Duality for Labelled Markov Processes

Michael Mislove[1], Joël Ouaknine[2], Dusko Pavlovic[3] and James Worrell[1]

[1] Tulane University, Department of Mathematics, New Orleans, USA
[2] Computer Science Department, Carnegie Mellon University, USA
[3] Kestrel Institute, Palo Alto, USA

Abstract. Labelled Markov processes (**LMP**s) are probabilistic labelled transition systems. In this paper we present a 'universal' **LMP** as the Stone-Gelfand-Naimark dual of a C^*-algebra consisting of formal linear combinations of labelled trees. We characterize the state space of the universal **LMP** as the set of homomorphims from an ordered commutative monoid of labelled trees into the multiplicative unit interval. This yields a simple semantics for **LMP**s which is fully abstract with respect to probabilistic bisimilarity. We also consider **LMP**s with entry points and exit points in the framework of Elgot's iterative theories. We define an iterative theory of **LMP**s by specifying its categorical dual: a category of commutative rings consisting of C^*-algebras of trees and 'shapely maps' between them. We find that the basic operations for composing or programming **LMP**s have simple definitions in the dual category.

1 Introduction

This paper is concerned with the semantics of certain probabilistic labelled transition systems, called labelled Markov processes (or **LMP**s) [10, 12, 8, 9]. Probabilistic models are important for capturing quantitative aspects of process behaviour, such as performance and reliability, e.g., the probability with which a failure occurs, or the average response time to a given action. For this reason there has been a lot of research into adapting concepts and results of classical concurrency theory to the probabilistic case. In particular, the notion of bisimilarity has been adapted to probabilistic systems [17, 10, 16], and its equational theory investigated in [22, 5, 19].

The bisimulation equivalence classes of **LMP**s can be gathered together into what could be termed a universal **LMP**. This object has previously been studied as the solution of a domain equation in the category of complete metric spaces [8], and in the category of coherent domains [12, 9]. However, none of these domain theoretic treatments yielded concrete representations of the elements of the universal **LMP**.

[1] Supported by Office of Naval Research, grant No. N000149910150 and National Science Foundation, grant No. CCR-0208743.
[2] Supported by ONR contract N00014-95-1-0520.
[3] Partially supported by the EHS and SGER programs of the National Science Foundation, under contract No. CCR-0209-004 and CCR-0345-397.

In this paper we exploit Stone-Gelfand-Naimark duality for C^*-algebras to show that the universal **LMP** has a very straightforward characterization as a space of order-preserving monoid homomorphisms from a partially ordered monoid $\mathbb{T}$ of trees to the multiplicative monoid $[0,1]$. The salient feature of this characterization is the lack of explicit reference to probabilities.

We think of the elements of $\mathbb{T}$ as *branching traces* or *trace trees*. Formally they are just finite trees whose edges are labelled by events from a given alphabet. The operation of grafting two such trees at the root gives the monoid multiplication in $\mathbb{T}$. The order on $\mathbb{T}$ is the natural generalization of the prefix order on traces. For a given **LMP** the corresponding homomorphism maps each trace tree to the probability that it gets performed.

In an earlier paper [9] we showed that two processes are bisimilar iff they perform each trace tree with the same probability. This generalized a result of Larsen and Skou [17].[1] The main result of this paper can be seen as extending this characterization result to 'build processes out of trace trees'. This is a natural variation of the familiar trace models of abstract machines.

The main mathematical tool that we use is the theorem of Stone-Gelfand-Naimark which asserts a dual equivalence between the category of compact Hausdorff spaces and continuous maps on the one hand, and the category of real C^*-algebras (a full subcategory of the category of commutative rings) on the other hand. This duality associates to each compact Hausdorff space the ring of continuous real-valued functions on the space, and to each C^*-algebra its spectral space of *characters*: ring homomorphisms into $\mathbb{R}$. We apply the Stone-Gelfand-Naimark duality to recover the universal **LMP** as the spectrum of a C^*-algebra consisting of formal linear combinations of trace trees.

The concrete representation of the universal **LMP**, obtained in the first part of this paper, opens a new effective approach to composing and programming **LMP**s. In the present paper, we outline the use of basic combinators, such as substitution, probabilistic choice and recursion, in the setting of Elgot's iterative theories [7]. They arise when the basic model of **LMP**s is extended by entry points an exit points. Thus we obtain a category whose objects are *arities*, i.e. nonnegative integers. A morphism $\mathcal{S}: n \to p$ is an **LMP** with n entry points and p exit points. This category turns out to be dual (contravariantly equivalent) to a category of commutative rings consisting of C^*-algebras of trace trees and 'shape preserving' ring homomorphisms. Represented through this duality, the basic combinators for **LMP**s have remarkably simple descriptions.

1.1 Related Work

Kozen [14] presents a predicate transformer semantics of an imperative programming language with probabilistic choice. This semantics is based on a duality between linear maps and probabilistic relations. The probabilistic relations model programs as state transformers. In the dual view, programs act as linear transformations, mapping measurable functions on final states to measurable functions

[1] In fact we took the view that trace trees are types of button-pressing tests.

on initial states. The semantics is formalized in the setting of iteration theories. The same themes of duality and iteration theories appear in the present paper, but our development is in the context of interactive processes rather than imperative programs. In particular, for us states are not just measurable functions on a space of variables, but have a recursively defined structure.

An iteration theory of probabilistic processes has been studied by Aceto, Ésik and Ingólfsdóttir [5], building on earlier work of Stark and Smolka [22]. These two papers treat finite-state **LMP**s as terms in a simple probabilistic process calculus. Their main contributions are soundness and completeness results for various axiomatizations of the bisimilarity relation. Since these papers deal with process calculi, the basic operations of prefixing, probabilistic choice and iteration are defined at the syntactic level using an operational semantics. These operations are then lifted to bisimulation equivalence classes of terms using the fact that bisimilarity is a congruence. In contrast, we utilize a concrete representation of bisimulation equivalence classes of **LMP**s as maps of C^*-algebras, and define the basic operations directly on these representations.

One of the most comprehensive applications of duality in semantics can be found in the work of Abramsky on domain theory in logical form [2]. This work is based on a Stone-type duality between a category of spectral spaces (SFP domains in their Scott topologies) and a category of distributive lattices. In particular, as a case study, Abramsky considers a domain equation for bisimulation and computes its Stone dual. What we have in done in this paper is compute the Stone-Gelfand-Naimark dual of a domain equation for probabilistic bisimulation. However, so far our work is much more modest in scope; in particular we have not tried to isolate a fragment of Stone-Gelfand-Naimark duality that is pertinent to any reasonable category of domains.

Another paper close in spirit to the present work is Abramsky and Vickers [3]. They consider a variety of equivalences for concurrent processes in a unified framework of quantale modules—actions of quantales on sup-lattices. In particular, they present quantales of tests using generators and relations, and model transition systems as right quantale modules (where the elements of the quantale act on states of the transition system). Using the self-duality of the category of sup-lattices they obtain left quantale modules of 'process capabilities' which they use to build fully abstract models of processes, where the meaning of a process is its capabilities.

Di Pierro, Hankin and Wicklicky [21] use C^*-algebras to define abstract interpretations of probabilistic transition systems. However, at this stage, their work does not seem closely related to ours.

2 Labelled Markov Processes

Below we give the formal definition of the class of probabilistic transition systems that we study in this paper. This definition extends that of Larsen and Skou [17] by including entry points and exit points as part of the basic data. This extension allows us to define composition (or substitution) and iteration.

Let $[n]$ denote the set $\{1, \ldots, n\}$ for a positive integer n, and let $[0]$ denote $\emptyset$. Furthermore we assume a fixed finite set Act of actions or events.

Definition 1. *Given non-negative integers n, p, a* labelled Markov process $\mathcal{S} : n \to p$ *is a tuple $\langle S, \mathrm{in}, \mu \rangle$ consisting of a set S of states, a function* $\mathrm{in} : [n] \to S$*, and, for each $s \in S$, a subprobability distribution μ_s on* $(\mathrm{Act} \times S) \cup [p]$.

Intuitively $\mu_s(a, t)$ is the probability that the process in state s makes an a-transition to state t, and $\mu_s(i)$ is the probability that it makes an transition to the exit point $i \in [p]$. Note that μ_s is a *subprobability* distribution, i.e., its total mass is no greater than 1. We interpret the difference between the total mass of μ_s and 1 as the probability of deadlock in state s. We also adopt the notation $\mu_{s,a}$ for the subprobability distribution on S given by $\mu_{s,a}(t) = \mu_s(a, t)$.

A theme of the work on **LMP**s [10] has been to allow more generality than Definition 1 by taking the state space S to be a measurable space, and the transitions to be given by subprobability *measures*. All our results hold in the more general setting—indeed the more general view is required in order to formulate and prove our main theorems. However we focus on the discrete case as much as possible to keep things simple.

Probabilistic bisimilarity [17] (henceforth just bisimilarity) is the probabilistic analog of Park and Milner's notion of strong bisimilarity [18]. It gives a branching-time notion of behavioural equivalence for **LMP**s.

Definition 2. *Let $\mathcal{S} : n \to p$ be an* **LMP** *with $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$. An equivalence relation R on S is a* bisimulation *if sRt implies that*

- *for each $a \in \mathrm{Act}$ and R-equivalence class A, $\mu_{s,a}(A) = \mu_{t,a}(A)$,*
- *for each $i \in [p]$, $\mu_s(i) = \mu_t(i)$.*

We say that two states are bisimilar if they are related by some bisimulation.

In words: an equivalence relation is a bisimulation if related states have matching probabilities of making transitions into any equivalence class and into any exit point.

3 Operations on LMPs

In this section we define some operations for composing **LMP**s. These are the counterparts on the semantic level of constructs that might be found in a typical process calculus. In particular, processes with exit points correspond to terms with free variables, composition corresponds to substitution of terms, and iteration to application of the recursion operator. These definitions will later form the basis of a category in which **LMP**s are morphisms. For a similar treatment of labelled transition systems see [7].

Coproducts. Given a strictly positive integer n and $i \in [n]$, the injection $\mathcal{S}_n^i : 1 \to n$ is an **LMP** with one state which makes a transition to the i-th output with probability 1. Formally we let $\mathcal{S}_n^i = \langle \{s\}, \mathrm{in}, \mu \rangle$, where $\mu_s(i) = 1$.

Tupling. Let $\mathcal{S}: n \to p$ and $\mathcal{S}': m \to p$ be **LMP**s with $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$ and $\mathcal{S}' = \langle S', \mathrm{in}', \mu' \rangle$. Then the tuple $\langle \mathcal{S}, \mathcal{S}' \rangle : n + m \to p$ is defined to be $\langle S \cup S', \mathrm{in}'', \mu'' \rangle$ where

$$\mathrm{in}''(i) = \begin{cases} \mathrm{in}(i) & \text{if } 1 \leqslant i \leqslant n \\ \mathrm{in}'(i-n) & \text{if } n+1 \leqslant i \leqslant n+m \end{cases}$$

$\mu''_s = \mu_s$ for $s \in S$, and $\mu''_s = \mu'_s$ for $s \in S'$.

Composition. Let $\mathcal{S}: n \to m$ and $\mathcal{S}': m \to p$ be **LMP**s with $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$ and $\mathcal{S}' = \langle S', \mathrm{in}', \mu' \rangle$. The composition $\mathcal{S} \fatsemi \mathcal{S}' : n \to p$ is obtained by connecting the outputs of $\mathcal{S}$ with the inputs of $\mathcal{S}'$. Formally it is defined to be $\langle S \cup S', \mathrm{in}'', \mu'' \rangle$, where $\mathrm{in}''(i) = \mathrm{in}(i)$, for $i \in [n]$,

$$\mu''_s(a,t) = \begin{cases} \mu_s(a,t) & \text{if } s,t \in S \\ \sum_{i=1}^{m} \mu_s(i) \cdot \mu'_{in'(i)}(a,t) & \text{if } s \in S, t \in S' \\ \mu'_s(a,t) & \text{if } s,t \in S' \\ 0 & \text{if } s \in S', t \in S \end{cases}$$

and $\mu''_s(i) = \mu'_s(i)$ for $s \in S'$ and $i \in [p]$.

Iteration. Given an **LMP** $\mathcal{S}: n \to n + p$ with $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$, the iterate $\dagger\mathcal{S}: n \to p$ is obtained by identifying the i-th exit point of $\mathcal{S}$ with the i-th entry point for each $i \in [n]$. Formally $\dagger\mathcal{S} = \langle S, \mathrm{in}, \mu' \rangle$ where μ' is defined by:

$$\mu'_s(a,t) = \mu_s(a,t) + \sum_{i=1}^{n} \mu_s(i) \cdot \mu'_{\mathrm{in}(i)}(a,t)$$

and

$$\mu'_s(i) = \mu_s(i+n) + \sum_{j=1}^{n} \mu_s(j) \cdot \mu'_{\mathrm{in}(j)}(i)\,.$$

Notice that the definition of μ' is recursive. It is straightforward that μ'_s is uniquely defined by the clauses above provided that $\sum_{j=1}^{n} \mu_{\mathrm{in}(i)}(j) < 1$ for all $i \in [n]$.

Probabilistic Sum. Let $\mathcal{S}: n \to p$ and $\mathcal{S}': n \to p$ be **LMP**s. Write $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$ and $\mathcal{S}' = \langle S', \mathrm{in}', \mu' \rangle$. Given a real number $0 \leqslant r \leqslant 1$, the probabilistic sum $\mathcal{S} \oplus_r \mathcal{S}' : n \to p$ is defined to be $\langle S \cup S' \cup I, \mathrm{in}'', \mu'' \rangle$, where $I = \{s_1, \ldots, s_n\}$ is disjoint from S and S', $\mathrm{in}''(i) = s_i$, and

$$\mu''_{s_i} = r \cdot \mu_{\mathrm{in}(i)} + (1-r) \cdot \mu'_{\mathrm{in}'(i)}\,,$$

$\mu''_s = \mu'_s$ for $s \in S$, and $\mu''_s = \mu'_s$ for $s \in S'$.

The operations defined above form the basis of the iteration theory of **LMP**s defined in Section 8. More precisely, this theory is predicated on **LMP**s modulo bisimilarity, where the bisimilarity relation is extended from states to **LMP**s by the definition below.

Definition 3. *We say that two* **LMP**s $\mathcal{S}, \mathcal{S}' : n \to p$ *are bisimilar, written* $\mathcal{S} \simeq \mathcal{S}'$*, if there is a bisimulation* R *on the tuple* $\langle \mathcal{S}, \mathcal{S}' \rangle$ *such that* $s_i R s_{i+n}$ *for each* $i \in [n]$ *where* s_j *is the* j*-th entry point of* $\langle \mathcal{S}, \mathcal{S}' \rangle$*.*

4 A Monoid of Trace Trees

In this section we present a grammar for a class of trees corresponding to branching-time traces of an **LMP**. This language (minus exit actions) corresponds to the test languages of [17, 1, 6, 9] which were shown to characterize, respectively, similarity in labelled transition systems, and probabilistic bisimilarity in labelled Markov processes.

Fix a finite set $\{X_1, \ldots, X_p\}$ (corresponding to the eXit points of an **LMP**). The language of trace trees is generated by the grammar

$$\tau ::= 1 \mid X_i \mid a\tau \mid \tau \cdot \tau \tag{1}$$

where $a \in \mathrm{Act}$ and $i \in [p]$.

A trace tree is either the null tree 1, an exit action X_i, an event $a \in \mathrm{Act}$ followed by the tree τ, or a branch point $\tau_1 \cdot \tau_2$. Note the distinction between prefixing (which is denoted by mere juxtaposition) and branching. We will typically elide the symbol 1 when denoting non-trivial trace trees, e.g., we write $a \cdot bc$ for $a1 \cdot bc1$. Without the branching construct '$\cdot$' the grammar above would just specify a language of traces. In order to physically realize a branching-time trace one would need to be able to duplicate the process at any point in a run, for instance, via a save-and-restore construct.

Definition 4. *Given an* **LMP** $\mathcal{S} : n \to p$*, with* $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$*, for each* $s \in S$ *we define* $\tau_{\mathcal{S}}(s)$*: the probability that* s *performs tree* τ*.*

- $1_{\mathcal{S}}(s) = 1$.
- $(X_i)_{\mathcal{S}}(s) = \mu_s(i)$.
- $(a\tau)_{\mathcal{S}}(s) = \int \tau_{\mathcal{S}} d\mu_{s,a}$.
- $(\tau_1 \cdot \tau_2)_{\mathcal{S}}(s) = (\tau_1)_{\mathcal{S}}(s)(\tau_2)_{\mathcal{S}}(s)$.

The null tree is performed with probability 1 in any state. The probability that $a\tau$ is performed in any given state is the weighted average of the probability that τ is performed in the next state after an a-transition. The last clause says that probability of performing an immediately branching tree $\tau_1 \cdot \tau_2$ is the product of the probabilities of performing each branch.

Given an **LMP** $\mathcal{S} : n \to p$, for each $i \in [n]$ we define the real-valued function $\widehat{\mathcal{S}}_i$ on trace trees by $\widehat{\mathcal{S}}_i(\tau) = \tau_{\mathcal{S}}(\mathrm{in}(i))$. Thus $\widehat{\mathcal{S}}_i(\tau)$ is the probability that $\mathcal{S}$ does τ on the i-th input. The following theorem is a generalization of the main result of [9] to allow for **LMP**s with entry and exit points.

Theorem 5. *Let* $\mathcal{S}, \mathcal{T} : n \to p$ *be* **LMP**s*. Then* $\mathcal{S}$ *and* $\mathcal{T}$ *are bisimilar iff* $\widehat{\mathcal{S}}_i = \widehat{\mathcal{T}}_i$ *for all* $i \in [n]$*.*

Having used trace trees to characterize equivalence of states we move on to the dual problem: when are two trees equivalent in the sense that each state performs them with the same probability? More generally we consider a preorder $\leqslant$ on trace trees defined by $\tau \leqslant \tau'$ iff $\tau_{\mathcal{S}} \leqslant \tau'_{\mathcal{S}}$ for all **LMP**s $\mathcal{S}$. The key to constructing a model for **LMP**s which is fully abstract with respect to bisimilarity is to axiomatize this preorder. Below we give a list of equations which are sufficient. Together they say that the set of trace trees forms a commutative monoid equipped with the smallest partial order in which 1 is the top element and prefixing and multiplication are monotone.

$$\begin{array}{cc} 1 \cdot \tau = \tau & \tau \leqslant 1 \\ \tau_1 \cdot \tau_2 = \tau_2 \cdot \tau_1 & \tau_1 \cdot \tau \leqslant \tau_2 \cdot \tau \text{ if } \tau_1 \leqslant \tau_2 \\ \tau_1 \cdot (\tau_2 \cdot \tau_3) = (\tau_1 \cdot \tau_2) \cdot \tau_3 & a\tau_1 \leqslant a\tau_2 \text{ if } \tau_1 \leqslant \tau_2 \end{array}$$

We denote the resulting partially ordered monoid $\mathbb{T}[p]$, where, as the reader may recall, the set of exit actions $\{X_1, \ldots, X_p\}$ was indexed over $[p]$. In case $p = 0$ we just write $\mathbb{T}$.

5 Stone-Gelfand-Naimark Duality

Our basic reference for this section is the monograph of Johnstone [15, Chapter IV 4]. We define C^*-algebras to be certain types of commutative rings. The category $\mathsf{C^* - Alg}$ is the resulting full subcategory of CRng. We should emphasize that we consider C^*-algebras as algebras over $\mathbb{R}$ as opposed to the more traditional presentation as algebras over $\mathbb{C}$.

Let A be a commutative ring. Since we are primarily interested in rings of functions, we use f, g to denote typical elements of A. We say that A is an *ordered ring* if it is equipped with a partial order satisfying

$$\begin{gathered} f + g \leqslant f' + g \text{ if } f \leqslant f' \\ f \cdot g \leqslant f' \cdot g \text{ if } f \leqslant f',\ g \geqslant 0 \\ f^2 \geqslant 0\,. \end{gathered}$$

We say that an ordered commutative ring A is *Archimedean* if for all f there exists a positive integer n with $f \leqslant n \cdot 1$. Given an Archimedean (ordered) ring A which admits a $\mathbb{Q}$-algebra[2] structure one may define a seminorm[3] by

$$||f|| = \inf\{q \in \mathbb{Q} \mid -q \cdot 1_A \leqslant f \leqslant q \cdot 1_A\}. \tag{2}$$

Definition 6. *A commutative ring A is a* real C^*-algebra *if*

- *A admits a $\mathbb{Q}$-algebra structure (equivalently the additive group of A is torsion free and divisible), and*

[2] That is, $(A, +)$ is a vector space over the rationals $\mathbb{Q}$, and scalar multiplication is compatible with multiplication in A.

[3] Non-zero elements can have norm zero.

– *A possesses an Archimedean partial order such that (2) defines a norm with respect to which A is complete.*

Definition 7. *A* character *of a C^*-algebra A is a ring homomorphism $\varphi\colon A \to \mathbb{R}$. The* spectrum *of A, denoted* $\mathrm{Spec}(A)$*, is the space of characters of A in the* Zariski topology*, which is generated by the cozero sets* $\mathrm{coz}(f) = \{\varphi : \varphi(f) \neq 0\}$ *where $f \in A$.*

It turns out that the spectrum of a C^*-algebra is a compact Hausdorff space. Conversely, the ordered ring $C^*(X)$ of continuous real-valued functions on a compact Hausdorff space X is always a C^*-algebra. This association of compact Hausdorff spaces and C^*-algebras is functorial, and is in fact a dual equivalence:

Theorem 8. *(Stone) The category* KHaus *of compact Hausdorff spaces and continuous maps is dually equivalent to* $\mathsf{C^*-Alg}$.

6 A Family of C^{Λ}-algebras

In this section we extend the monoid of trace trees to a C^*-algebra whose spectrum will be the state space of a universal **LMP**.

Fix a set $\{X_1, \ldots, X_p\}$ of exit points. We extend the grammar (1) for trace trees to a grammar of *functional expressions* by allowing rational linear combinations. Thus functional expressions are given by

$$f ::= q \mid X_i \mid af \mid f \cdot f \mid f + f \tag{3}$$

where $a \in \mathrm{Act}$, $i \in [p]$ and $q \in \mathbb{Q}$.

Note that we use the letters f and g to denote functional expressions. We adopt the convention that a term denoted τ has been generated using only the sub-grammar (1). We reserve the phrase trace tree for such terms.

We use functional expressions as generators in a presentation of a family of ordered rings $\mathbb{O}[p]$, where the index p indicates the dependance on the set $\{X_1, \ldots, X_p\}$ of exit variables. In this presentation '$\cdot$' acts as multiplication, 1 is the multiplicative identity, and $+$ acts as addition in $\mathbb{O}[p]$.

The relations in the presentation of $\mathbb{O}[p]$ include the equations for an ordered ring: the Abelian group axioms for $+$, the commutative monoid axioms for '$\cdot$', the distributive law of '$\cdot$' over $+$, and axioms asserting the compatibility of the order relation with the ring structure. To these equations we add (4–7) below. The effect of these equations is to fix the semantics of prefixing as integration against a subprobability measure. Note that the distributive law (7) implies that every functional expression is equal to the linear sum of trace trees.

$$0 \leqslant X_i \tag{4}$$

$$af \leqslant ag \text{ if } f \leqslant g \tag{5}$$

$$\textstyle\sum_{a \in \mathrm{Act}} a + \sum_{i=1}^{p} X_i \leqslant 1 \tag{6}$$

$$a(q_1 \cdot f + q_2 \cdot g) = q_1 \cdot af + q_2 \cdot ag \tag{7}$$

Definition 9. *Define* $\mathbb{O}[p]$ *to be the free ordered ring*[4] *generated by the set of functional expressions and satisfying equations (4–7).*

Proposition 10. $\mathbb{O}[p]$ *is a torsion-free divisible Archimedean ordered ring.*

$\mathbb{O}[p]$ is Archimedean since each functional expression is equal to a linear combination of trace trees, and each trace tree τ satisfies $\tau \leqslant 1$.

Definition 11. *The* C^**-algebra* $\mathbb{A}[p]$ *is defined to be the Cauchy completion of* $\mathbb{O}[p]$ *in the norm (2). The ring operations on* $\mathbb{O}[p]$ *are nonexpansive in this norm, so they extend to* $\mathbb{A}[p]$.

Proposition 12. $\mathbb{A}[p]$ *is the free* C^**-algebra over* $\mathbb{O}[p]$ *qua ordered ring.*

Remark 13. Combining Definition 9 and Proposition 12 we see that in order to specify a ring homomorphism from $\mathbb{A}[p]$ to a C^*-algebra R it suffices to give an interpretation of the functional expressions in R such that the relations in the presentation of $\mathbb{O}[p]$ all hold. Since the interpretation of $+$ and $\cdot$ is forced, this boils down to interpreting prefixing $a(-)$ and exit actions X_i.

Definition 14. *Let* $\mathcal{S} : n \to p$ *be an* **LMP** *with* $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$. *We define a ring homomorphism*

$$\mathbb{A}[p] \xrightarrow{(-)_{\mathcal{S}}} C^*(S)$$

by the following clauses:

$$(af)_{\mathcal{S}}(s) = \int_S f_{\mathcal{S}} d\mu_{s,a}$$
$$(X_i)_{\mathcal{S}}(s) = \mu_s(i)\,.$$

Furthermore we define $\widehat{\mathcal{S}}_i \in \mathrm{Spec}(\mathbb{A}[p])$ *by* $\widehat{\mathcal{S}}_i(f) = f_{\mathcal{S}}(\mathrm{in}(i))$.

Note that this extends Definition 4. Indeed, since every element of $\mathbb{O}[p]$ is equal to a linear combination of trace trees, an element of $\mathrm{Spec}(\mathbb{A}[p])$ is determined by an order preserving monoid homomorphism $\mathbb{T}[p] \to [0,1]$ which satisfies equation (6).

7 Universal LMPs

In this section we show how to define a universal **LMP** on p outputs—$\mathcal{U}[p]$.

The state space of $\mathcal{U}[p]$ is defined to be $\mathrm{Spec}(\mathbb{A}[p])$. In order to manufacture the transition probabilities we use the Riesz representation theorem [20]. First some terminology: A linear map $\varphi : C^*(X) \to \mathbb{R}$ is said to be *positive* if $\varphi(f) \geqslant 0$ whenever $f \geqslant 0$.

[4] Note in passing that the existence of a free ordered ring on a given set of generators and relations can be seen to follow from the existence of free algebras for Horn clause theories.

Theorem 15. *(Riesz) Let X be a compact Hausdorff space and $\varphi\colon C^*(X) \to \mathbb{R}$ a positive linear map. Then there is a unique Borel measure μ on X such that $\varphi(f) = \int f d\mu$ for all $f \in C^*(X)$.*

Given $\varphi \in \mathrm{Spec}(\mathbb{A}[p])$ and $a \in \mathrm{Act}$, let $\varphi_a\colon \mathbb{A}[p] \to \mathbb{R}$ be defined by $\varphi_a(f) = \varphi(af)$. The distributive law (7) ensures that φ_a is a linear map. φ_a is also positive since φ is positive and prefixing is monotone in $\mathbb{A}[p]$. We define $\mu_{\varphi,a}$ to be the Borel subprobability measure corresponding to the linear map

$$C^*(\mathrm{Spec}(\mathbb{A}[p])) \cong \mathbb{A}[p] \xrightarrow{\varphi_a} \mathbb{R}.$$

Note the application of Theorem 8 in the above isomorphism.

We now define the transition behaviour of φ by $\varphi \mapsto \mu_\varphi$, where

$$\mu_\varphi = \sum_{a \in \mathrm{Act}} \mu_{\varphi,a} + \sum_{i=1}^{p} \varphi(X_i)\delta_i\,.$$

Equations (4) and (6) guarantee that μ_φ is a subprobability measure.

We have not defined a set of entry points to $\mathcal{U}[p]$. Nevertheless it is convenient to admit $\mathcal{U}[p]$ as a partially defined **LMP**. In order to state the universal property of $\mathcal{U}[p]$, we define the notion of zig-zag [10]

Definition 16. *Let $\mathcal{S}, \mathcal{S}'$ be* **LMP***s on p outputs. Suppose that $\mathcal{S} = \langle S, \mathrm{in}, \mu\rangle$ and $\mathcal{S}' = \langle S', \mathrm{in}', \mu'\rangle$. A function $h\colon S \to S'$ is a zig-zag map iff*

- *$\mu_{s,a}(h^{-1}(t)) = \mu_{h(s),a}(t)$ for all $s \in S, t \in S'$ and $a \in \mathrm{Act}$.*
- *$\mu_s(i) = \mu_{h(s)}(i)$ for all $s \in S$, $i \in [p]$.*

Proposition 17. *Let $\mathcal{S}, \mathcal{S}'$ be* **LMP***s on p outputs. A function $h\colon S \to S'$ is a zig-zag map iff the kernel of h is a bisimulation.*

Proposition 18. *Let $\mathcal{S} = \langle S, \mathrm{in}, \mu\rangle$ be an* **LMP** *on p outputs. Then a function $h\colon S \to \mathrm{Spec}(\mathbb{A}[p])$ is a zig-zag map $\mathcal{S} \to \mathcal{U}[p]$ iff the transpose $\overline{h}\colon \mathbb{A}[p] \to C^*(S)$ satisfies*

- *$\overline{h}(af)(s) = \int_S \overline{h}(f) d\mu_{s,a}$, and*
- *$\overline{h}(X_i)(s) = \mu_s(i)$ for all $s \in S, i \in [p]$.*

By Remark 13 there is a unique map satisfying the clauses in Proposition 18—namely the map $(-)_{\mathcal{S}}$ from Definition 14. Thus we obtain:

Theorem 19. *$\mathcal{U}[p]$ is final in the category whose objects are* **LMP***s on p outputs and whose morphisms are zig-zag maps.*

In conjunction with Proposition 17, the finality of $\mathcal{U}[p]$ implies that the relation of bisimilarity on a given **LMP** $\mathcal{S}$ is the kernel of the unique zig-zag map to $\mathcal{U}[p]$. In this way we recover Theorem 5.

Corollary 20. *Given an* **LMP** *$\mathcal{S}\colon n \to p$, there is a unique* **LMP** *$\mathcal{T}\colon n \to p$ extending $\mathcal{U}[p]$ such that $\mathcal{S} \simeq \mathcal{T}$. $\mathcal{T}$ is defined by assigning $\widehat{\mathcal{S}}_i$ as the i-th entry state for each $i \in [n]$.*

8 Iterative Theories

In this section we define an iterative theory of **LMP**s modulo bisimilarity. They build on algebraic theories by adding a fixed point operator †. Iterative theories arose in the semantics of flowchart algorithms, and have since been applied to study, among other things, regular and context-free languages, synchronization trees and Floyd-Hoare logic [7].

The canonical representative of an **LMP** $\mathcal{S}: n \to p$ up-to bisimulation has been defined to be the n-tuple $\langle \widehat{\mathcal{S}}_1, \ldots, \widehat{\mathcal{S}}_n \rangle$, where $\widehat{\mathcal{S}}_i \in \mathrm{Spec}(\mathbb{A}[p])$. Below we combine the $\widehat{\mathcal{S}}_i$ into a single ring homomorphism $\widehat{\mathcal{S}}: \mathbb{A}[p] \to \mathbb{A}[n]$. Note the reversal of direction in going from $\mathcal{S}$ to $\widehat{\mathcal{S}}$. The point of this new representation is that we want to equate composition of **LMP**s with functional composition on the dual side.

Definition 21. *$\widehat{\mathcal{S}}$ is defined by the following two clauses. (Recall from Remark 13 that to define a ring map $\mathbb{A}[p] \to \mathbb{A}[n]$ one must explain how to interpret prefixing and exit variables from the grammar for functional expressions in the target ring.)*

$$\widehat{\mathcal{S}}(af) = a\widehat{\mathcal{S}}(f) + \sum_{i=1}^{n} X_i \cdot \widehat{\mathcal{S}}_i(af)$$

$$\widehat{\mathcal{S}}(X_j) = \sum_{i=1}^{n} X_i \cdot \widehat{\mathcal{S}}_i(X_j)\,.$$

Clearly $\widehat{S}$ so defined is determined by $\langle \widehat{\mathcal{S}}_1, \ldots, \widehat{\mathcal{S}}_n \rangle$. Conversely, defining the 'projections' $\pi_i: \mathbb{A}[n] \to \mathbb{R}$, where $i \in [n]$, by

$$\pi_i(af) = 0 \text{ and } \pi_i(X_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

we get that $\widehat{\mathcal{S}}_i = \pi_i \circ \widehat{\mathcal{S}}$. Note that these projections are the representations as characters of the 'injection' processes defined in Section 3. Proposition 22 now follows from Theorem 5.

Proposition 22. *$\widehat{\mathcal{S}} = \widehat{\mathcal{T}}$ iff $\mathcal{S}$ and $\mathcal{T}$ are bisimilar.*

The following definition gives a necessary and sufficient condition for a ring homomorphism $\theta: \mathbb{A}[p] \to \mathbb{A}[n]$ to arise as $\widehat{S}$ for some **LMP** $S: n \to p$.

Definition 23. *A ring homomorphism $\theta: \mathbb{A}[p] \to \mathbb{A}[n]$ is called a* shapely *map if $\theta(X_j)$ is a linear combination of the variables $\{X_1, \ldots, X_n\}$ for each $j \in [p]$, and $\theta(af)$ is equal to the sum of $a\theta(f)$ and a linear combination of variables $\{X_1, \ldots, X_n\}$ for each functional expression af.*

To explain the terminology, note that a shapely map sends a trace tree $\tau \in \mathbb{T}[p]$ to a linear combination of those trees in $\mathbb{T}[n]$ obtained by replacing actions in τ by exit actions from the set $\{X_1, \ldots, X_n\}$. Given a shapely map θ, we can recover θ as $\widehat{S}$, where $\mathcal{S}: n \to p$ is the **LMP** with components $\langle \pi_1 \circ \theta, \ldots, \pi_n \circ \theta \rangle$ as constructed in Corollary 20.

Example 24. Let $a, b, c \in \mathrm{Act}$. A shapely map $\mathbb{A} \to \mathbb{A}[1]$ maps $ab \cdot c$ to a linear combination of the trees $ab \cdot c$, $X_1 \cdot c$, $ab \cdot X_1$, $aX_1 \cdot c$, $aX_1 \cdot X_1$, $X_1 \cdot X_1$.

The following lemma explains the intuition that $\widehat{\mathcal{T}}$ acts like a predicate transformer.

Lemma 25. *Suppose that $\mathcal{S} : n \to m$ and $\mathcal{T} : m \to p$ are* **LMP***s. Then*

$$f_{\mathcal{S};\mathcal{T}}(s) = (\widehat{\mathcal{T}} f)_{\mathcal{S}}(s)\,.$$

for all states $s \in \mathcal{S}$.

In the following sequence of propositions we characterize the operations of composition, iteration and probabilistic choice on **LMP**s as constructions on shapely maps. We deal first with composition of **LMP**s, showing that it corresponds to functional composition of shapely maps. Note that we write composition of **LMP**s using diagrammatic notation, whereas composition of shapely maps is the usual functional composition notation.

Proposition 26. *Suppose that $\mathcal{S} : n \to m$ and $\mathcal{T} : m \to p$ are* **LMP***s, and define $\mathcal{V} = \mathcal{S} \fatsemi \mathcal{T}$. Then $\widehat{\mathcal{V}} = \widehat{\mathcal{S}} \circ \widehat{\mathcal{T}}$.*

Probabilistic choice of shapely maps is implemented as convex combination.

Proposition 27. *Let $\mathcal{S}, \mathcal{T} : n \to p$ be* **LMP***s and define $\mathcal{V} = \mathcal{S} \oplus_r \mathcal{T}$. Then $\widehat{\mathcal{V}}$ is specified by*

$$\widehat{\mathcal{V}}(af) = r \cdot \widehat{\mathcal{S}}(af) + (1 - r) \cdot \widehat{\mathcal{T}}(af)$$
$$\widehat{\mathcal{V}}(X_i) = r \cdot \widehat{\mathcal{S}}(X_i) + (1 - r) \cdot \widehat{\mathcal{T}}(X_i)$$

Proposition 28. *Let $\mathcal{S} : n \to n + p$ be an* **LMP** *with $\widehat{\mathcal{S}}_i(\sum_{j=1}^{n} X_j) < 1$ for all $i \in [n]$. Let $\mathcal{I}_p : p \to p$ be the 'identity'* **LMP** *$\mathcal{I}_p = \langle \{s_1, \ldots, s_p\}, \mathrm{in}, \mu \rangle$, with $\mathrm{in}(i) = s_i$ and $\mu_{s_i}(i) = 1$, and define $\mathcal{V} = \langle \dagger\mathcal{S}, \mathcal{I}_p \rangle$. Then $\widehat{\mathcal{V}}$ satisfies*

$$\widehat{\mathcal{V}}(af) = a\widehat{\mathcal{V}}(f) + \sum_{i=1}^{n} X_i \cdot \widehat{\mathcal{S}}_i(\widehat{\mathcal{V}}(af))$$
$$\widehat{\mathcal{V}}(X_j) = \sum_{i=1}^{n} X_i \cdot \widehat{\mathcal{S}}_i(\widehat{\mathcal{V}}(X_j)) + X_{n+j} \quad \text{for } j \in [p].$$

Note that the term $\widehat{\mathcal{V}}(af)$ occurs on both sides of the first clause above.

The following proposition asserts that the Elgot fixed point identity [7] holds up to bisimulation for **LMP**s. The proof works by showing that both sides denote the same shapely map.

Proposition 29. *Let $\mathcal{S} : n \to n + p$ be an* **LMP***. Then $\dagger\mathcal{S} \simeq \mathcal{S} \fatsemi \langle \dagger\mathcal{S}, \mathcal{I}_p \rangle$*

In the remainder of this section we turn the above characterizations of operations on **LMP**s as operations on shapely maps into a *definition* of an iterative theory. Our main reference for iterative theories is [7]. It is standard to denote composition in an iterative theory using diagrammatic notation.

Definition 30. *A* theory *is a category whose objects are the non-negative integers, where n is the n-th copower of 1. For each n we pick a coproduct cocone of 'distinguished morphisms' $\kappa_n^i : 1 \to n$ for $i \in [n]$. Given a family of morphisms $g_i : 1 \to p$, $i \in [n]$, the unique morphism $g : n \to p$ with $\kappa_n^i \fatsemi g = g_i$ is denoted $\langle g_1, \ldots, g_n \rangle$.*

Definition 31. *A* pointed theory *is a theory with a distinguished morphism $\bot : 1 \to 0$: the 'point'. An* ideal theory *is a theory with the property that if $f : 1 \to p$ is not a distinguished morphism, then neither is $f \fatsemi g$ for any $g : p \to n$. In an ideal theory, the morphisms $n \to p$ none of whose components is distinguished are called* ideal morphisms. *An* iterative theory *is an ideal theory with the property that for each ideal morphism $f : n \to n + p$ there is a unique solution $\dagger f : n \to p$ to the fixed point equation $\dagger f = f \fatsemi \langle \dagger f, 1_p \rangle$.*

The pointed iterative theory LMP is specified by the following data.

- Objects are the non-negative integers.
- A morphism $\varphi : n \to p$ is a shapely map $\varphi^* : \mathbb{A}[p] \to \mathbb{A}[n]$.
- Given $\varphi : n \to m$ and $\psi : m \to p$, the composition $\varphi \fatsemi \psi$ is defined by $(\varphi \fatsemi \psi)^* = \varphi^* \circ \psi^*$.
- The identity $\mathrm{id}_n : n \to n$ is defined by $(\mathrm{id}_n)^* = \mathrm{id}_{\mathbb{A}[n]}$.
- The point $\bot : 1 \to 0$ is the shapely map defined by $\bot^*(af) = a\bot^*(f)$.
- The coproduct injection $\kappa_n^i : 1 \to n$ is the shapely map defined by $(\kappa_n^i)^*(af) = a(\kappa_n^i)^*(f)$ and $(\kappa_n^i)^*(X_i) = X_1$ if $i = j$ and 0 otherwise.

It remains to define the class of ideal maps and iteration.

Given a morphism $\varphi : n \to p$, for $i \in [n]$ define $\varphi_i^* = \pi_i \circ \varphi^* : \mathbb{A}[p] \to \mathbb{R}$. We say that a morphism $\varphi : n \to n + p$ is *ideal* if $\varphi_i^*(\sum_{j=1}^n X_j) < 1$ for all $i \in [n]$. Suppose that $\varphi : n \to n + p$ is an ideal map and let $\theta : n + p \to p$ be defined by induction on functional expressions by the clauses:

$$\theta^*(af) = a\theta^*(f) + \sum_{i=1}^{n} X_i \cdot \varphi_i^*(\theta^*(at))$$

$$\theta^*(X_j) = \sum_{i=1}^{n} X_i \cdot \varphi_i^*(\theta^*(X_j)) + X_{n+j} \text{ for } j \in [p].$$

Note that the definition of $\theta^*(af)$ is recursive. Using the fact that φ is ideal and the complete normed structure of $\mathbb{A}[n + p]$ it is readily seen that $\theta^*(af)$ is well-defined. Finally, $\dagger\varphi : n \to p$ is defined by $\dagger\varphi = \kappa \fatsemi \theta$ where $\kappa : n \to n + p$ is the coproduct injection $\langle \kappa_{n+p}^1, \ldots, \kappa_{n+p}^n \rangle$. The proof of Proposition 29 can be carried over to show that $\dagger\varphi$ satisfies the Elgot fixed point equation.

9 Summary and Future Work

In programming and semantical frameworks, there are usually many different ways to represent the same computational behaviours. In concurrency, canonical

representatives of the equivalence classes of bisimilar processes are represented as elements of final coalgebras, often constructed in categories of domains. The applicability of such theories hinges on convenient representations of those elements.

The final coalgebra capturing **LMP**s has been described in [8]. The domain-theoretic treatment is in [12, 9]. The issues of representation have so far not been tackled. In the present paper, a method for obtaining canonical representatives of **LMP**s has been presented. Their states are represented as simple monoid homomorphisms. The effectiveness of this representation supports hope for a wider practical applicability of the **LMP** model.

The presented application of the Stone-Gelfand-Naimark duality in deriving canonical representatives of **LMP**s is an instance of a general approach to representing computational behaviours by lifting dualities, and adjunctions. A detailed account of this general framework, with applications to other computational structures, will be described in forthcoming work.

Unlike the papers [10, 12, 9] we did not emphasize the measure-theoretic aspects of **LMP**s, but stuck to the discrete case. As we already said, the idea was to communicate the essential concepts with the minimum overhead. However, another reason for this policy is that treating **LMP**s at the level of measurable spaces sits rather uneasily with the assumption of finite sets of entry and exit points. This suggests that an interesting direction for further work is to allow the domain and codomain of an **LMP** to be measurable spaces. This would yield a category of measurable spaces and **LMP**s. It would be interesting to compare such a category to the category of probabilistic relations studied in [4].

The iteration theory LMP is a subtheory of an *abstract matricial theory* [7]. Any such theory can be represented as a theory of modules on a semiring. This suggests some potential connections between our approach and that of Abramsky and Vickers [3].

Finally we would like to investigate connections between our representation of **LMP**s and the notion of formal tree series, and between shapely maps and transducers of formal tree series [13].

References

1. S. Abramsky. Observation equivalence as a testing equivalence. *Theoretical Computer Science*, 53:225–241, 1987.
2. S. Abramsky. A Domain Equation for Bisimulation. *Information and Computation* 92:161–218, 1991.
3. S. Abramsky and S. Vickers. Quantales, observational logic and process semantics. *Mathematical Structures in Computer Science*, 3:161–227, 1993.
4. S. Abramsky, R. Blute and P. Panangaden. Nuclear and Trace Ideals in Tensor-*-categories. *Journal of Pure and Applied Algebra*, 143:3–47, 1999.
5. L. Aceto, Z. Ésik and A. Ingólfsdóttir. Equational Axioms for Probabilistic Bisimilarity. In *Proceedings of 9th AMAST*, Lecture Notes in Computer Science, volume 2422, pages 239–253, 2002.
6. B. Bloom and A. Meyer. Experimenting with process equivalence. *Theoretical Computer Science*, 101:223–237, 1992.

7. S. Bloom and Z. Esik. The Equational Logic of Fixed Points. *Theoretical Computer Science*, 179:1–60.
8. F. van Breugel and J. Worrell. An Algorithm for Quantitative Verification of Probabilistic Transition Systems. In *Proceedings of CONCUR'01*, volume 2154 of *LNCS*, Springer-Verlag, 2001.
9. F. van Breugel, M. Mislove, J. Ouaknine and J. Worrell. An Intrinsic Characterization of Approximate Probabilistic Bisimilarity. In *Proceedings of FOSSACS'03* , volume 2620 of *LNCS*, Springer-Verlag, 2003.
10. J. Desharnais, A. Edalat and P. Panangaden. A Logical Characterization of Bisimulation for Labelled Markov Processes. In *Proc. 13th IEEE Symposium on Logic in Computer Science*, pages 478-487, Indianapolis, 1998. IEEE.
11. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for Labeled Markov Processes. In *Proc. 10th International Conference on Concurrency Theory*, volume 1664 of *LNCS*, Springer-Verlag, 1999.
12. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating Labeled Markov Processes. *Information and Computation*, 184(1):160–200, 2003.
13. Z. Esik and W. Kuich. Formal Tree Series. *Journal of Automata Languages and Combinatorics*, 8(2):145–185, 2003.
14. D. Kozen. The Semantics of Probabilistic Programs. *Journal of Computer and System Science*, 22:328–350, 1981.
15. P. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.
16. B. Jonsson, K. Larsen and W. Yi. Probabilistic Extensions of Process Algebras. In J.A. Bergstra, A. Ponse and S. Smolka, editors, *Handbook of Process Algebra*, pages 685–710, Elsevier, 2001.
17. K.G. Larsen and A. Skou. Bisimulation through Probabilistic Testing. *Information and Computation*, 94(1):1–28, 1991.
18. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
19. M. Mislove, J. Ouaknine and J. Worrell. Axioms for Probability and Nondeterminism. To appear, In *Proc. EXPRESS'03*, 2003.
20. K.R. Parthasarathy. *Probability Measures on Metric Spaces*. Academic Press, 1967.
21. A. Di Pierro, C. Hankin, and H. Wiklicky. Quantitative Relations and Approximate Process Equivalences. In *Proceedings of CONCUR'03*, volume 2761 of *LNCS*, Springer-Verlag, 2003.
22. E.W. Stark and S.A. Smolka. A complete axiom system for finite-state probabilistic processes. In *Proof, Language, and Interaction: Essays in Honour of Robin Milner*. MIT Press, 2000.

10 Appendix

Lemma 25. *Suppose that* $\mathcal{S} : n \to m$ *and* $\mathcal{T} : m \to p$ *are* **LMP***s. Then*

$$f_{\mathcal{S};\mathcal{T}}(s) = (\widehat{\mathcal{T}} f)_{\mathcal{S}}(s)\,.$$

for all states $s \in \mathcal{S}$.

Proof. Write $\mathcal{S} = \langle S, \mathrm{in}, \mu \rangle$ and $\mathcal{T} = \langle S', \mathrm{in}', \mu' \rangle$. Write $\langle S'', \mathrm{in}'', \mu'' \rangle$ for the composition $\mathcal{S} \fatsemi \mathcal{T}$ as defined in Section 3.

First note that it is straightforward that $f_{\mathcal{S};\mathcal{T}}(t) = f_{\mathcal{T}}(t)$ for all $t \in T$. We now prove the lemma by induction on functional expressions. The induction step for prefixing is as follows.

$$\begin{aligned}
(af)_{\mathcal{S};\mathcal{T}}(s) &= \int_{S''} f_{\mathcal{S};\mathcal{T}} d\mu''_{s,a} \\
&= \int_{S} f_{\mathcal{S};\mathcal{T}} d\mu_{s,a} + \sum_{i=1}^{m} \mu_s(i) \int_{S'} f_{\mathcal{S};\mathcal{T}} d\mu'_{\mathrm{in}'(i),a} \\
&= \int_{S} (\widehat{\mathcal{T}} f) d\mu_{s,a} + \sum_{i=1}^{m} \mu_s(i) \widehat{\mathcal{T}}_i(af) \\
&= a(\widehat{\mathcal{T}} f)(s) + \sum_{i=1}^{m} (X_i)_{\mathcal{S}}(s) \widehat{\mathcal{T}}_i(af) \\
&= \widehat{\mathcal{T}}(af)(s) \,. \Box
\end{aligned}$$

Proposition 29. *Let $\mathcal{S} : n \to n+p$ be an* **LMP**. *Then $\dagger\mathcal{S} \simeq \mathcal{S} \fatsemi \langle \dagger\mathcal{S}, \mathcal{I}_p \rangle$*

Proof. For simplicity we give the proof in the case $n = 1$ and $p = 0$. In view of Proposition 22 it suffices to show that $\widehat{\mathcal{S}} \circ \widehat{\mathcal{V}} = \widehat{\mathcal{V}} : \mathbb{A} \to \mathbb{A}[1]$, where $\mathcal{V} = \dagger\mathcal{S}$. We show by induction on functional expressions f that $\widehat{\mathcal{S}}(\widehat{\mathcal{V}}(f)) = \widehat{\mathcal{V}}(f)$.

The induction step for prefixing is as follows.

$$\begin{aligned}
\widehat{\mathcal{S}}(\widehat{\mathcal{V}}(af)) &= \widehat{\mathcal{S}}(a\widehat{\mathcal{V}}(f) + X_1 \cdot \widehat{\mathcal{S}}_1(\widehat{\mathcal{V}}(af))) \\
&= \widehat{\mathcal{S}}(a\widehat{\mathcal{V}}(f)) + X_1 \cdot \widehat{\mathcal{S}}_1(X_1)\widehat{\mathcal{S}}_1(\widehat{\mathcal{V}}(af)) \\
&= a\widehat{\mathcal{S}}(\widehat{\mathcal{V}}(f)) + X_1 \cdot \widehat{\mathcal{S}}_1(a\widehat{\mathcal{V}}(f)) + X_1 \cdot \widehat{\mathcal{S}}_1(X_1)\widehat{\mathcal{S}}_1(\widehat{\mathcal{V}}(af)) \\
&= a\widehat{\mathcal{V}}(f) + X_1 \cdot \widehat{\mathcal{S}}_1(\widehat{\mathcal{V}}(af)) \\
&= \widehat{\mathcal{V}}(af) \,. \Box
\end{aligned}$$