \bar{a} \}$ then $\vdash \mathcal{C}\{ \circ \}$, for any atom a .*

413 The following co-rule elimination lemma involves the probabilistic sub-additives.

414 ▶ **Lemma 12.** *If $\vdash \mathcal{C}\{ (P \oplus_p Q) \otimes (R \&_p S) \}$ holds, $\vdash \mathcal{C}\{ (P \otimes R) \oplus_p (Q \otimes S) \}$ holds.*

415 The four extra *medial* rules can also be eliminated.

416 ▶ **Lemma 13.** *For any $(\sqcap, \sqcup) \in \{(\&_q, \otimes), (\&_q, \oplus), (\oplus_p, \oplus), (\triangleleft, \otimes)\}$, if $\vdash \mathcal{C}\{ (P \sqcap R) \sqcup (Q \sqcap S) \}$*
 417 *then $\vdash \mathcal{C}\{ (P \sqcup Q) \sqcap (R \sqcup S) \}$.*

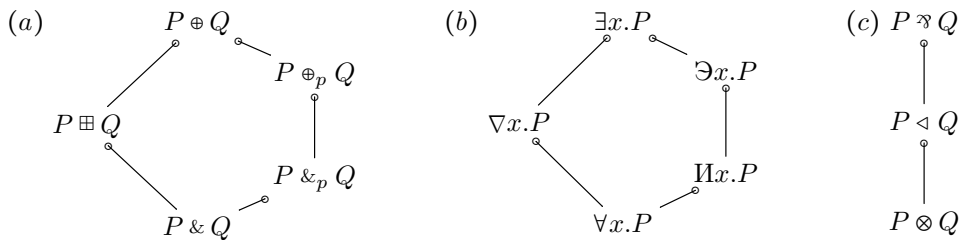
418 We can now establish cut elimination for the proof system described at the beginning of
 419 this section, without *idempotency*, but with three inference rules: *contract*, *tidy distribution*
 420 and the *special case of confine*. Having applied decomposition (Lemma 6) to push *contract*
 421 to the bottom of the proof, the proof combines the above lemmas to remove each *co-rule*.
 422 This leaves a system without *co-rules*.

423 Finally, we obtain our main result (Theorem 2): cut elimination in the more controlled
 424 system implies cut elimination in ΔMAV , simply by substituting *contract*, *tidy distribution*
 425 and the *special case of confine* with instances of *idempotency* and *confine*.

426 6 Related work on Sub-Additive Operators and Nominal Quantifiers

427 Between the standard additives of multiplicative linear logic, *with* and *plus*, there are further
 428 sub-additive operators. Roversi [35] proposed a sub-additive operator, say \boxplus , also forbidding
 429 projection and injection, that is self-dual. Note a self-dual operator is such that the linear
 430 negation of $P \boxplus Q$ is $\bar{P} \boxplus \bar{Q}$, i.e., the operator is de Morgan dual to itself.

431 Such a self-dual sub-additive operator cannot be used to model probabilistic choice in
 432 the processes-as-formulae paradigm. The problem is the following implication is provable
 433 $(a \boxplus b) \otimes (c \boxplus d) \multimap (a \otimes c) \boxplus (b \otimes d)$. Consequently, self-dual sub-additives are unsound with
 434 respect to probabilistic simulation (notice the possibility of $a \otimes d$ or $b \otimes c$ occurring has
 435 been excluded in the formula on the right). The pair of probabilistic sub-additives $\&_p$ and
 436 \oplus_p , were discovered by seeking more controlled variants of \boxplus such that the above unsound
 437 distributivity property is **forbidden**.



438 ■ **Figure 3** Relationships between various operators in extensions of linear logic: (a) the additives
 439 and sub-additives, (b) the first-order quantifiers and nominal quantifiers, (c) the multiplicatives.

438 Figure 3(a) compares additives $\&$, $\&_p$, \oplus_p , \oplus and \boxplus . Notice similarities with Fig 3(b)
 439 depicting de Morgan dual pair of nominal quantifiers, $\exists x.P$ and $\nabla x.P$, located between *for*

440 *all* and *exists* [23]. Similarly, to the sub-additives, the justification for the pair of nominal
 441 quantifiers, rather than a self-dual nominal quantifier [14, 34, 35], say $\nabla x.P$, was to soundly
 442 model private names in direct logical embeddings of π -calculus processes [32].

443 Related work at the intersection of linear logic and probabilistic programs is typically
 444 denotational (of a model theoretic flavour). For example, *probabilistic coherence spaces* [16,
 445 12] provide a *probabilistic denotational semantics* [26, 7] for linear logic but with standard
 446 additives *with* and *plus* only. Probabilistic coherence spaces and related models are typically
 447 used directly to provide a semantics for functional probabilistic programming languages, such
 448 as PCF with random number generators [13, 6] or a probabilistic λ -calculus [29]. However,
 449 probabilistic extensions of linear logic itself, giving rise to probabilistic sub-additives sound
 450 with respect the probabilistic choice in process calculi, have not previously been investigated.

451 **7 Conclusion**

452 This paper exposes an extended *syntax* and proof system for linear logic with explicit prob-
 453 abilistic choice operators. The rules for these *sub-additives* are determined by studying a
 454 generalisation of *cut elimination* (Theorem 2), leaving no room for design decisions. When
 455 designing process preorders, we are confronted by a vast design space. Thus Δ MAV (Fig. 2)
 456 can assist objectively with resolving language design decisions. I argue linear implication is
 457 a compelling notion of probabilistic refinement, being sound with respect to *weak (complete)*
 458 *probabilistic simulation* (Theorem 4), hence also *probabilistic may testing*. Furthermore, lin-
 459 ear implication has the advantage that it is the coarsest notion of refinement for probabilistic
 460 concurrent processes in the literature respecting action refinement (Corollary 5).

461 Interestingly, the proof of cut elimination demands a technique called *decomposition*,
 462 Lemma 6, to handle idempotency of choice, which, previously, has only been *necessary* for
 463 handling modalities in non-commutative logic NEL [39, 19]. Details of the proof theory are
 464 reserved for an extended version.

465 Future work includes explaining the connections between the quantitative modal logics,
 466 such as the quantitative modal μ -calculus [25], and Δ MAV. Future work may also consider
 467 richer process models in Δ MAV and its extensions [24]. For example, by using positive and
 468 negative atoms to model inputs and outputs [3, 22], we can model probabilistic calculi with
 469 communication. A related question is whether the operator $\&_p$ is useful when modelling
 470 processes. Recall $\&_p$ was discovered, synthetically, as the operator de Morgan dual to prob-
 471 abilistic choice \oplus_p . To help understand the nature of $\&_p$, observe that it is related to \oplus_p
 472 in a similar fashion that, in the internal π -calculus [36], fresh name binding ν is related to
 473 internal input (which receives a name, but only if it is fresh). By using this analogy, $\&_p$
 474 can model branches of an input that preserves a probability distribution by using knowledge
 475 of the probability distribution over branches with which it interacts (perhaps by measuring
 476 previous interactions with a controller, for example), and only interacts if the distribution
 477 matches the criteria specified by the internal choice (as suggested by rule *confine*). Such con-
 478 straints could be useful for preventing systems from being composed whenever the random
 479 behaviour of one component falls out of expected bounds of another component (possibly
 480 causing the component that receives messages on a random channel from failing to meet its
 481 specification). Considering possible connections between $\&_p/\oplus_p$ and angelic/daemonic prob-
 482 abilistic choices [31] is also future work. To help the reader digest this novel theory, initially,
 483 only simple and indisputable core process models are discussed in the current paper.

484 — **References** —

- 485 **1** Samy Abbes and Albert Benveniste. True-concurrency probabilistic models: Branching
486 cells and distributed probabilities for event structures. *Information and Computation*,
487 204(2):231–274, 2006. doi:10.1016/j.ic.2005.10.001.
- 488 **2** Christel Baier and Holger Hermanns. Weak bisimulation for fully probabilistic pro-
489 cesses. In *Computer Aided Verification*, pages 119–130. Springer, 1997. doi:10.1007/
490 3-540-63166-6_14.
- 491 **3** Paola Bruscoli. A purely logical account of sequentiality in proof search. In *ICLP*, volume
492 2401 of *LNCS*, pages 302–316. Springer, 2002. doi:10.1007/3-540-45619-8_21.
- 493 **4** Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. The focused calculus of struc-
494 tures. In *CSL*, volume 12, pages 159–173, 2011. doi:10.4230/LIPIcs.CSL.2011.159.
- 495 **5** Gabriel Ciobanu and Ross Horne. Behavioural analysis of sessions using the calculus of
496 structures. In *PSI 2015, 25-27 August, Kazan, Russia*, volume 9609 of *LNCS*, pages 91–106,
497 2015. doi:10.1007/978-3-319-41579-6_8.
- 498 **6** Ugo Dal Lago, Claudia Faggian, Benoît Valiron, and Akira Yoshimizu. The geometry
499 of parallelism: Classical, probabilistic, and quantum effects. In *Proceedings of the 44th
500 ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL 2017, pages
501 833–845. ACM, 2017. doi:10.1145/3009837.3009859.
- 502 **7** Vincent Danos and Russell S. Harmer. Probabilistic game semantics. *ACM Trans. Comp.*
503 *Logic*, 3(3):359–382, July 2002. doi:10.1145/507382.507385.
- 504 **8** Jerry den Hartog, Erik P. de Vink, and Jacobus W. De Bakker. Metric semantics and full
505 abstractness for action refinement and probabilistic choice. *Electronic Notes in Theoretical
506 Computer Science*, 40:72–99, 2001. doi:10.1016/S1571-0661(05)80038-6.
- 507 **9** Yuxin Deng. *Semantics of Probabilistic Processes: An Operational Approach*. Springer,
508 2015. doi:10.1007/978-3-662-45198-4.
- 509 **10** Yuxin Deng, Matthew Hennessy, Rob van Glabbeek, and Carroll Morgan. Characterising
510 testing preorders for finite probabilistic processes. In *22nd Annual IEEE Symposium on
511 Logic in Computer Science (LICS 2007)*, pages 313–325, July 2007. doi:10.1109/LICS.
512 2007.15.
- 513 **11** Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Characterising
514 testing preorders for finite probabilistic processes. *Logical Methods in Computer Science*,
515 4(4), 2008. doi:10.2168/LMCS-4(4:4)2008.
- 516 **12** Thomas Ehrhard, Michele Pagani, and Christine Tasson. The computational meaning of
517 probabilistic coherence spaces. In *LICS*, pages 87–96. IEEE, 2011. doi:10.1109/LICS.
518 2011.29.
- 519 **13** Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces
520 are fully abstract for probabilistic PCF. In *Proc. POPL*, 49(1):309–320, 2014. doi:10.
521 1145/2535838.2535865.
- 522 **14** Andrew Gacek, Dale Miller, and Gopalan Nadathur. Nominal abstraction. *Information
523 and Computation*, 209(1):48–73, 2011. doi:10.1016/j.ic.2010.09.004.
- 524 **15** Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–112, 1987. doi:
525 10.1016/0304-3975(87)90045-4.
- 526 **16** Jean-Yves Girard. Between logic and quantic: a tract. *Linear logic in computer science*,
527 316:346–381, 2004. doi:10.1017/CBO9780511550850.011.
- 528 **17** Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Compu-
529 tational Logic*, 8(1), 2007. doi:10.1145/1182613.1182614.
- 530 **18** Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic
531 flows. *Logical Methods in Computer Science*, 4(1), 2008. doi:10.2168/LMCS-4(1:9)2008.

- 532 **19** Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure V: the
533 exponentials and splitting. *Mathematical Structures in Computer Science*, 21(3):563–584,
534 2011. doi:10.1017/S096012951100003X.
- 535 **20** Ross Horne. The consistency and complexity of multiplicative additive system virtual. *Sci.*
536 *Ann. Comp. Sci.*, 25(2):245–316, 2015. doi:10.7561/SACS.2015.2.245.
- 537 **21** Ross Horne, Sjouke Mauw, and Alwen Tiu. Semantics for specialising attack trees based on
538 linear logic. *Fundamenta Informaticae*, 153(1-2):57–86, 2017. doi:10.3233/FI-2017-1531.
- 539 **22** Ross Horne and Alwen Tiu. Constructing weak simulations from linear implications for
540 processes with private names. *Mathematical Structure in Computer Science*, pages 1–34,
541 2019. doi:10.1017/S0960129518000452.
- 542 **23** Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. Private names in non-
543 commutative logic. In *CONCUR 2016*, pages 31:1–31:16. LIPIcs, 2016. doi:10.4230/
544 LIPIcs.CONCUR.2016.31.
- 545 **24** Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. De morgan dual nom-
546 inal quantifiers modelling private names in non-commutative logic. *ACM Transactions on*
547 *Computational Logic (TOCL)*, 2019. in press.
- 548 **25** Michael Huth and Marta Z. Kwiatkowska. Quantitative analysis and model checking. In
549 *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages
550 111–122, 1997. doi:10.1109/LICS.1997.614940.
- 551 **26** Cliff Jones and Gordon Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings*
552 *of the Fourth Annual Symposium on Logic in Computer Science*, pages 186–195, June 1989.
553 doi:10.1109/LICS.1989.39173.
- 554 **27** Bengt Jonsson and Wang Yi. Compositional testing preorders for probabilistic processes.
555 In *LICS*, pages 431–441. IEEE, 1995. doi:10.1109/LICS.1995.523277.
- 556 **28** Bengt Jonsson, Wang Yi, and Kim G Larsen. Probabilistic extensions of process algebras.
557 *Handbook of process algebra*, pages 685–710, 2001. doi:10.1016/B978-044482830-9/
558 50029-1.
- 559 **29** Ugo Dal Lago and Margherita Zorzi. Probabilistic operational semantics for the lambda
560 calculus. *RAIRO - Theoretical Informatics and Applications*, 46(3):413–450, 2012. doi:
561 10.1051/ita/2012012.
- 562 **30** Kim G Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and*
563 *computation*, 94(1):1–28, 1991. doi:10.1016/0890-5401(91)90030-6.
- 564 **31** Annabelle McIver and Carroll Morgan. *Abstraction, Refinement And Proof For Probabilistic*
565 *Systems (Monographs in Computer Science)*. SpringerVerlag, 2004. doi:10.1007/b138392.
- 566 **32** Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I and II.
567 *Information and Computation*, 100(1):1–77, 1992. doi:10.1016/0890-5401(92)90008-4.
- 568 **33** Catuscia Palamidessi and Oltea Mihaela Herescu. A randomized encoding of the π -calculus
569 with mixed choice. *Theoretical Computer Science*, 335(2-3):373–404, 2005. doi:10.1016/
570 j.tcs.2004.11.020.
- 571 **34** Andrew Pitts. Nominal logic, a first order theory of names and binding. *Information and*
572 *Computation*, 186(2), 2003. doi:10.1016/S0890-5401(03)00138-X.
- 573 **35** Luca Roversi. A deep inference system with a self-dual binder which is complete for linear
574 lambda calculus. *J. of Logic and Computation*, 26(2):677–698, 2016. doi:10.1093/logcom/
575 exu033.
- 576 **36** Davide Sangiorgi. pi-calculus, internal mobility, and agent-passing calculi. *Theor. Comput.*
577 *Sci.*, 167(1&2):235–274, 1996. doi:10.1016/0304-3975(96)00075-8.
- 578 **37** Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes.
579 *Nordic Journal of Computing*, 2(2):250–273, 1995. doi:10.1007/3-540-56454-3_13.
- 580 **38** Lutz Straßburger. A local system for linear logic. In *LPAR*, volume 2514, pages 388–402.
581 Springer, 2002. doi:10.1007/3-540-36078-6_26.

- 582 **39** Lutz Straßburger and Alessio Guglielmi. A system of interaction and structure IV: the
583 exponentials and decomposition. *ACM T. Comp. Log.*, 12(4):23:1–39, 2011. doi:10.1145/
584 1970398.1970399.
- 585 **40** Andrea Aler Tubella, Alessio Guglielmi, and Benjamin Ralph. Removing cycles from proofs.
586 In *CSL*, volume 82 of *LIPICs*, pages 9:1–9:17, 2017. doi:10.4230/LIPICs.CSL.2017.9.
- 587 **41** Rob van Glabbeek and Ursula Goltz. Refinement of actions and equivalence notions for con-
588 current systems. *Acta Informatica*, 37(4-5):229–327, 2001. doi:10.1007/s002360000041.
- 589 **42** Daniele Varacca, Hagen Völzer, and Glynn Winskel. Probabilistic event structures and
590 domains. In *CONCUR*, pages 481–496, 2004. doi:10.1007/978-3-540-28644-8_31.
- 591 **43** Daniele Varacca and Glynn Winskel. Distributing probability over non-determinism.
592 *Mathematical Structures in Computer Science*, 16(1):87–113, 2006. doi:10.1017/
593 S0960129505005074.