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

Ross Horne
Computer Science and Communications, University of Luxembourg, Esch-sur-Alzette, Luxembourg
ross.horne@uni.lu

Abstract

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.

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

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

Digital Object Identifier 10.4230/LIPIcs.FSCD.2019.23

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

1 Introduction

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 uncertainly in the environment, such as a decision made by tossing a coin.

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*.

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*.

4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019).
Editor: Herman Geuvers; Article No. 23; pp. 23:1–23:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1 The processes-as-formulae paradigm

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

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

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

1.2 Motivation: uncovering the probabilistic sub-additive operators

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

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

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

Although projection/injection are forbidden, non-deterministic choice and probabilistic choice are related. For example, non-deterministic choice $\mathit{heads} \,\&\, \mathit{tails}$ can be refined to probabilistic choice $\mathit{heads} \oplus_{1/2} \mathit{tails}$. This refinement can be established by proving the following using the logical system in the body of this work.

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

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

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

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

Outline of the paper. Section 2, provides established background material on probabilistic processes. Section 3, recalls MALL in the calculus of structures, and introduces the extended system $\Delta\mathsf{MAV}$ featuring a pair of sub-additive operators. Section 4 provides a series of examples illustrating how we can construct, more traditional, probabilistic simulations from proofs in $\Delta\mathsf{MAV}$. Section 5, outlines the proof of cut elimination, necessary to justify the logical system proposed. Section 6 highlights the existence of further sub-additive operators between the standard operators of linear logic.

2 Background: an established notion of probabilistic simulation

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

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

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

$$\begin{array}{rcl} t & ::= & \texttt{ok}\ \text{(successful completion)} \mid a.t\ \text{(action prefix)} \mid t \parallel t\ \text{(parallel composition)} \\ & \mid & t \sqcap t\ \text{(non-deterministic choice)} \mid t +_p t\ \text{(probabilistic choice)} \end{array}$$

Discrete probability distributions are uniquely determined by a *probability mass function* $\Delta : S \to [0,1]$ over a set S of process terms such that $\sum_{t \in S} \Delta(t) = 1$. A Dirac distribution for process term s, written $\mathbf{1}_s$, is defined by the probability mass function such that $\Delta(s) = 1$. For probability p and distributions, Δ_1 and Δ_2 *linear combination* $p\Delta_1 + (1-p)\Delta_2$, defined 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 defined such that $(\Delta_1 \cdot \Delta_2)(t \parallel u) = \Delta_1(t)\Delta_2(u)$ and 0 elsewhere.

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

$$\delta(\mathtt{ok}) = \mathbf{1}_{\mathtt{ok}} \qquad \delta(a.t) = \mathbf{1}_{a.t} \qquad \delta(t \sqcap t) = \mathbf{1}_{t \sqcap t}$$
$$\delta(t +_p t) = p\delta(t) + (1-p)\delta(t) \qquad \delta(t \parallel t) = \delta(t) \cdot \delta(t)$$

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

$$\frac{}{a.t \xrightarrow{a} \delta(t)} \qquad \frac{i \in \{1,2\}}{t_1 \sqcap t_2 \xrightarrow{\tau} \delta(t_i)} \qquad \frac{t_1 \xrightarrow{\alpha} \Delta}{t_1 \parallel t_2 \xrightarrow{\alpha} \Delta \cdot \delta(t_2)} \qquad \frac{t_2 \xrightarrow{\alpha} \Delta}{t_1 \parallel t_2 \xrightarrow{\alpha} \delta(t_1) \cdot \Delta}$$

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

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

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

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

- *If $s \mathrel{\mathcal{R}} \Delta$ and $s \xrightarrow{\alpha} \mathcal{E}$, there exists $\mathcal{E}'$ such that $\Delta \overset{\alpha}{\Longrightarrow} \mathcal{E}'$ and $\mathcal{E} \mathrel{\hat{\mathcal{R}}} \mathcal{E}'$.*
- *If $t \mathrel{\mathcal{R}} \Delta$ and $t\checkmark$ then there exists $\mathcal{E}$ such that $\Delta \overset{\tau}{\Longrightarrow} \mathcal{E}$ and $\mathcal{E}\checkmark$.*

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

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

3 Extending linear logic with probabilistic sub-additive operators

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

3.1 An established presentation of MALL in the calculus of structures

The fragment of linear logic MALL was one of the first proof systems studied in the calculus of structures [38]. Fig 1 recalls a proof system for multiplicative-additive linear logic MALL in the calculus of structures. Inference rules apply in any context. We assume formulae are always in negation-normal-form, where negation is always pushed to atoms, a, by the following function, inducing De Morgan dualities.

$$\overline{P \oplus Q} = \overline{P} \mathbin{\&} \overline{Q} \qquad \overline{P \mathbin{\&} Q} = \overline{P} \oplus \overline{Q} \qquad \overline{\overline{a}} = a \qquad \overline{P \otimes Q} = \overline{P} \mathbin{⅋} \overline{Q} \qquad \overline{P \mathbin{⅋} Q} = \overline{P} \otimes \overline{Q} \qquad \overline{\circ} = \circ$$

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

structural congruence:

$$P \parr Q \equiv Q \parr P \qquad (P \parr Q) \parr R \equiv P \parr (Q \parr R) \qquad \circ \parr P \equiv P$$

$$P \otimes Q \equiv Q \otimes P \qquad (P \otimes Q) \otimes R \equiv P \otimes (Q \otimes R) \qquad \circ \otimes P \equiv P$$

inference rules:

$$\frac{\mathcal{C}\{\ \circ\ \}}{\mathcal{C}\{\ \overline{a} \parr a\ \}}\ \text{interact} \qquad \frac{\mathcal{C}\{\ (P \parr Q) \otimes R\ \}}{\mathcal{C}\{\ P \parr (Q \otimes R)\ \}}\ \text{switch} \qquad \frac{\mathcal{C}\{\ \circ\ \}}{\mathcal{C}\{\ \circ \mathbin{\&} \circ\ \}}\ \text{tidy}$$

$$\frac{\mathcal{C}\{\ P_1\ \}}{\mathcal{C}\{\ P_1 \oplus P_2\ \}}\ \text{choose left} \qquad \frac{\mathcal{C}\{\ P_2\ \}}{\mathcal{C}\{\ P_1 \oplus P_2\ \}}\ \text{choose right} \qquad \frac{\mathcal{C}\{\ (P \parr R) \mathbin{\&} (Q \parr R)\ \}}{\mathcal{C}\{\ (P \mathbin{\&} Q) \parr R\ \}}\ \text{external}$$

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

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

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

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

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

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

The rule of the structural congruence for the probabilistic sub-additives, Fig. 2, ensures the balance of probabilities is maintained when applying idempotency, associativity and commutativity. By maintaining the balance of probabilities, structural congruence preserves underlying probability distributions. For example $p\Delta + (1-p)\Delta = \Delta$, hence we have a weighted form of idempotency $P \oplus_p P = P$.

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

structural congruence:

$$P \mathbin{\&_r} Q \equiv Q \mathbin{\&_{1-r}} P \qquad P \mathbin{\&_r} P \equiv P \qquad (P \mathbin{\&_p} Q) \mathbin{\&_q} R \equiv P \mathbin{\&_{pq}} \left(Q \mathbin{\&_{\frac{q(1-p)}{1-pq}}} R\right)$$

$$P \oplus_r Q \equiv Q \oplus_{1-r} P \qquad P \oplus_r P \equiv P \qquad (P \oplus_p Q) \oplus_q R \equiv P \oplus_{pq} \left(Q \oplus_{\frac{q(1-p)}{1-pq}} R\right)$$

$$\circ \triangleleft P \equiv P \qquad P \equiv P \triangleleft \circ \qquad (P \triangleleft Q) \triangleleft R \equiv P \triangleleft (Q \triangleleft R)$$

inference rules:

$$\frac{\mathcal{C}\{\ (P \parr R) \mathbin{\&_p} (Q \parr S)\ \}}{\mathcal{C}\{\ (P \oplus_p Q) \parr (R \mathbin{\&_p} S)\ \}}\ \text{confine}$$

$$\frac{\mathcal{C}\{\ (P \parr R) \oplus_q (Q \parr S)\ \}}{\mathcal{C}\{\ (P \oplus_q Q) \parr (R \oplus_q S)\ \}}\ \text{medial} \qquad \frac{\mathcal{C}\{\ (P \mathbin{\&_p} R) \oplus_q (Q \mathbin{\&_p} S)\ \}}{\mathcal{C}\{\ (P \oplus_q Q) \mathbin{\&_p} (R \oplus_q S)\ \}}\ \text{medial}$$

$$\frac{\mathcal{C}\{\ (P \mathbin{\&} R) \oplus_q (Q \mathbin{\&} S)\ \}}{\mathcal{C}\{\ (P \oplus_q Q) \mathbin{\&} (R \oplus_q S)\ \}}\ \text{medial} \qquad \frac{\mathcal{C}\{\ (P \mathbin{\&} R) \mathbin{\&_p} (Q \mathbin{\&} S)\ \}}{\mathcal{C}\{\ (P \mathbin{\&_p} Q) \mathbin{\&} (R \mathbin{\&_p} S)\ \}}\ \text{medial}$$

$$\frac{\mathcal{C}\{\ (P \parr R) \triangleleft (Q \parr S)\ \}}{\mathcal{C}\{\ (P \triangleleft Q) \parr (R \triangleleft S)\ \}}\ \text{medial} \qquad \frac{\mathcal{C}\{\ (P \mathbin{\&} R) \triangleleft (Q \mathbin{\&} S)\ \}}{\mathcal{C}\{\ (P \triangleleft Q) \mathbin{\&} (R \triangleleft S)\ \}}\ \text{medial}$$

$$\frac{\mathcal{C}\{\ (P \mathbin{\&_p} R) \triangleleft (Q \mathbin{\&_p} S)\ \}}{\mathcal{C}\{\ (P \triangleleft Q) \mathbin{\&_p} (R \triangleleft S)\ \}}\ \text{medial} \qquad \frac{\mathcal{C}\{\ (P \triangleleft R) \oplus_p (Q \triangleleft S)\ \}}{\mathcal{C}\{\ (P \oplus_p Q) \triangleleft (R \oplus_p S)\ \}}\ \text{medial}$$

linear negation:

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

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

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

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

$$\frac{(P \sqcap R) \sqcup (Q \sqcap S)}{(P \sqcup Q) \sqcap (R \sqcup S)}\ \text{medial} \qquad \text{where } (\sqcap, \sqcup) \in \{\ (\parr, \oplus_q), (\&_p, \oplus_q), (\&, \oplus_q), (\&, \&_p), (\parr, \triangleleft), (\&, \triangleleft), (\&_p, \triangleleft), (\triangleleft, \oplus_q)\ \}$$

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

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

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

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

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

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

3.3 Embedding of Probabilistic Processes in ΔMAV

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

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

Name of operator	Process term	Logical operator
success	$[\![\mathtt{ok}]\!]$	$\circ$
prefix	$[\![\alpha.t]\!]$	$\alpha \triangleleft [\![t]\!]$
parallel composition	$[\![t_1 \parallel t_2]\!]$	$[\![t_1]\!] \otimes [\![t_2]\!]$
external choice	$[\![t_1 \sqcap t_2]\!]$	$[\![t_1]\!] \& [\![t_2]\!]$
probabilistic choice	$[\![t_1 +_p t_2]\!]$	$[\![t_1]\!] \oplus_p [\![t_2]\!]$

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

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

▶ **Theorem 4.** *If* $\vdash [\![t_1]\!] \multimap [\![t_2]\!]$, *in* ΔMAV, *then* t_1 *simulates* t_2 *(Def. 1).*

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

The converse of Theorem 4 does not hold. As reinforced by related work [21], linear implication has non-interleaving properties. For example $a \parr a \multimap a \triangleleft a$ does **not** hold, but these processes are equivalent in any interleaving semantics, including probabilistic simulation in Def. 1. This can be regarded as a strength of linear implication, since such non-interleaving semantics are preserved under *action refinement* [41] – the substitution of an atomic action with any process. For the minimal process language in this this work, we consider only refinement of an action with a sequence of actions.

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

▶ **Corollary 5.** *For process terms t_1 and t_2, and substitution σ mapping actions, say a, to a sequence of actions, say $b_1 \cdots b_n$, if $\vdash [\![t_1]\!] \multimap [\![t_2]\!]$ then $\vdash [\![t_1\sigma]\!] \multimap [\![t_2\sigma]\!]$.*

For example, since $\vdash [\![a \parallel a]\!] \multimap [\![a.a]\!]$ holds, by applying the action refinement $\sigma = \left\{{}^{b.c}/_{a}\right\}$, the following holds: $\vdash [\![b.c \parallel b.c]\!] \multimap [\![b.c.b.c]\!]$.

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

4 Examples of properties established using linear implication

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

4.1 Refinements also provable using probabilistic simulation

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

$$\mathit{heads} +_{1/2} \mathit{tails} \quad \text{is unrelated to} \quad \mathit{heads} \quad \text{and also is unrelated to} \quad \mathit{tails}$$

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

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

$$P \mathbin{\&} Q \multimap P \mathbin{\&}_p Q \qquad\qquad P \mathbin{\&}_p Q \multimap P \oplus_p Q \qquad\qquad P \oplus_p Q \multimap P \oplus Q$$

The first implication has a proof of the following form.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\circ}{\circ \mathbin{\&}_p \circ}\ \mathit{idempotency}}{\left(\overline{P} \parr P\right) \mathbin{\&}_p \left(\overline{Q} \parr Q\right)}\ \text{Proposition 3}}{\left(\left(\overline{P} \oplus \overline{Q}\right) \parr P\right) \mathbin{\&}_p \left(\left(\overline{P} \oplus \overline{Q}\right) \parr Q\right)}\ \mathit{choose}}{\left(\left(\overline{P} \oplus \overline{Q}\right) \oplus_p \left(\overline{P} \oplus \overline{Q}\right)\right) \parr \left(P \mathbin{\&}_p Q\right)}\ \mathit{confine}}{\left(\overline{P} \oplus \overline{Q}\right) \parr \left(P \mathbin{\&}_p Q\right)}\ \mathit{idempotency}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\circ}{\circ \mathbin{\&}_p \circ}\ \mathit{idempotency}}{\left(\overline{P} \parr P\right) \mathbin{\&}_p \left(\overline{Q} \parr Q\right)}\ \text{Proposition 3}}{\left(\overline{P} \mathbin{\&}_p \overline{Q}\right) \parr \left(P \oplus_p Q\right)}\ \mathit{confine}}{\left(\overline{P} \oplus_p \overline{Q}\right) \parr \left(\circ \mathbin{\&}_p \circ\right) \parr \left(P \oplus_p Q\right)}\ \mathit{confine}}{\left(\overline{P} \oplus_p \overline{Q}\right) \parr \left(P \oplus_p Q\right)}\ \mathit{idempotency}$$

Notice, by instantiating the above with process embeddings, $\vdash [\![t_1 \sqcap t_2]\!] \multimap [\![t_1]\!] \mathbin{\&_p} [\![t_2]\!]$ and $\vdash [\![t_1]\!] \mathbin{\&_p} [\![t_2]\!] \multimap [\![t_1 +_p t_2]\!]$ hold. Hence, by Theorem 2, there is also a proof of the following.

$$\vdash [\![t_1 \sqcap t_2]\!] \multimap [\![t_1 +_p t_2]\!]$$

As guaranteed by Theorem 4, the above linear implication can also be established by probabilistic simulation. For example, process $a \sqcap b$ simulates $a +_p b$. This holds since $\mathcal{R}$ such that $a \mathrel{\mathcal{R}} \mathbf{1}_{a \sqcap b}$, $b \mathrel{\mathcal{R}} \mathbf{1}_{a \sqcap b}$, and $\mathtt{ok} \mathrel{\mathcal{R}} \mathbf{1}_{\mathtt{ok}}$ defines a weak probabilistic simulation such that $[\![a \mathbin{\&_p} b]\!] \mathrel{\hat{\mathcal{R}}} [\![a \sqcap b]\!]$. The converse does not hold since $a \sqcap b \xRightarrow{a} \mathbf{1}_{\mathtt{ok}}$, which is a transition that cannot be matched by distribution $p\mathbf{1}_a + (1-p)\mathbf{1}_b$. Hence, by Theorem 4, the converse implication $P \oplus_p Q \multimap P \mathbin{\&} Q$ also does **not** hold in general.

4.2 Distributivity properties, some forbidden others permitted

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

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

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

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

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

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

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

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

There are also distributivity properties relating non-deterministic and probabilistic choice [43]. For example we have that $\vdash (P \mathbin{\&} Q) \oplus_p (P \mathbin{\&} R) \multimap P \mathbin{\&} (Q \oplus_p R)$ holds, as established by the following proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\circ}{\circ \mathbin{\&} \circ}\ \textit{tidy}}{(\circ \mathbin{\&_p} \circ) \mathbin{\&} (\circ \mathbin{\&_p} \circ)}\ \textit{idempotency}}{\left(\left(\overline{P} \parr P\right) \mathbin{\&_p} \left(\overline{P} \parr P\right)\right) \mathbin{\&} \left(\left(\overline{Q} \parr Q\right) \mathbin{\&_p} \left(\overline{R} \parr R\right)\right)}\ \text{by Proposition 3}}{\left(\left(\overline{P} \mathbin{\&_p} \overline{P}\right) \parr (P \oplus_p P)\right) \mathbin{\&} \left(\left(\overline{Q} \mathbin{\&_p} \overline{R}\right) \parr (Q \oplus_p R)\right)}\ \text{by } \textit{confine}}{\left(\left(\overline{P} \mathbin{\&_p} \overline{P}\right) \parr P\right) \mathbin{\&} \left(\left(\overline{Q} \mathbin{\&_p} \overline{R}\right) \parr (Q \oplus_p R)\right)}\ \textit{idempotency}}{\left(\left(\left(\overline{P} \oplus \overline{Q}\right) \mathbin{\&_p} \left(\overline{P} \oplus \overline{R}\right)\right) \parr P\right) \mathbin{\&} \left(\left(\left(\overline{P} \oplus \overline{Q}\right) \mathbin{\&_p} \left(\overline{P} \oplus \overline{R}\right)\right) \parr (Q \oplus_p R)\right)}\ \text{by } \textit{choose}}{\left(\left(\overline{P} \oplus \overline{Q}\right) \mathbin{\&_p} \left(\overline{P} \oplus \overline{R}\right)\right) \parr (P \mathbin{\&} (Q \oplus_p R))}\ \text{by } \textit{external}$$

By Theorem 4, we have that $(t_1 \sqcap t_2) +_p (t_1 \sqcap t_3)$ simulates $t_1 \sqcap (t_2 +_p t_3)$, for any process. For example, $a \sqcap (b +_p c)$ is simulated by $(a \sqcap b) +_p (a \sqcap c)$. To see why, observe relation $\mathcal{S}$ defined such that $a \sqcap (b +_p c)\ \mathcal{S}\ p\mathbf{1}_{a \sqcap b} + (1-p)\mathbf{1}_{a \sqcap c}$ and $s\ \mathcal{S}\ \mathbf{1}_s$, for any s, is a simulation; for which $[\![a \sqcap (b +_p c)]\!]\ \hat{\mathcal{S}}\ [\![(a \sqcap b) +_p (a \sqcap c)]\!]$.

The converse of the above simulation does not hold. Hence, as a consequence of Theorem 4, the converse of the above implication does not hold in ΔMAV. I.e., in general, the following is **not** provable: $P \mathbin{\&} (Q \oplus_p R) \multimap (P \mathbin{\&} Q) \oplus_p (P \mathbin{\&} R)$.

4.3 But are the medial rules necessary in ΔMAV?

The most mysterious rules of ΔMAV are the *medial* rules. The justification we provide here is purely logical, although these rules are likely to play a more significant role when considering more expressive process calculi with full sequential composition and mixing suitable notions of internal and external choice (sometimes known as angelic/daemonic choices [31]).

Here we show the *medial* rules are necessary in order for cut-elimination to hold. *Medial* rules capture a pattern where a weaker additive distributes over a stronger additive, where $\& < \&_p < \oplus_p < \oplus$. This is a derived property of the standard additives in linear logic; namely the implication $(P \mathbin{\&} Q) \oplus (R \mathbin{\&} S) \multimap (P \oplus R) \mathbin{\&} (Q \oplus S)$ is provable, while its converse does not hold. The corresponding property for the sub-additive is not derivable without the *medials*. Only by including an explicit *medial* rule in Fig. 2 can we prove the following property.

$$(P \mathbin{\&}_p Q) \oplus_q (R \mathbin{\&}_p S) \multimap (P \oplus_q R) \mathbin{\&}_p (Q \oplus_q S)$$

We are forced to include several further *medial* rules, induced by associativity and commutativity. This is more surprising since all other *medial* rules correspond to implications provable without including any *medial* rules. For example, we have the following proof of implication $(P \mathbin{\&} Q) \mathbin{\&}_q (R \mathbin{\&} S) \multimap (P \mathbin{\&}_q R) \mathbin{\&} (Q \mathbin{\&}_q S)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\circ}{(\circ \mathbin{\&}_q \circ) \mathbin{\&} (\circ \mathbin{\&}_q \circ)}\ \textit{tidy and itempotency}}{\left(\left(\overline{P} \parr P\right) \mathbin{\&}_q \left(\overline{R} \parr R\right)\right) \mathbin{\&} \left(\left(\overline{Q} \parr Q\right) \mathbin{\&}_q \left(\overline{S} \parr S\right)\right)}\ \textit{interact}}{\left(\left(\left(\overline{P} \oplus \overline{Q}\right) \parr P\right) \mathbin{\&}_q \left(\left(\overline{R} \oplus \overline{S}\right) \parr R\right)\right) \mathbin{\&} \left(\left(\left(\overline{P} \oplus \overline{Q}\right) \parr Q\right) \mathbin{\&}_q \left(\left(\overline{R} \oplus \overline{S}\right) \parr S\right)\right)}\ \textit{choose}}{\left(\left(\left(\overline{P} \oplus \overline{Q}\right) \oplus_q \left(\overline{R} \oplus \overline{S}\right)\right) \parr (P \mathbin{\&}_q R)\right) \mathbin{\&} \left(\left(\left(\overline{P} \oplus \overline{Q}\right) \oplus_q \left(\overline{R} \oplus \overline{S}\right)\right) \parr (Q \mathbin{\&}_q S)\right)}\ \textit{confine}}{\left(\left(\overline{P} \oplus \overline{Q}\right) \oplus_q \left(\overline{R} \oplus \overline{S}\right)\right) \parr \left((P \mathbin{\&}_q R) \mathbin{\&} (Q \mathbin{\&}_q S)\right)}\ \textit{external}$$

The above implication does not mean rule $\dfrac{(P \mathbin{\&} Q) \mathbin{\&}_q (R \mathbin{\&} S)}{(P \mathbin{\&}_q R) \mathbin{\&} (Q \mathbin{\&}_q S)}$ is admissible (redundant in ΔMAV). To see why, consider the following observations. Firstly, observe the following is provable without using any medial rules.

$$(a_1 \parr a_2) \mathbin{\&}_p ((b_1 \mathbin{\&}_q (c \mathbin{\&} d)) \parr (b_2 \oplus_q (c \mathbin{\&} d))) \multimap (a_1 \mathbin{\&}_p (b_1 \mathbin{\&}_q (c \mathbin{\&} d))) \parr (a_2 \oplus_p (b_2 \oplus_q (c \mathbin{\&} d)))$$

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

$$(a_1 \mathbin{\&}_p (b_1 \mathbin{\&}_q (c \mathbin{\&} d))) \parr (a_2 \oplus_p (b_2 \oplus_q (c \mathbin{\&} d))) \equiv (b_1 \mathbin{\&}_r (a_1 \mathbin{\&}_s (c \mathbin{\&} d))) \parr (b_2 \oplus_r (a_2 \oplus_s (c \mathbin{\&} d)))$$

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

$$\begin{aligned}&(b_1 \mathbin{\&}_r (a_1 \mathbin{\&}_s (c \mathbin{\&} d))) \parr (b_2 \oplus_r (a_2 \oplus_s (c \mathbin{\&} d)))\\ &\qquad \multimap (b_1 \mathbin{\&}_r ((a_1 \mathbin{\&}_s c) \mathbin{\&} (a_1 \mathbin{\&}_s d))) \parr (b_2 \oplus_r ((a_2 \oplus_s c) \mathbin{\&} (a_2 \oplus_s d)))\end{aligned}$$

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

$$(a_1 \parr a_2) \mathbin{\&_p} ((b_1 \mathbin{\&_q} (c \mathbin{\&} d)) \parr (b_2 \oplus_q (c \mathbin{\&} d))) \\ \multimap (b_1 \mathbin{\&_r} ((a_1 \mathbin{\&_s} c) \mathbin{\&} (a_1 \mathbin{\&_s} d))) \parr (b_2 \oplus_r ((a_2 \oplus_s c) \mathbin{\&} (a_2 \oplus_s d)))$$

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

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

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

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

$$\frac{\mathcal{C}\{\ R \oplus_p R\ \}}{\mathcal{C}\{\ R\ \}}\ \textit{contract} \qquad \frac{\mathcal{C}\{\ \circ\ \}}{\mathcal{C}\{\ \circ \mathbin{\&_p} \circ\ \}}\ \textit{tidy distribution} \qquad \frac{\mathcal{C}\{\ P \mathbin{\&_r} Q\ \}}{\mathcal{C}\{\ P \oplus_r Q\ \}}\ \textit{special case of confine}$$

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

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

$$\frac{\mathcal{C}\{\ \circ \oplus \circ\ \}}{\mathcal{C}\{\ \circ\ \}}\ \text{co-tidy} \qquad \frac{\mathcal{C}\{\ a \otimes \overline{a}\ \}}{\mathcal{C}\{\ \circ\ \}}\ \text{co-interact} \qquad \frac{\mathcal{C}\{\ P\ \}}{\mathcal{C}\{\ P \mathbin{\&_p} P\ \}}\ \text{co-contract}$$

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

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

▶ **Lemma 6** (decomposition). *For any derivation* $\dfrac{S}{P}$*, including co-rules, there exists* Q *and* R *such that there is a derivation:*

$$\begin{array}{l} S \\ \hline R \\ \hline Q \\ \hline P \end{array} \begin{array}{l} \\ \textit{using co-contract only} \\ \textit{including co-rules but without contract or co-contract} \\ \textit{using contract only} \end{array}$$

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

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

- *If* $\vdash (P \mathbin{\&} Q) \mathbin{⅋} R$ *then* $\vdash P \mathbin{⅋} R$ *and* $\vdash Q \mathbin{⅋} R$.
- *If* $\vdash (P \mathbin{\&_p} Q) \mathbin{⅋} R$, *there exist* U, V *such that* $\dfrac{U \oplus_p V}{R}$ *and both* $\vdash P \mathbin{⅋} U$ *and* $\vdash Q \mathbin{⅋} V$ *hold.*
- *If* $\vdash (P \oplus_p Q) \mathbin{⅋} R$, *there exist* U, V *such that* $\dfrac{U \mathbin{\&_p} V}{R}$ *and both* $\vdash P \mathbin{⅋} U$ *and* $\vdash Q \mathbin{⅋} V$ *hold.*
- *If* $\vdash (P \lhd Q) \mathbin{⅋} R$, *there exist* $\mathcal{T}\{\ \}$, U_i *and* V_i *such that* $\dfrac{\mathcal{T}\{\ U_i \lhd V\ \}}{R}$ *and, for all* i, *both* $\vdash P \mathbin{⅋} U_i$ *and* $\vdash Q \mathbin{⅋} V_i$ *hold.*
- *If* $\vdash (P \otimes Q) \mathbin{⅋} R$, *there exist* $\mathcal{T}\{\ \}$, U_i *and* V_i *such that* $\dfrac{\mathcal{T}\{\ U_i \mathbin{⅋} V_i\ \}}{R}$ *and, for all* i, $\vdash P \mathbin{⅋} U_i$ *and* $\vdash Q \mathbin{⅋} V_i$.
- *If* $\vdash (P \oplus Q) \mathbin{⅋} R$ *then, there exist* W_i *such that* $\dfrac{\mathcal{T}\{\ W_i\ \}}{R}$ *and, for all* i, *either* $\vdash P \mathbin{⅋} W_i$ *or* $\vdash Q \mathbin{⅋} W_i$ *hold.*
- *If* $\vdash a \mathbin{⅋} R$ *then* $\dfrac{\mathcal{T}\{\ \overline{a}\ \}}{R}$.
- *If* $\vdash \overline{a} \mathbin{⅋} R$ *then* $\dfrac{\mathcal{T}\{\ a\ \}}{R}$.

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

▶ **Lemma 8** (context reduction). *If, for all* R, $\vdash P \mathbin{⅋} R$ *yields* $\vdash Q \mathbin{⅋} R$ *then, for all contexts* $\mathcal{C}\{\ \}$, $\vdash \mathcal{C}\{\ P\ \}$ *yields* $\vdash \mathcal{C}\{\ Q\ \}$.

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

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

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

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

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

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

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

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

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

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

6 Related work on Sub-Additive Operators and Nominal Quantifiers

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

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

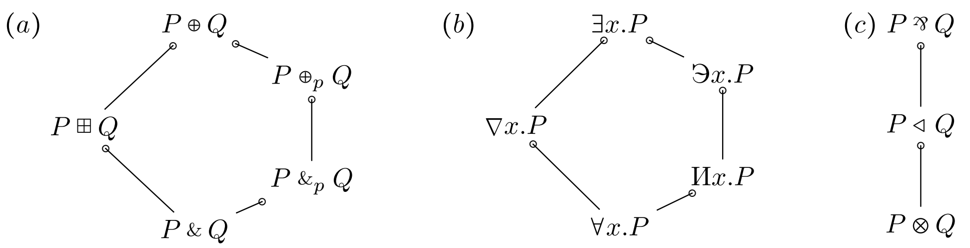

Figure 3 Relationships between various operators in extensions of linear logic: (a) the additives and sub-additives, (b) the first-order quantifiers and nominal quantifiers, (c) the multiplicatives.

Figure 3(a) compares additives $\&$, $\&_p$, $\oplus_p$, $\oplus$ and $\boxplus$. Notice similarities with Fig 3(b) depicting de Morgan dual pair of nominal quantifiers, И$x.P$ and Ǝ$x.P$, located between *for all* and *exists* [23]. Similarly, to the sub-additives, the justification for the pair of nominal quantifiers, rather than a self-dual nominal quantifier [14, 34, 35], say $\nabla x.P$, was to soundly model private names in direct logical embeddings of π-calculus processes [32].

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

7 Conclusion

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

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

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

References

1 Samy Abbes and Albert Benveniste. True-concurrency probabilistic models: Branching cells and distributed probabilities for event structures. *Information and Computation*, 204(2):231–274, 2006. doi:10.1016/j.ic.2005.10.001.

2 Christel Baier and Holger Hermanns. Weak bisimulation for fully probabilistic processes. In *Computer Aided Verification*, pages 119–130. Springer, 1997. doi:10.1007/3-540-63166-6_14.

3 Paola Bruscoli. A Purely Logical Account of Sequentiality in Proof Search. In *ICLP*, pages 302–316, 2002. doi:10.1007/3-540-45619-8_21.

4 Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. The focused calculus of structures. In *CSL*, volume 12, pages 159–173, 2011. doi:10.4230/LIPIcs.CSL.2011.159.

5 Gabriel Ciobanu and Ross Horne. Behavioural Analysis of Sessions using the Calculus of Structures. In *PSI 2015, 25-27 August, Kazan, Russia*, volume 9609 of *LNCS*, pages 91–106, 2015. doi:10.1007/978-3-319-41579-6_8.

6 Ugo Dal Lago, Claudia Faggian, Benoît Valiron, and Akira Yoshimizu. The Geometry of Parallelism: Classical, Probabilistic, and Quantum Effects. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL 2017, pages 833–845. ACM, 2017. doi:10.1145/3009837.3009859.

7 Vincent Danos and Russell S. Harmer. Probabilistic Game Semantics. *ACM Trans. Comp. Logic*, 3(3):359–382, July 2002. doi:10.1145/507382.507385.

8 Jerry den Hartog, Erik P. de Vink, and Jacobus W. De Bakker. Metric semantics and full abstractness for action refinement and probabilistic choice. *Electronic Notes in Theoretical Computer Science*, 40:72–99, 2001. doi:10.1016/S1571-0661(05)80038-6.

9 Yuxin Deng. *Semantics of Probabilistic Processes: An Operational Approach*. Springer, 2015. doi:10.1007/978-3-662-45198-4.

10 Yuxin Deng, Matthew Hennessy, Rob van Glabbeek, and Carroll Morgan. Characterising Testing Preorders for Finite Probabilistic Processes. In *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 313–325, July 2007. doi:10.1109/LICS.2007.15.

11 Yuxin Deng, Rob van Glabbeek, Matthew Hennessy, and Carroll Morgan. Characterising Testing Preorders for Finite Probabilistic Processes. *Logical Methods in Computer Science*, 4(4), 2008. doi:10.2168/LMCS-4(4:4)2008.

12 Thomas Ehrhard, Michele Pagani, and Christine Tasson. The computational meaning of probabilistic coherence spaces. In *LICS*, pages 87–96. IEEE, 2011. doi:10.1109/LICS.2011.29.

13 Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. *In Proc. POPL*, 49(1):309–320, 2014. doi:10.1145/2535838.2535865.

14 Andrew Gacek, Dale Miller, and Gopalan Nadathur. Nominal abstraction. *Information and Computation*, 209(1):48–73, 2011. doi:10.1016/j.ic.2010.09.004.

15 Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–112, 1987. doi:10.1016/0304-3975(87)90045-4.

16 Jean-Yves Girard. Between logic and quantic: a tract. *Linear logic in computer science*, 316:346–381, 2004. doi:10.1017/CBO9780511550850.011.

17 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1), 2007. doi:10.1145/1182613.1182614.

18 Alessio Guglielmi and Tom Gundersen. Normalisation Control in Deep Inference via Atomic Flows. *Logical Methods in Computer Science*, 4(1), 2008. doi:10.2168/LMCS-4(1:9)2008.

19 Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure V: the exponentials and splitting. *Mathematical Structures in Computer Science*, 21(3):563–584, 2011. doi:10.1017/S096012951100003X.

20 Ross Horne. The Consistency and Complexity of Multiplicative Additive System Virtual. *Sci. Ann. Comp. Sci.*, 25(2):245–316, 2015. doi:10.7561/SACS.2015.2.245.

21 Ross Horne, Sjouke Mauw, and Alwen Tiu. Semantics for Specialising Attack Trees based on Linear Logic. *Fundamenta Informaticae*, 153(1-2):57–86, 2017. doi:10.3233/FI-2017-1531.

22 Ross Horne and Alwen Tiu. Constructing Weak Simulations from Linear Implications for Processes with Private Names. *Mathematical Structure in Computer Science*, pages 1–34, 2019. doi:10.1017/S0960129518000452.

23 Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. Private Names in Non-Commutative Logic. In *CONCUR 2016*, pages 31:1–31:16. LIPIcs, 2016. doi:10.4230/LIPIcs.CONCUR.2016.31.

24 Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. De Morgan Dual Nominal Quantifiers Modelling Private Names in Non-Commutative Logic. *ACM Transactions on Computational Logic (TOCL)*, 20(4), 2019.

25 Michael Huth and Marta Z. Kwiatkowska. Quantitative analysis and model checking. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 111–122, 1997. doi:10.1109/LICS.1997.614940.

26 Cliff Jones and Gordon Plotkin. A probabilistic powerdomain of evaluations. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*, pages 186–195, June 1989. doi:10.1109/LICS.1989.39173.

27 Bengt Jonsson and Wang Yi. Compositional testing preorders for probabilistic processes. In *LICS*, pages 431–441. IEEE, 1995. doi:10.1109/LICS.1995.523277.

28 Bengt Jonsson, Wang Yi, and Kim G Larsen. Probabilistic extensions of process algebras. *Handbook of process algebra*, pages 685–710, 2001. doi:10.1016/B978-044482830-9/50029-1.

29 Ugo Dal Lago and Margherita Zorzi. Probabilistic operational semantics for the lambda calculus. *RAIRO - Theoretical Informatics and Applications*, 46(3):413–450, 2012. doi:10.1051/ita/2012012.

30 Kim G Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and computation*, 94(1):1–28, 1991. doi:10.1016/0890-5401(91)90030-6.

31 Annabelle McIver and Carroll Morgan. *Abstraction, Refinement And Proof For Probabilistic Systems (Monographs in Computer Science)*. SpringerVerlag, 2004. doi:10.1007/b138392.

32 Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–77, 1992. doi:10.1016/0890-5401(92)90008-4.

33 Catuscia Palamidessi and Oltea Mihaela Herescu. A randomized encoding of the π-calculus with mixed choice. *Theoretical Computer Science*, 335(2-3):373–404, 2005. doi:10.1016/j.tcs.2004.11.020.

34 Andrew Pitts. Nominal Logic, a First Order Theory of Names and Binding. *Information and Computation*, 186(2), 2003. doi:10.1016/S0890-5401(03)00138-X.

35 Luca Roversi. A deep inference system with a self-dual binder which is complete for linear lambda calculus. *J. of Logic and Computaion*, 26(2):677–698, 2016. doi:10.1093/logcom/exu033.

36 Davide Sangiorgi. pi-Calculus, Internal Mobility, and Agent-Passing Calculi. *Theor. Comput. Sci.*, 167(1&2):235–274, 1996. doi:10.1016/0304-3975(96)00075-8.

37 Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995. doi:10.1007/3-540-56454-3_13.

38 Lutz Straßburger. A local system for linear logic. In *LPAR*, volume 2514, pages 388–402. Springer, 2002. doi:10.1007/3-540-36078-6_26.

39 Lutz Straßburger and Alessio Guglielmi. A system of interaction and structure IV: The exponentials and decomposition. *ACM T. Comp. Log.*, 12(4):23:1–39, 2011. doi:10.1145/1970398.1970399.

40 Andrea Aler Tubella, Alessio Guglielmi, and Benjamin Ralph. Removing Cycles from Proofs. In *CSL*, pages 9:1–9:17, 2017. doi:10.4230/LIPIcs.CSL.2017.9.

41 Rob van Glabbeek and Ursula Goltz. Refinement of actions and equivalence notions for concurrent systems. *Acta Informatica*, 37(4-5):229–327, 2001. doi:10.1007/s002360000041.

42 Daniele Varacca, Hagen Völzer, and Glynn Winskel. Probabilistic event structures and domains. In *CONCUR*, pages 481–496, 2004. doi:10.1007/978-3-540-28644-8_31.

43 Daniele Varacca and Glynn Winskel. Distributing probability over non-determinism. *Mathematical Structures in Computer Science*, 16(1):87–113, 2006. doi:10.1017/S0960129505005074.