To appear in Proc. IEEE Logic in Computer Science, 2015.

Abstract Hidden Markov Models: a monadic account of quantitative information flow

Annabelle McIver
Dept. Computing
Macquarie University
Sydney, Australia
Email: annabelle.mciver@mq.edu.au

Carroll Morgan
School of Comp. Sci. and Eng.
Univ. New South Wales
Sydney, Australia
Email: carrollm@cse.unsw.edu.au

Tahiry Rabehaja
Dept. Computing
Macquarie University
Sydney, Australia
Email: tahiry.rabehaja@mq.edu.au

***Abstract*—Hidden Markov Models, *HMM*'s, are mathematical models of Markov processes whose state is hidden but from which information can leak via channels. They are typically represented as 3-way joint probability distributions.**

We use *HMM*'s as denotations of probabilistic hidden-state sequential programs, after recasting them as "abstract" *HMM*'s, i.e. computations in the Giry monad $\mathbb{D}$, and equipping them with a partial order of increasing security. However to encode the monadic type *with hiding* over state $\mathcal{X}$ we use $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ rather than the conventional $\mathcal{X}\to\mathbb{D}\mathcal{X}$. We illustrate this construction with a very small Haskell prototype.

We then present *uncertainty measures* as a generalisation of the extant diversity of probabilistic entropies, and we propose characteristic analytic properties for them. Based on that, we give a "backwards", uncertainty-transformer semantics for *HMM*'s, dual to the "forwards" abstract *HMM*'s.

Finally, we discuss the Dalenius desideratum for statistical databases as an issue in semantic compositionality, and propose a means for taking it into account.

***Index Terms*—Abstract Hidden Markov Models, Giry Monad, Quantitative Information Flow.**

I. Introduction

A. Setting and overview

Probabilisitic sequential programs with hidden state are effectively Hidden Markov Models, or *HMM*'s, formulated as joint probability distributions over initial state, observations, and final state. We recast *HMM*'s as computations over the Giry monad, suitable for program semantics. Indeed the monadic view of Markov processes in particular is well established [1], [2], using $\mathcal{X}\to\mathbb{D}\mathcal{X}$ where type-constructor $\mathbb{D}$ makes distributions on its base type $\mathcal{X}$; the Kleisli extension is then of type $\mathbb{D}\mathcal{X}\to\mathbb{D}\mathcal{X}$, representing action of multiplying an initial-state-distribution vector by a Markov matrix. But that does not account for hidden state and information flow.

We include hidden state by beginning with $\mathbb{D}\mathcal{X}$ (rather than $\mathcal{X}$): the computation type we obtain is "one level up", of type $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$, the extension is $\mathbb{D}^2\mathcal{X}\to\mathbb{D}^2\mathcal{X}$; and we call the double-distribution type *hyper-distributions*.

Although the Giry monad is formuated in terms of general measures [2], we will need only discrete distributions for matrix-based *HMM*'s. Nevertheless we give our constructions and results in more general terms, anticipating e.g. infinite sequences of *HMM*'s, nondeterminism, and iterations for which proper measures will be necessary [3].

In earlier work we have used $\mathbb{D}^2\mathcal{X}$, equipped with a partial order of increasing security, to establish compositionality results [4], to explore the effect of including demonic non-determinism [5] and to give an abstract treatment of probabilistic channels [6], [7]. A second common theme has been the generalisation of entropies (such as Shannon) to a more abstract setting where only their essential properties are preserved [4], [5], [7], [8]. Here we use monads to bring all those separate strands together and go further.

One further step is to show that there is a dual backwards view for abstract *HMM*'s, based on "uncertainty" *transformers* that transform post- uncertainty measures into pre- uncertainty measures where, in turn, *uncertainty measures* generalise probabilistic entropies.

We and others have argued that specific entropies (e.g. Shannon) have limitations in security work generally [4], [9]. Therefore we focus here on their essential properties: continuity and concavity. That view is supported by powerful theorems that such a generalisation supports, and a methodological criterion that uncertainty measures capture contexts in a way that individual styles of entropy cannot.

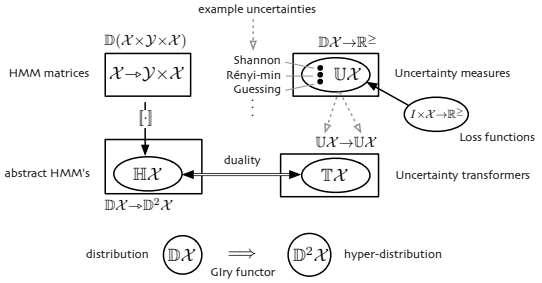
A second further step is to extend our recent treatment [7] of the *Dalenius Desideratum*, the "collateral" leakage of information due to unknown correlations with third-party data, from merely channels (a "read only" scenario [10], [11], such as access to a statistical database) to programs that might alter the database (thus "read/write" as well). The Dalenius perspective here is the fact that care must be taken wrt. compositionality in a context containing variables to which a program fragment does not explicitly refer [4].

To remain accessible to a broader security community, we do not begin from Giry: rather we first work in elementary terms. In §VI the monadic structures will be seen to have informed our earlier definitions and theorems.

B. Principal contributions and aims of the paper: summary

Our principal contributions are these, in which the **new constructions and results** are given in bold:

— We note that (finite) classical *HMM*'s are a model for straight-line sequential probabilistic programs with hidden state (top-left Fig. 1 shows the result type once applied to a prior of type $\mathbb{D}\mathcal{X}$, §III-A).



— Fig. 1. Relationship between the semantic spaces —

— We formulate *abstract HMM's*, that is $\mathbb{H}\mathcal{X}$ as a **monadic model for *HMM*'s** over state $\mathcal{X}$, and **give their characteristic properties** (bottom left: §III-B,V,VII).
— We formulate *uncertainty measures* $\mathbb{U}\mathcal{X}$ as a generalisation of diverse entropies (top centre), and **give their characteristic properties** (top right: §VIII-A). [1]
— We note that uncertainty measures **have a complete representation** as *loss functions* (centre right: §IX).
— **We give a dual, uncertainty-transformer semantics $\mathbb{T}\mathcal{X}$ of $\mathbb{H}\mathcal{X}$, stating its characteristic properties (bottom right) and proving that they enable the duality with $\mathbb{H}\mathcal{X}$ (centre: Thm. 29 in §VIII-B).**
— We show how all of this is an instance of the general Giry monad as computation, of which (finite) *HMM*'s use a discrete portion (bottom centre: §VI).
— **We explain how the "Dalenius effect" is manifested in this framework**, and how it can be treated (§XI).

In other sections we review abstract channels (§II-B), hyper-distributions (§II-C) and the security order (§V) on hypers.

We believe that Thm. 29, in particular its assumptions and proof, is a significant new result.

Our **principal aims** in this paper are these:
— (More abstract) We construct forward- and dual backward semantic spaces for probabilistic sequential computations over hidden state, using monadic computations and partial (refinement) orders in this new context, and formulate and prove the general properties that make them suitable for embedding finite (for the moment) *HMM*'s.
— (More concrete) We want to provide the basis for a source-level reasoning method, analogous to Hoare logic or weakest preconditions, for quantitative non-interference in sequential programs. For this, the dual, transformer semantics for *HMM*'s seems to be a necessary first step, together with a link between the social aspects of security and the mathematical behaviour of a program (§X).

The conclusion §XIII discusses **the benefits** of doing this.

C. General notations (see also App. A in the appendix)

Application of function f to argument x is written $f.x$, to reduce parentheses. It associates to the left.

Although a matrix M with rows, columns indexed by R, C is a function $R\times C\to\mathbb{R}$, we avoid constant reference to the reals $\mathbb{R}$ by writing just $R\!\to\!\!\triangleright C$ for that; similarly we write the type of a vector over X as $\overrightarrow{X}$. We write $M_{r,c}$ for the element of matrix M indexed by row r and column c; the r-th row of M is $M_{r,-}$; and the c-th column is $M_{-,c}$, of types $\overrightarrow{Y}, \overrightarrow{X}$ resp. For row- or column vector v: $\overrightarrow{I}$ we write v_i for its i-th element. Thus e.g. we have $(M_{-,c})_r = M_{r,c}$.

When multiplying vectors and matrices we assume without comment that the vector has been oriented properly, i.e. as a row or column as required. Thus v acts as a row in $v\cdot M$ but as a column in $M\cdot v$. [2]

We write for example $x\colon X$ when we are introducing x at that point (i.e. a binding occurrence); with $x\in X$ we are stating a property of some x and X already introduced.

Other specific notations are explained at first use, and a glossary in order of their occurrence is given in App. A.

II. Abstract channels and hyper-distributions

We review abstract channels as a conceptual stepping-stone to hyper-distributions, or "hypers" for short.

A. Channels and distributions as matrices and vectors

A *channel* is a (stochastic) matrix of non-negative reals with 1-summing rows; we use upper-case Roman letters like C. The rows are labelled with elements from some set $\mathcal{X}$; and the columns from some set $\mathcal{Y}$. Thus a channel typically has type $\mathcal{X}\!\to\!\!\triangleright\mathcal{Y}$; here, both $\mathcal{X}, \mathcal{Y}$ will be finite.

A distribution in $\mathbb{D}\mathcal{X}$ can be presented as a 1-summing vector $\overrightarrow{\mathcal{X}}$, usually lower-case Greek: especially π for prior; sometimes ρ for posterior; or simply δ for distribution.

Definition 1: **Weight** Let C or π be a matrix or vector resp. Then $\sum C$ or $\sum\pi$ is its *weight*, the sum $\sum_{x,y} C_{x,y}$ or $\sum_x \pi_x$ taken over all its indices. □
Thus e.g. we have $\sum C_{x,-} = \sum_y C_{x,y}$ and that C is stochastic just when $1 = \sum C_{x,-}$ for all x.

Each row $C_{x,-}$ of a channel C is a conditional probability distribution over $\mathcal{Y}$ given that particular $x\colon\mathcal{X}$. That is, the y-th element $C_{x,y}$ of $C_{x,-}$ is the probability that C takes input x to output y.

B. Informal channel semantics: abstract *channels*

A (1-summing) prior π and (stochastic) channel C together determine a joint distribution as follows.

Definition 2: **Channel applied to prior** Given a prior $\pi\colon\overrightarrow{\mathcal{X}}$ and channel $C\colon\mathcal{X}\!\to\!\!\triangleright\mathcal{Y}$ we write $\pi\triangleright C$ for the joint-distribution matrix of type $\mathcal{X}\!\to\!\!\triangleright\mathcal{Y}$ resulting from "applying" the channel to the prior, defined $(\pi\triangleright C)_{x,y} := \pi_x C_{x,y}$. [3] □
Note that matrix $\pi\triangleright C$ is not stochastic: rather because C itself is stochastic we have $\sum(\sum(\pi\triangleright C)_{x,-}) = \sum\pi_x = 1$.

A non-zero vector is normalised as follows.

Definition 3: **Normalisation** Let $\delta\colon\overrightarrow{\mathcal{X}}$ be such that $0\neq\sum\delta$. Then the *normalisation* $\mathsf{nrm}.\delta$ of δ is given by $(\mathsf{nrm}.\delta)_x := \delta_x / \sum\delta$ for each $x\colon\mathcal{X}$. □

Now for some π: $\overrightarrow{\mathcal{X}}$ and channel $C\colon\mathcal{X}\!\to\!\!\triangleright\mathcal{Y}$ define joint distribution $J\colon\mathcal{X}\!\to\!\!\triangleright\mathcal{Y}$ by $J = \pi\triangleright C$. The (marginal) probability

[1] These were studied, but less extensively, as "disorders", in [5].

[2] Thus for v: $\overrightarrow{X}$ and $M\colon X\!\to\!\!\triangleright Y$ the matrix product $v\cdot M$ is in $\overrightarrow{Y}$.

[3] Here juxtaposition is ordinary multiplication of reals.

of each output $y\colon\mathcal{Y}$ is $\sum J_{-,y}$ and, associated with each, there is a posterior distribution $\mathsf{nrm}.(J_{-,y})$ on $\mathcal{X}$.

Abstracting from the y-values, but retaining the link between the marginal probabilities and the posterior distributions, gives an informal description of our intended "abstract channel" semantics [6]. We make this precise in §II-D.

C. Hypers abstract from joint distributions

The joint-distribution matrix $J{=}\pi{\triangleright}C$ contains "too much" information if we do not need the actual value of y that led to a particular posterior. This is appropriate in security since the information leakage of a channel C wrt. a prior π concerns what an adversary can discover about π, and not the actual observations that led to that discovery. We can abstract from the observations in $\pi{\triangleright}C$ as follows.

If column y of $J = \pi{\triangleright}C$ is all zero, then that y will never occur (for any prior); thus we can omit that column.

And if two columns $y_{1,2}$ of J are proportional to each other, are *similar* (as for triangles), then we can add them since for a given prior the same posterior will be inferred for y_1 as for y_2 and the probability of inferring that posterior will be the sum of the marginal probabilities for $y_{1,2}$.[4]

Finally, a 1-1 renaming of the y-values has no effect on the posteriors and their respective probabilities; so we can remove those names as long as we retain the distinction between separate (non-zero, non-similar) columns.

Abstracting from all that arguably inessential information (about y) leaves only a distribution of posteriors on $\mathcal{X}$ and, for us, this is the semantic view. Writing in general $\mathbb{D}\mathcal{X}$ for 1-summing functions of type $\mathcal{X}{\to}\mathbb{R}^{\geq}$, a distribution over $\mathcal{X}$ has type $\mathbb{D}\mathcal{X}$ and so a distribution of such distributions has type $\mathbb{D}(\mathbb{D}\mathcal{X})$ that is $\mathbb{D}^2\mathcal{X}$.[5] Those latter are our hypers, and they are our abstraction of joint-distributions $\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$.

D. The semantic function from joints to hypers

In this section we define precisely the denotation $[\![J]\!]$ in $\mathbb{D}^2\mathcal{X}$ of a joint-distribution matrix $J\colon\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$. The principal tool for that is the "push forward", here in general form:

Definition 4: **Push-forward of a function**
Given sets $\mathcal{Z}, \mathcal{Z}'$ and function $f\colon\mathcal{Z}{\to}\mathcal{Z}'$, we write $\mathbb{D}f$ for the *push-forward* of f, a "lifted" function of type $\mathbb{D}\mathcal{Z}{\to}\mathbb{D}\mathcal{Z}'$ [12]. For $z'\colon\mathcal{Z}'$ and $\delta\colon\mathbb{D}\mathcal{Z}$ we have [6]

$$\mathbb{D}f.\delta.z' \quad := \quad \sum_{\substack{z:\,\mathcal{Z}\\ f.z=z'}} \delta.z \ . \quad [7]$$

□

We now define the semantic function itself:

Definition 5: **Reduced joint-distribution denotes hyper**
Let $J\colon\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$ satisfy $1 = \sum J$ so that it describes a discrete joint distribution in $\mathbb{D}(\mathcal{X}{\times}\mathcal{Y})$. Recalling §II-C, define a *reduced* matrix $J^{\downarrow}$ by (1) removing all-zero columns from J and (2) adding any similar columns of J together, retaining the label of only one of them. Let the remaining labels $\mathcal{Y}^{\downarrow}{\subseteq}\mathcal{Y}$ be the column-indices of this reduced matrix $J^{\downarrow}$.[8] [9]

Now define the $\mathcal{Y}^{\downarrow}$-marginal $\delta_y{:=}\sum J^{\downarrow}_{-,y}$ of $J^{\downarrow}$, and note from (1) just above that it is nowhere zero (on $\mathcal{Y}^{\downarrow}$). Define function $j\colon\mathcal{Y}^{\downarrow}{\to}\mathbb{D}\mathcal{X}$ by $j.y{:=}\,\mathsf{nrm}.J^{\downarrow}_{-,y}$, i.e. so that $j.y{\in}\mathbb{D}\mathcal{X}$ is the posterior distribution over $\mathcal{X}$ that $J^{\downarrow}$ induces given y. Note from (2) that j is an injection, a fact we use later in Lem. 14. Then $[\![J]\!]$, the hyper in $\mathbb{D}^2\mathcal{X}$ denoted by J in $\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$, is given by $[\![J]\!]{:=}\,(\mathbb{D}j).\delta$. □

An example is given at §III-D below.

E. Abstract channels — review

In earlier work [6] we described an "abstract channel" as a function from prior distributions to hypers. We restate that here in our current denotational style:

Definition 6: **Denotation of channel** Let $C\colon\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$ be a channel matrix. Its denotation, of type $\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$, is called an *abstract channel* and is defined for $\pi\colon\mathbb{D}\mathcal{X}$ by $[\![C]\!].\pi{:=}[\![\pi{\triangleright}C]\!]$, where the $[\![\cdot]\!]$ on the left is what we are defining, and the $[\![\cdot]\!]$ on the right is given by Def. 5.[10] □

In fact the prior π can be recovered from $\pi{\triangleright}C$:

Definition 7: **Average of a hyper** For hyper $\Delta\colon\mathbb{D}^2\mathcal{X}$ define its average $\mathsf{avg}.\Delta$ in $\mathbb{D}\mathcal{X}$ by

$$\mathsf{avg}.\Delta.x \quad := \quad \sum_{\delta:\,\mathbb{D}\mathcal{X}} (\Delta.\delta)(\delta.x) \qquad \text{for all } x\colon\mathcal{X}, \quad [11]$$

where we use upper-case Greek for hypers. □

We then have

$$(\mathsf{avg}.([\![C]\!].\pi))_x \;=\; (\mathsf{avg}.[\![\pi{\triangleright}C]\!])_x \;=\; \sum(\pi{\triangleright}C)_{x,-} \;=\; \pi_x \ ,$$

so that $\mathsf{avg}.([\![C]\!].\pi) = \pi$.

III. Classical- vs. abstract *HMM*'s

A. Classical HMM's: single HMM-steps as matrices

Classically a Hidden Markov Model comprises a set $\mathcal{X}$ of states, a set $\mathcal{Y}$ of observations and two stochastic matrices C, M that give resp. the *emission* probabilities $C_{x,y}$ that x will emit observation y and the *transition* probabilities $M_{x,x'}$ that x will change to x' [14]. Usually, the homogenous case, computation evolves in (probabilistic) steps each determined by the same C, M, with each output state x' becoming the following input x and with the emissions y accumulating: the steps all have the same pair C, M. In our case however, heterogeneous, we can vary the matrices from step to step, each standing for various (different) programs statements.

We show two computations in Fig. 2. If π is the distribution of incoming x, the distribution π'' of intermediate x'' is $\pi{\cdot}M^1$. The distribution of observations y_1 is $\pi{\cdot}C^1$. The second step's input x'' is the output of the first step.

[4] We write $y_{1,2}$ rather than y_1, y_2 for brevity.

[5] Thus 1-summing vectors δ in $\vec{\mathcal{X}}$ describe distributions in $\mathbb{D}\mathcal{X}$.

[6] Lifting, as in $\mathbb{D}f$, binds tightest: the conventional notation for $\mathbb{D}f.\delta.z'$ would be $(\mathbb{D}f)(\delta)(z')$, so that $(\mathbb{D}f)(\delta){\in}\mathbb{D}\mathcal{Z}'$.

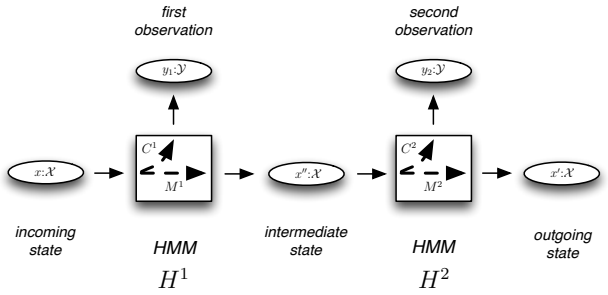
[7] $\mathbb{D}f$ is the action of functor $\mathbb{D}$ on arrow f: see §VI.

[8] The reduction is analogous to reduced *channels* in [6]. Although $J^{\downarrow}$ is not unique, the ambiguity does not affect $[\![J]\!]$.

[9] We thank a referee for suggesting that this definition might be simplified by using the converse of stochastic relations, as developed by Doberkat [13]. This is discussed further in §XII.

[10] We use $[\![\cdot]\!]$ uniformly for denotation functions, relying on context instead of e.g. using subscripts like $[\![\cdot]\!]_{\mathrm{chan}}$ and $[\![\cdot]\!]_{\mathrm{joint}}$.

[11] This avg is multiplication μ from the Giry monad: see §VI.



Each step $H^{1,2}$ takes an input- to an output state in $\mathcal{X}$; the observations $y_{1,2}{:}\,\mathcal{Y}$ are accumulated. In each step $H^{1,2}$ the output state is determined by a markov $M^{1,2}$ on the input to that step, and the observation is determined independently by a channel $C^{1,2}$ on the same input, i.e. *before* application of the markov.

Fig. 2. Two successive steps H^1 and H^2 of a heterogeneous *HMM*.

A classical *HMM* hides all of three of x, x'', x', but still the observations $y_{1,2}$ tell us something about each of them provided we know $\pi, M^{1,2}, C^{1,2}$. (This is analogous to knowing the source code of a program, but not being able to observe its variables as it executes.)

From now on we call the emission part of an *HMM* the *channel* and the transition part the *markov* (lower case).

Definition 8: **Single *HMM*-step**
Given channel $C{:}\,\mathcal{X}\!\to\!\mathcal{Y}$ and markov $M{:}\,\mathcal{X}\!\to\!\mathcal{X}$, define the *HMM*-matrix $(C{:}M)$ of type $\mathcal{X}\!\to\!\mathcal{Y}\!\times\!\mathcal{X}$ by

$$(C{:}M)_{x,y,x'} \quad := \quad C_{x,y}M_{x,x'}\ .$$

This (row-1-summing) matrix $(C{:}M)$ becomes a joint distribution of type $\mathbb{D}(\mathcal{X}\!\times\!\mathcal{Y}\!\times\!\mathcal{X})$, as top-left in Fig. 1, once the prior is fixed. [12] □

B. Abstract HMM's represent classical HMM's

For abstract channels (§II-E) we focussed on the hyper of posteriors on the *input*; for *HMM*'s we focus on the hyper of posteriors on the *output*, because *HMM*'s are computations and so it is over their outputs we wish to reason. [13]

Definition 9: **Matrix *HMM* denotes abstract *HMM***
Let $H{:}\,\mathcal{X}\!\to\!\mathcal{Y}\!\times\!\mathcal{X}$ be an *HMM* presented as a matrix. Its denotation, of type $\mathbb{D}\mathcal{X}\!\to\!\mathbb{D}^2\mathcal{X}$, is called an *abstract HMM* and is defined $[\![H]\!].\pi := [\![J]\!]$ where $\pi{:}\,\mathbb{D}\mathcal{X}$, and the joint-distribution matrix $J{:}\,\mathcal{X}\!\to\!\mathcal{Y}$ is given by $J_{x',y} := \sum_x \pi_x H_{x,y,x'}$. □

In §XI we discuss the (Dalenius) implications of having abstracted from the *HMM*'s input (with the $\sum_x$ just above).

C. Special cases of HMM's: pure markovs

Markovs are the special case of *HMM* where the channel-part effectively outputs nothing. If an *HMM*-step $(C{:}M)$ has channel C an all-one column vector nc [14] then $\mathcal{Y}$ is a singleton and J becomes a column vector: i.e. $J_{x'} = \sum_x \pi_x M_{x,x'}$, so that in fact J is the usual matrix product $\pi{\cdot}M$.

Definition 10: **One- and two-point distributions**
For $z, z'{:}\,\mathcal{Z}$ we write $[z]$ for the *point distribution* on z, assigning probability 1 to z and 0 to all other elements of $\mathcal{Z}$. [15] We write $z_p{\oplus}z'$ for the two-point distribution that assigns p to z and $1{-}p$ to z' and 0 to everything else in $\mathcal{Z}$.

Thus $z_1{\oplus}z' = [z]$ and $z_0{\oplus}z' = [z']$. □

Taking nc as the default channel gives $[\![{:}M]\!].\pi = [\![\mathsf{nc}{:}M]\!].\pi = [\pi{\cdot}M]$, the point hyper on $\pi{\cdot}M$. A general H is a markov just when $\sum_{x'} H_{x,y,x'}$ is nc.

Consider the program of Fig. 3 whose single variable is a two-bit string $\mathtt{xs}$. We model it with $\mathcal{X}{=}\{00, 01, 10, 11\}$; prior $\pi{:}\,\mathbb{D}\mathcal{X}$ is uniform, and its markov M is as just below:

The output distribution is of course $\pi'{=}\pi{\cdot}M{=}\pi$, and so the attacker's guess is optimally any of the four values in $\mathcal{X}$: they are equally good.

$$\begin{array}{c} \\ 00{:} \\ 01{:} \\ 10{:} \\ 11{:} \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{pmatrix} 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \end{pmatrix}. \end{array}$$

```
// xs is initialised uniformly at random.
xs:= xs 1/2⊕ -xs
// What does an attacker guess for xs finally?
```

The secret two-bit bit-string `xs` is set initially from $\{00, 01, 10, 11\}$ with equal probabiity $1/4$ for each; the following assignment either leaves `xs` unchanged (probability $1/2$) or bit-wise inverts all of it.

— Fig. 3. Pure-markov *HMM* program —

```
// xs is initialised uniformly at random.
print xs[0] 1/2⊕ xs[1]
```

The value of either bit 0 or bit 1 of `xs` is revealed; the attacker learns that value, but does not know which bit it came from. What should he guess for `xs` after execution in this case?

— Fig. 4. Pure-channel *HMM* program —

This system viewed as an abstract *HMM* would give output *hyper* $\Delta' = [\![{:}M]\!].\pi = [\pi]$, in fact the *point* hyper on π indicating that the attacker is certain (point-probability 1) that the posterior distribution π' on the final value of `xs` is equal to π in this case, i.e. still uniform.

D. Special cases of HMM's: pure channels

Channels are the special case where the input- and the output state are the same. If $(C{:}M)$ has markov M as the identity id, then it is a "pure channel" with output the same as its input. In that case Def. 9 gives $J_{x',y} = \sum_x \pi_x C_{x,y}\mathsf{id}_{x,x'} = (\pi{\triangleright}C)_{x',y}$, and so $[\![C{:}\mathsf{id}]\!]$ from Def. 9 is just $[\![C]\!]$ from Def. 6.

With id as the default markov, we have $[\![C{:}]\!] = [\![C]\!]$.

Now consider Fig. 4 where some of `xs` is leaked, but `xs` itself is not changed. Thus our state $\mathcal{X}$ and prior π are as

[12] Although $(C{:}M)$ has the property that for each $x{:}\,\mathcal{X}$ the (remaining) joint distribution $(C{:}M)_{x,-,-}$ is independent in y, x', this property is not preserved once steps are composed (§IV).

[13] The *prior* on the output would be our calculation from the input prior and the markov of what the output distribution would be, but *before* running the program and making observations.

[14] for "null channel"

[15] Function $[\cdot]$ is the unit η of the monad: see §VI.

before, the observation space is $\mathcal{Y}{=}\{0,1\}$ and the channel C representing this program is here at left:

$$C = \begin{array}{c} \\ 00: \\ 01: \\ 10: \\ 11: \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 0 & 1 \end{pmatrix} \end{array} \qquad J^{\downarrow} = \begin{array}{c} \\ 00: \\ 01: \\ 10: \\ 11: \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 1/4 & 0 \\ 1/8 & 1/8 \\ 1/8 & 1/8 \\ 0 & 1/4 \end{pmatrix} \end{array} .$$

The (reduced) joint distribution based on $J{=}\pi{\triangleright}C$ is above at right. (In fact no reduction was necessary.) The construction of Def. 5 gives us a hyper Δ' as

$$\begin{array}{ll} \text{“inner” distributions} & \text{“outer” distribution} \\ (1/2, 1/4, 1/4, 0) & @\ 1/2 \\ (0, 1/4, 1/4, 1/2) & @\ 1/2\ , \end{array} \tag{1}$$

where in general we write $z_1@p_1,\ z_2@p_2, \cdots$ for the discrete distribution that assigns probability p_1 to z_1 etc. In (1) the values $z_1, z_2, \cdots$ are themselves (inner, posterior) distributions. This hyper shows that with probability $1/2$ the attacker will guess `00` (because he saw a `0` printed, and deduces *a posteriori* that `00` now has the highest probability, twice either of the others; and with probability $1/2$ the attacker will guess `11` (because he saw a `1`).

We have abstracted from the printed-out $\mathcal{Y}$-values, i.e. what he saw, concentrating simply on what he deduces.

IV. *HMM* PROGRAMMING: SEQUENTIAL COMPOSITION

A. Classical HMM composition: matrices

Let $H^1, H^2{:}\,\mathcal{X}{\to}\mathcal{Y}{\times}\mathcal{X}$ be two *HMM*'s. Their sequential composition $H = H^1; H^2$ describes the distribution on x, $y_{1,2}$ together and x' as

$$(H^1; H^2)_{x,(y_1,y_2),x'} \quad := \quad \sum_{x''} H^1_{x,y_1,x''} H^2_{x'',y_2,x'} \ . \tag{2}$$

This can be seen as rewriting H^1 as type $\mathcal{X}{\times}\mathcal{Y}{\to}\!\!\!\!\circ\ \mathcal{X}$, then matrix-multipling by H^2, and then re-converting the resulting $\mathcal{X}{\times}\mathcal{Y}{\to}\!\!\!\!\circ\ \mathcal{Y}{\times}\mathcal{X}$ back to $\mathcal{X}{\to}\!\!\!\!\circ\ \mathcal{Y}^2{\times}\mathcal{X}$.[16] Note how the set of observables is now $\mathcal{Y}^2$, compounding the observations $\mathcal{Y}$ from each component. (This is why infinite composition of *HMM*'s cannot easily be represented as a finite matrix.)

Remarkably, the action of *HMM*-composition on pure-markovs *HMM*'s is effectively their matrix multiplication, yet its action on pure channels is effectively their “concatenation”: a single definition of composition specialises automatically to the two principal sub-cases. (See App. B.) Thm. 12 shows that the same happens for abstract *HMM*'s.

B. Abstract HMM's: Kleisli composition

Now we consider $h_1; h_2$ where $h_{1,2}$ are abstract *HMM*'s.[17] Because the components' types $\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$ do not match directly, i.e. the $\mathbb{D}^2\mathcal{X}$ from the left is not the $\mathbb{D}\mathcal{X}$ required on the right, we use Kleisli composition for that.[18]

[16]Lambert Meertens pointed out this nice formulation.

[17]We use upper-case for matrices and lower-case for denotations.

[18]This is the usual composition in a Kleisli category. See §VI.

```
// xs is set uniformly at random.
print xs[0] 1/2⊕ xs[1] ;
xs:= xs 1/2⊕ -xs
```

The value of either bit 0 or bit 1 of `xs` is revealed; the attacker learns that value, but does not know which bit it is. Then `xs` is either unchanged or inverted, but the attacker does not know which. What's his best guess now for the final value of `xs`?

— Fig. 5. *HMM* program as sequential composition —

Definition 11: **Kleisli composition of abstract *HMM*'s** Given two abstract *HMM*'s $h_{1,2}{:}\,\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$, their Kleisli composition is defined $(h_1; h_2).\pi := \mathsf{avg}.(\mathbb{D}h_2.(h_1.\pi))$ for $\pi{:}\,\mathbb{D}\mathcal{X}$, where $\mathbb{D}h_2$ is the push-forward of h_2 (as given in Def. 4). Equivalently $h_1; h_2 := \mathsf{avg} \circ \mathbb{D}h_2 \circ h_1$.

That is, the lifting inherent in Kleisli-composition applies the right-hand abstract *HMM* h_2 to each inner (i.e. posterior) produced by the left-hand h_1 from prior π, preserving the way in which they are all combined together by the outer distribution. Then the intermediate result, of type $\mathbb{D}^3\mathcal{X}$, is averaged to bring it back to the required type $\mathbb{D}^2\mathcal{X}$. □

C. Proof that composition is faithfully denoted

It is important (though unsurprising) for our interpretation that composition of *HMM*'s as matrices (2) is correctly mapped by $[\![\cdot]\!]$ to their Kleisli composition as abstract *HMM*'s (Def. 11). That is, we have

Theorem 12: **Composition faithfully denoted** Let $H^{1,2}{:}\,\mathcal{X}{\to}\!\!\!\!\circ\ \mathcal{Y}{\times}\mathcal{X}$ be *HMM*'s as matrices. Then we have $[\![H^1; H^2]\!] = [\![H^1]\!]; [\![H^2]\!]$, where (2) is used on the left and Def. 11 on the right.

Proof: Given in App. C. □

D. Channel and markov together: example of composition

As an example of sequential composition return to `xs` and consider Fig. 5 where it is both leaked *and* (possibly) changed. The final hyper Δ' in this case is obtained by applying the markov M to the *inners* generated by C in §III-C while retaining their outers: that gives

$$\begin{array}{ll} (& 1/2{\times}1/2 + 1/2{\times}0, \\ & 1/2{\times}1/4 + 1/2{\times}1/4, \\ & 1/2{\times}1/4 + 1/2{\times}1/4, \\ & 1/2{\times}0 + 1/2{\times}1/2 \quad)@\ 1/2 \end{array} \qquad \begin{array}{ll} (& 1/2{\times}0 + 1/2{\times}1/2, \\ & 1/2{\times}1/4 + 1/2{\times}1/4, \\ & 1/2{\times}1/4 + 1/2{\times}1/4, \\ & 1/2{\times}1/2 + 1/2{\times}0 \quad)@\ 1/2 \end{array}$$

which is simplified first to this $\to$ $\begin{array}{ll} (1/4, 1/4, 1/4, 1/4) & @\ 1/2 \\ (1/4, 1/4, 1/4, 1/4) & @\ 1/2 \end{array}$ and then, since the two inners are the same, as a distribution collapses to just $[\pi]$ again.[19]

Thus the program of Fig. 5 reveals nothing about the final value of `xs` when the intitial distribution was uniform. Informally we would explain this by noting that the information about `xs` released by the `print` becomes “stale”, irrelevant once we do not know whether `xs` has subsequently been inverted or not. (See §XI however for a discussion of why the initial value of `xs` might in some cases still be important.)

[19]We use an explicit $(\times)$ for multiplication of specific numbers.

It would be wrong however to conclude, from $\pi{=}\pi'$ in this case, that the program is secure for $\mathtt{xs}$ in general — for when the initial distribution is *not* uniform, the final value of $\mathtt{xs}$ *can* be less secure than the initial. This illustrates the danger in assuming something is uniformly distributed simply because we know nothing about it. (See App. D.)

In App. E a small Haskell prototype verifies these calculations.

V. The structure of hyper-space

Our hyper-space $\mathbb{D}^2\mathcal{X}$ has been synthesised by abstraction from the classical "matrix style" description of *HMM*'s. We now recall that there is a partial order ($\sqsubseteq$) of refinement, where for two hypers $\Delta_{S,I}{:}\,\mathbb{D}^2\mathcal{X}$ we say that Δ_S (a specification) is "refined by" Δ_I (implementation) when, in a sense we make precise below, the implementation Δ_I releases no more information than the specification Δ_S does [3]–[5], [8]. That order lifts pointwise to $\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$, i.e. that $h_S{\sqsubseteq}h_I$ just when $h_S.\pi \sqsubseteq h_I.\pi$ for all $\pi{:}\,\mathbb{D}\mathcal{X}$, thus giving a new *refinement* order for (abstract) *HMM*'s. We write $\Delta_S{\sqsubseteq}\Delta_I$, and call it "uncertainty refinement" if we need to distinguish it from other kinds of refinement. Its ultimate antecedent is the lattice of information [15], which it generalises significantly.

Definition 13: **Uncertainty refinement [3], [5]**
Let $\Delta_{S,I}{:}\,\mathbb{D}^2\mathcal{X}$ be two hypers on $\mathcal{X}$. We say that Δ_S is refined by Δ_I just when there is a distribution $\underline{\Delta}{:}\,\mathbb{D}^3\mathcal{X}$, that is a distribution of *hypers*, such that

$$\Delta_S \;=\; \mathsf{avg}.\underline{\Delta} \quad\text{and}\quad (\mathbb{D}\mathsf{avg}).\underline{\Delta} \;=\; \Delta_I \;. \qquad \square$$

Recall that $\mathbb{D}\mathsf{avg}$ is the push-forward of avg (Defs. 4,7).

The advantage of the abstract formulation in Def. 13 is that it is defined on hypers directly, and can be generalised to proper measures, thus extending discrete distributions [5]. But in the case (as here) where we remain discrete, there is an equivalent matrix-style characterisation:

Lemma 14: **Refinement of joint-distributions [4], [8]**
Let $J_S{:}\,\mathcal{X}{\twoheadrightarrow}\mathcal{Y}_S$ and $J_I{:}\,\mathcal{X}{\twoheadrightarrow}\mathcal{Y}_I$ be joint-distribution matrices, both of them *reduced* in the sense of Def. 5, such that $[\![J_{S,I}]\!]{=}\Delta_{S,I}$ resp.[20] Then

$$\Delta_S \sqsubseteq \Delta_I \quad\text{iff}\quad J_S \cdot R = J_I \tag{3}$$

for some stochastic *refinement matrix* $R{:}\,\mathcal{Y}_S{\twoheadrightarrow}\mathcal{Y}_I$. Note that the state-spaces of $\Delta_{S,I}$ are the same, but their observation spaces $\mathcal{Y}_{S,I}$ can differ.

Proof: Illustrated in App. F; sketch proof in App. G. $\square$

With Lem. 14 the reflexivity and transitivity of relation ($\sqsubseteq$) is clear from elementary matrix properties. For antisymmetry we refer to [6, Thm 6], whose supporting Lemma 1 there we adapt to suit our purposes here:

Definition 15: **Expected value** For distribution $\delta{:}\,\mathbb{D}\mathcal{Z}$ and function $f{:}\,\mathcal{Z}{\to}V$ for vector space V, the expected value of f on δ is $\mathcal{E}_\delta\, f{:=}\sum_{z:\,\mathcal{Z}} \delta_z{\times}f.z$, where $\sum$ and $(\times)$ are taken in the vector space.[21] $\square$

[20] Recall that the $\mathcal{X}$ here in type $\mathcal{X}{\twoheadrightarrow}\mathcal{Y}_S$ is the final-, not the initial state.

[21] More generally it is $\int f\,\mathrm{d}\delta$ and requires measurability of f.

We will be using $\mathcal{E}$ principally over hypers, i.e. the case $\mathcal{Z} = \mathbb{D}\mathcal{X}$ in the definition.

Lemma 16: **(Strict) monotonicity** Given are two hypers $\Delta_{S,I}{:}\,\mathbb{D}^2\mathcal{X}$ and a strictly concave function $f{:}\,\mathbb{D}\mathcal{X}{\to}\mathbb{R}^{\geq}$.

If $\Delta_S{\sqsubset}\Delta_I$ then $\mathcal{E}_{\Delta_S} f < \mathcal{E}_{\Delta_I} f$. And if f is (non-strictly) concave, then $\Delta_S{\sqsubseteq}\Delta_I$ implies $\mathcal{E}_{\Delta_S} f \leq \mathcal{E}_{\Delta_I} f$.

Proof: Proved for abstract channels in [6, Lem 1]; the proof for hypers is essentially identical. $\square$

We now have antisymmetry, because $\Delta_S{\sqsubseteq}\Delta_I{\sqsubseteq}\Delta_S$ and $\Delta_S{\neq}\Delta_I$ implies $\Delta_S{\sqsubset}\Delta_I{\sqsubset}\Delta_S$ whence we have from Lem. 16 the contradiction $\mathcal{E}_{\Delta_S} f < \mathcal{E}_{\Delta_I} f < \mathcal{E}_{\Delta_S} f$ for any strictly concave $f{:}\,\mathbb{D}\mathcal{X}{\to}\mathbb{R}^{\geq}$ of our choice (for example Shannon entropy).

Hyper-space $\mathbb{D}^2\mathcal{X}$ also admits a metric, the Kantorovich metric [16] based on the Manhattan metric on $\mathbb{D}\mathcal{X}$ (§VI). It is used for continuity properties (as we will see §VII-A).

VI. Monads: Giry, Kleisli and Kantorovich

With $\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$ we have given a discrete model of abstract *HMM*'s, suitable for interpreting probabilistic sequential programs with hidden state, together with concrete programming examples (Figs. 3–5). We now show how that embeds into structures based on a Giry monad.

The Giry/Lawvere monad based on the category *Mes* of measurable spaces comprises an endofunctor Π and two natural transformations η (unit) and μ (multiply) [2]; following [1] we take that as a basis for the denotation of computations. We have been using $\mathbb{D}$ as a specialisation of Π to the discrete case, to construct sets of *discrete* probability distributions, with unit-function $[\cdot]$ specialising η that makes a point distribution, and multiply-function avg specialising μ that takes the average of a distribution (of distributions); typically we have $[\cdot]{\in}\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$ and $\mathsf{avg}{\in}\mathbb{D}^2\mathcal{X}{\to}\mathbb{D}\mathcal{X}$.

The functions $\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$ are arrows in the related Kleisli category $(\Pi, \eta, -^\dagger)$, and our Kleisli composition for them (§IV-B) is from there.

Van Breugel compares the Giry *Mes*-monad to a "metric monad" in which a functor $\mathcal{B}$ maps 1-bounded compact metric spaces (S, d) to sets $\mathcal{B}S$ with Borel probability measures generated from the topology of the Kantorovich metric — which itself is derived from the underlying metric d on S [16]. He shows that the Giry- and metric monads are related: if from a metric space (S, d) you generate the Borel algebra $\mathcal{S}$ and thence via Giry the measurable space $(\underline{S}, \underline{\mathcal{S}}) = \Pi(S, \mathcal{S})$, then the Giry-generated σ-algebra $\underline{\mathcal{S}}$ on $\underline{S}$ is the same as the one generated by the Kantorovich metric on $\underline{S}$ derived from the original d on S.

Our use of those metrics is that we begin with a finite space $\mathcal{X}$ and we give it the discrete metric d_1. Our space $\mathbb{D}\mathcal{X}$ of discrete distributions on $\mathcal{X}$ inherits the Kantorovich metric based on d_1, which is in fact (exactly $1/2$ times) the Manhattan metric d_M on $\mathbb{D}\mathcal{X}$, that is where $d_M(\delta^1, \delta^2) = \sum_x |\delta^1_x{-}\delta^2_x|$. And our hyper-space $\mathbb{D}^2\mathcal{X}$ has the Borel algebra generated by Π from $\mathbb{D}\mathcal{X}$, which is determined by the Kantorovich metric derived from d_M. This means for us that Kantorovich continuity implies Giry measurability.

We cannot however use van Breugel's monad $\mathcal{B}$ directly because the denotations $[\![H]\!]$ of *HMM*'s in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ are not 1-Lipschitz in general, and 1-Lipschitz is a condition he imposes. They are however continuous (Lem. 17 to come); and so we use Giry instead, which imposes only measurability (implied by continuity).

Because we continue to use finite $\mathcal{X}$ below, all functions in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ are trivially measurable, thus arrows in the monad. In spite of that, we will occasionally indicate where measurability would apply in a more general treatment. On the other hand, Kantorovich-continuity of functions in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ is not automatic, since $\mathbb{D}\mathcal{X}$ is nondenumerable; and so we cannot assume that the Kleisli composition $f_1;f_2$ of two continuous functions $f_{1,2}:\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ is itself continuous: that must be proved (Lem. 21).

Giry proposes a second probability monad based on the category *Pol* of Polish spaces (rather than measurable spaces, as for *Mes* above). It is possible that this second monad simplifies our development, if the Kantorovich metric metrises the weak topology. In that case, Lem. 21 would follow directly from Giry's construction. [22]

VII. Characteristics of $\mathbb{H}\mathcal{X}$, the abstract *HMM*'s

A. Continuity and super-linearity

The semantic function $[\![\cdot]\!]$ (Def. 9) takes *HMM* matrices in $\mathcal{X}\!\!\rightarrowtriangle\!\mathcal{Y}\times\mathcal{X}$ to functions in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$; but not all of those are denotations $[\![H]\!]$ for some H. We now describe two important characteristics satisfied by $[\![H]\!]$ as H ranges over *HMM*'s: they are continuity and super-linearity. We will define abstract *HMM*'s $\mathbb{H}\mathcal{X}$ to be the functions satisfying those conditions.

Our first condition concerns continuity wrt. the Kantorovich metrics on $\mathbb{D}\mathcal{X}$ and $\mathbb{D}^2\mathcal{X}$.

Lemma 17: **Denotations of *HMM*'s are continuous** For all $H:\mathcal{X}\!\!\rightarrowtriangle\!\mathcal{Y}\times\mathcal{X}$ we have that $[\![H]\!]$ is a continuous function in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ wrt. the Kantorovich metrics.

Proof: Given in App. H. □

Our second condition concerns linear combinations.

Definition 18: **Weighted sum** For $\delta_{1,2}:\mathbb{D}\mathcal{X}$ we write $\delta_1{}_p{+}\delta_2$ for the weighted sum of the two distributions, so that $(\delta_p{+}\delta')_x = p\delta_x + (1{-}p)\delta'_x$. [23] □

Lemma 19: **Denotations of *HMM*'s are super-linear** For all $H:\mathcal{X}\!\!\rightarrowtriangle\!\mathcal{Y}\times\mathcal{X}$ we have

$$[\![H]\!].\pi_1\ {}_p{+}\ [\![H]\!].\pi_2 \quad\sqsubseteq\quad [\![H]\!].(\pi_1\ {}_p{+}\ \pi_2)\ , \tag{4}$$

where $(\sqsubseteq)$ is refinement as defined in Def. 13. [24]

Proof: Given in App. H. □

Motivated by those two lemmas, we now define

Definition 20: **The space $\mathbb{H}\mathcal{X}$ of abstract *HMM*'s** We write $\mathbb{H}\mathcal{X}$ for those h in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ satisfying Lemmas 17,19, i.e. that are Kantorovich-continuous and that satisfy $h.\pi_1{}_p{+}h.\pi_2 \sqsubseteq h.(\pi_1{}_p{+}\pi_2)$. □

[22] We thank a referee for this observation.

[23] Note that $\delta_p{+}\delta'$ and $\delta_p{\oplus}\delta'$ differ: the former is a single distribution made from merging δ,δ'; the latter is a hyper with support δ,δ'.

[24] Super-linearity can also be seen as a form of monotoniticy. See App. I1.

Thus our two lemmas above establish that $[\![H]\!]{\in}\mathbb{H}\mathcal{X}$ for any classical *HMM* H.

Since we will therefore be restricting our denotations to $\mathbb{H}\mathcal{X}$, a subset of the arrows in the Giry monad, we expect $\mathbb{H}\mathcal{X}$ to be closed under composition.

Lemma 21: **Abstract *HMM*'s closed under composition** For any two $h_{1,2}:\mathbb{H}\mathcal{X}$ we have $h_1;h_2 \in \mathbb{H}\mathcal{X}$ as well, where $(;)$ is as in Def. 11.

Proof: Although a direct proof is possible, the result is much easier once we have introduced "uncertainty" transformers (§VIII), then becoming a consequence of Thm. 29 and in particular its Cor. 30, which depends crucially on the dual view we develop in §VIII. □

It is shown in App. I2 that composition in $\mathbb{H}\mathcal{X}$ is monotonic with respect to the refinement order $(\sqsubseteq)$.

This completes our construction of our forward, abstract semantics for *HMM*'s. We now propose a dual view.

VIII. A dual view: uncertainty measures, and their transformers

A. Uncertainty measures, and their relation to refinement

"Uncertainty measures" generalise the diversity of entropy measures (e.g. Shannon), the functions from distributions to reals that measure increasing disorder.

Definition 22: **Uncertainty measure** An *uncertainty measure* over $\mathcal{X}$ is a Kantorovich-continuous- and concave function in $\mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$, i.e. one taking distributions (on $\mathcal{X}$ in this case) to non-negative reals. It is intended that a distribution's greater uncertainty indicates more resilience (less vulnerability) to the distributions's being exploited by an adversary. [25]

We write $\mathbb{U}\mathcal{X}$ for the uncertainty measures over $\mathcal{X}$, and call them "*UM*'s" in the text for brevity. □

A typical example of a *UM* applied to a hyper is as follows. Given prior $\pi:\mathbb{D}\mathcal{X}$ and channel $C:\mathcal{X}\!\!\rightarrowtriangle\!\mathcal{Y}$, the resulting hyper is $\Delta{:=}[\![\pi{\triangleright}C]\!]$ and the "conditional u uncertainty" of that (compare conditional Shannon entropy) would be $\mathcal{E}_\Delta\, u$. This could be compared to the uncertainty $u.\pi$ of the prior, to give a "u-leakage" of the channel on that prior.

There is a compelling connection between *UM*'s (Def. 22) and refinement (Def. 13, Lem. 14): we have

Lemma 23: **Soundness and completeness of uncertainty measures [6]** For any hypers $\Delta_{1,2}:\mathbb{D}^2\mathcal{X}$ we have

$$\Delta_1 \sqsubseteq \Delta_2 \quad\text{iff}\quad \mathcal{E}_{\Delta_1} u \leq \mathcal{E}_{\Delta_2} u \quad\text{for all } u:\mathbb{U}\mathcal{X}.$$

□

We regard "only if" as *soundness* in the sense that if we have a witness to the refinement relation $\Delta_1{\sqsubseteq}\Delta_2$, i.e. either $\underline{\Delta}$ (Def. 13) or R (Lem. 14), then no *UM* can show Δ_2 to be less uncertain than Δ_1. It is related to the *Data-Processing Inequality*, as explained in [6].

We regard "if" as *completeness* in the sense that if refinement fails, that is $\Delta_1{\not\sqsubseteq}\Delta_2$, then there is a *UM* demonstrating the failure [4]–[6], [8].

In App. J is background on the proof of Lem. 23, whose completeness part was originally called "Coriaceous" [8].

[25] Smith's "vulnerability measure" based on Bayes Risk [9] is an uncertainty measure except that it goes in the opposite direction.

B. Abstract HMM's to UM-transformers

In §III we introduced a "forward" denotational view of *HMM*'s that takes initial distributions to final hypers. Here we take the dual view, where an *HMM* takes a "post-uncertainty" to a "pre-uncertainty".

Definition 24: **Uncertainty-measure transformers** Take $h\colon \mathbb{H}\mathcal{X}$ and $u\colon \mathbb{U}\mathcal{X}$. Define the *uncertainty transformer* $\text{wp}.h$ of type $\mathbb{U}\mathcal{X}\to\mathbb{U}\mathcal{X}$ so that for any $u\colon \mathbb{U}\mathcal{X}$ and $\pi\colon \mathbb{D}\mathcal{X}$ we have

$$\text{wp}.h.u.\pi \quad := \quad \mathcal{E}_{h.\pi}\, u \ ,$$

where on the right we are taking the expected value of u on the hyper $h.\pi$. (Because u is continuous, it is measurable.) By analogy with weakest preconditions for ordinary sequential programming [17], the *UM*-transformer $\text{wp}.h$ takes a *UM* to be applied *after* h and produces a *UM* that equivalently can be applied *before* h. (Compare also [18], [19], [20] for probabilistic/demonic sequential programs.) □

In Lem. 25 we show well definedness of Def. 24, that is that $\text{wp}.h.u$ is indeed in $\mathbb{U}\mathcal{X}$.

Lemma 25: **Well-definedness of Def. 24** If $h\colon \mathbb{H}\mathcal{X}$ is an abstract *HMM* and $u\colon \mathbb{U}\mathcal{X}$ is a *UM*, then $\text{wp}.h.u$ is in $\mathbb{U}\mathcal{X}$.

Proof: See App. K. □

C. Characteristic properties of wp.h

For $h\colon \mathbb{H}\mathcal{X}$ the *UM*-transformer $\text{wp}.h$ has a number of characteristic properties.

Lemma 26: **wp.h is linear and total** For every $h\colon \mathbb{H}\mathcal{X}$ and $t = \text{wp}.h$ we have that t is:

1) *linear* so that for $a_{1,2}\colon \mathbb{R}^{\geq}$ and $u_{1,2}\colon \mathbb{U}\mathcal{X}$ we have
$$t.(a_1u_1 + a_2u_2) \quad = \quad a_1t.u_1 + a_2t.u_2 \ ;$$
2) *monotonic*, so that $t.u_1.\delta \geq t.u_2.\delta$ for every $u_1 \geq u_2$ with $u_{1,2}\colon \mathbb{U}\mathcal{X}$ and $\delta\colon \mathbb{D}\mathcal{X}$; [26] and
3) *total*, so that $t.\mathbf{1}{=}\mathbf{1}$ where $\mathbf{1}.\delta{:=}1$ for all $\delta\colon \mathbb{D}\mathcal{X}$.

Proof: These properties are immediate from Def. 24. □

A further property of *UM*-transformers is that they are 1-Lipshitz in a certain sense:

Lemma 27: **wp.h is 1-Lipschitz** Take $h\colon \mathbb{H}\mathcal{X}$ and define $t{:=}\,\text{wp}.h$. Let $|\cdot|$ be absolute value. Then t is 1-Lipschitz in the sense that

$$\sup_{\delta\colon \mathbb{D}\mathcal{X}} |t.u_1.\delta - t.u_2.\delta| \quad \leq \quad \sup_{\delta\colon \mathbb{D}\mathcal{X}} |u_1.\delta - u_2.\delta| \ .$$

Proof: See App. M. □

Motivated by those lemmas, we define uncertainty transformers to be exactly the functions in $\mathbb{U}\mathcal{X}\to\mathbb{U}\mathcal{X}$ that satisfy the properties listed.

Definition 28: **The uncertainty transformers $\mathbb{T}\mathcal{X}$** The uncertainty transformers $\mathbb{T}\mathcal{X}$ are the functions in $\mathbb{U}\mathcal{X}\to\mathbb{U}\mathcal{X}$ that satisfy Lems. 26,27. □

We note that transformers $\mathbb{T}\mathcal{X}$ are closed under composition. In App. I3 we show that refinement for $\mathbb{T}\mathcal{X}$ is pointwise ($\leq$).

[26] We lift $\geq$ pointwise.

D. UM-transformers back to abstract HMM's

The function $\text{wp}.(\cdot)$ has been shown to be of type $\mathbb{H}\mathcal{X}\to\mathbb{T}\mathcal{X}$. Here we show that this correspondence is exact, i.e. that for every $t\colon \mathbb{T}\mathcal{X}$ there is an $h\colon \mathbb{H}\mathcal{X}$ such that $t{=}\text{wp}.h$ and, moreover, that h is unique.

The following theorem thus establishes the exact correspondence between $\mathbb{H}\mathcal{X}$ and $\mathbb{T}\mathcal{X}$, giving an analog for hidden-state probabilistic programs to the well known correspondence between demonic relations and conjunctive predicate transformers [17], or between demonic/probabilistic programs and super-linear expectation transformers [19], [21].

Theorem 29: **Characterisation of transformers** For any $t\colon \mathbb{T}\mathcal{X}$ there is a unique $h\colon \mathbb{H}\mathcal{X}$ such that $t{=}\text{wp}.h$.

Proof: Details are given in App. N. □

With these characterisations, we can prove two technical facts. In the discrete case (as earlier) they seem self-evident. In the more general setting, however, the work is mainly in ensuring well definedness (e.g. that only measurable functions are integrated, etc.) The first establishes the usual connection between composition, this time between the forwards- and backwards semantics; the second confirms that $\mathbb{H}\mathcal{X}$ is closed under composition (i.e. preserves continuity and super-linearity, Lem. 21).

Corollary 30: **Transformer composition** For any $h_{1,2}\colon \mathbb{H}\mathcal{X}$ we have that also $h_1;h_2 \in \mathbb{H}\mathcal{X}$, and furthermore that $\text{wp}.(h_1;h_2) = \text{wp}.h_1 \circ \text{wp}.h_2$.

Proof: Direct calculation shows that $\text{wp}.(h_1;h_2) = \text{wp}.h_1 \circ \text{wp}.h_2$, although the working is intricate in the general (Giry) case. Well definedness of $h_1;h_2$ itself uses the simpler properties of (functional) composition on the transformer side. See App. O. □

Also, transformer composition respects refinement (App. I4).

IX. Gain- and loss functions define uncertainty measures

A. Gain- and loss functions

Although Def. 22 of uncertainty measures is abstract, they can be made concrete via gain functions [8] or equivalently "loss functions" [4, Eqn. (5)] that encode an attacker's (e.g.) economic interest in the secrets and the cost of obtaining them. We use loss functions here.

Definition 31: **Loss function determines uncertainty measure** A *loss function* ℓ is of type $I\to\mathcal{X}\to\mathbb{R}^{\geq}$ for some index set I, with the intuitive meaning that $\ell.i.x$ is the cost to the attacker of using "attack strategy" i when the hidden value turns out actually to be x. His expected cost for an attack planned but not yet carried out is then $\mathcal{E}_\delta\,(\ell.i)$ if δ is the distribution in $\mathbb{D}\mathcal{X}$ he believes to be governing x currently.

From such an ℓ we define an uncertainty measure

$$U_\ell.\rho \quad := \quad \inf_{i\colon I} \ \mathcal{E}_\rho\,(\ell.i) \ . \tag{5}$$

When I is finite, the $\inf$ can be replaced by $\min$. □

The $\inf$ represents a rational strategy of minimising cost or risk, and a typical attacker will act as follows: he chooses the

attack strategy (i.e. he chooses i) whose expected cost $\mathcal{E}_\rho\,(\ell.i)$ to him, where ρ is the posterior in $\mathbb{D}\mathcal{X}$ he infers from his observations in $\mathcal{Y}$, will be the least.

Lemma 32: **Well-definedness for Def. 31** For any loss function $\ell\colon I{\to}\mathcal{X}{\to}\mathbb{R}^{\geq}$ the function U_ℓ in Def. 31 is continuous and concave.

Proof: We give here the proof for the finite-I case. (The infinite case is considered in [22]; it might require further assumptions on I.) Let ℓ be a loss function and U_ℓ be the associated uncertainty measure.

U_ℓ is concave: Take $\rho_{1,2}\colon \mathbb{D}\mathcal{X}$ and $p\colon [0,1]$. We have

$$\begin{array}{lll} & U_\ell.(\rho_1{}_p{+}\rho_2) & \\ = & \min_{i\colon I}\,(\mathcal{E}_{\rho_1{}_p+\rho_2}\,\ell.i) & \text{“definition } U_\ell\text{”} \\ = & \min_{i\colon I}\,(\mathcal{E}_{\rho_1}\,\ell.i_p{+}\mathcal{E}_{\rho_2}\,\ell.i) & \text{“}\lambda\delta\cdot\mathcal{E}_\delta\,u \text{ is linear”} \\ \geq & \quad(\min_{i\colon I}\,\mathcal{E}_{\rho_1}\,\ell.i) & \text{“}(\min f)_p{+}(\min g) \\ & {}_p{+}\quad(\min_{i\colon I}\,\mathcal{E}_{\rho_2}\,\ell.i) & \leq \min(f_p{+}g)\text{”} \\ = & U_\ell.\rho_1{}_p{+}U_\ell.\rho_2\ . & \text{“definition } U_\ell\text{”} \end{array}$$

U_l is continuous: Since I is finite and each function $(\lambda\rho\cdot\mathcal{E}_\rho\,\ell.i)$ is continuous (Lem. 37 in App. K), applied to $\mathbb{D}\mathcal{X}$ instead of $\mathbb{D}^2\mathcal{X}$), the function U_ℓ is also continuous. □

Remarkably, loss functions are *complete* for uncertainty measures: any uncertainty measure in $\mathbb{U}\mathcal{X}$ can be expressed as U_ℓ for some loss function ℓ in $I{\to}\mathcal{X}{\to}\mathbb{R}^{\geq}$, but possibly requiring I to be infinite [22]. Roughly speaking, this is because of the way concave functions can be expressed as the minimum of their tangential hyperplanes: the coefficients of the hyperplanes' normals are the loss functions. [27]

It is compellingly shown elsewhere how versatile loss (equiv. gain) functions are [8]. Of particular interest is that Lem. 23 applies, both in the discrete [4] and the continuous cases [5], even when uncertainties are restricted to those generated by loss functions: the “distinguishing witness” constructed for completeness is in fact a loss function [4].

X. A *UM*-TRANSFORMER EXAMPLE FOR §IV-D

A. Profiling an attacker with a loss function

In the context of Fig. 5 we imagine an attacker whose livelihood depends on his guessing whether `xs[0]=xs[1]` or not, finally. If he guesses incorrectly he loses \$1; if correctly, he breaks even (loses \$0). This is as much a mathematical- as a *social* issue: attacks will be discouraged if they are not worthwhile for the attacker in terms of his own criteria. (See also App. D for this social aspect.)

In this example, following §IX, we express the attacker's criteria as two strategies “guess same” and “guess different” (thus $I = \{\mathsf{same}, \mathsf{diff}\}$) and a loss function ℓ defined

$$\begin{array}{cccccccc} & \ell.\mathsf{same}.(00) & = & 0 & & \ell.\mathsf{diff}.(00) & = & 1 \\ \dagger & \ell.\mathsf{same}.(01) & = & 1 & & \ell.\mathsf{diff}.(01) & = & 0 \quad \ddagger \\ & \ell.\mathsf{same}.(10) & = & 1 & & \ell.\mathsf{diff}.(10) & = & 0 \\ & \ell.\mathsf{same}.(11) & = & 0 & & \ell.\mathsf{diff}.(11) & = & 1\ , \end{array}$$

based on the informal description just above: for example if `xs=01` but he guesses same, the case indicated by †, then he loses \$1; but if he guesses diff, he breaks even ‡. Using (5) we define our *UM* as $u.\delta = U_\ell.\delta =$

$$\begin{array}{rcl} & \ell.\mathsf{same}.\delta \quad \min \quad \ell.\mathsf{diff}.\delta & \\ = & \mathcal{E}_\delta\,(\ell.\mathsf{same}) \quad \min \quad \mathcal{E}_\delta\,(\ell.\mathsf{diff}) & \\ = & \quad(\delta_{00}{+}\delta_{11}) \quad \min \quad (\delta_{01}{+}\delta_{10})\ . & \end{array}$$

B. Using UM's and transformers to plan an attack

We can use our transformer semantics to answer u-dependent questions about Fig. 5 over *all* priors: we use the two we chose earlier in §IV-D as examples.

Writing $[\![P]\!]$ for the abstract *HMM* denoted by the two lines of code in Fig. 5, we have for any π that

$$\begin{array}{rcl} \mathrm{wp}.[\![P]\!].u.\pi & = & \pi_{00}\ \min\ (\pi_{01}{+}\pi_{10})/2 \\ & + & \pi_{11}\ \min\ (\pi_{01}{+}\pi_{10})/2\ . \end{array} \tag{6}$$

(See App. P below for how this wp.$(\cdot)$ is calculated.)

Now let π^5 be the prior described by the initial comment in Fig. 5. The attacker's (expected) uncertainty wrt. the *final* hyper $[\![P]\!].\pi^5$ is wp.$[\![P]\!].u$ applied to that *initial* (uniform) prior π^5, that is $\mathrm{wp}.[\![P]\!].u.\pi^5 = 1/2$ directly from (6). Since $u.\pi^5$ is also $1/2$, he is indifferent wrt. whether he should attack before or after P has been allowed to run.

Now suppose that `xs[0]=1` is known initially, thus with prior π being $(0, 0, 1/2, 1/2)$ so that u applied initially gives $u.\pi{=}1/2$. But u applied finally would give $\mathrm{wp}.[\![P]\!].u.\pi = (0 \min 1/4) + (1/2 \min 1/4) = 1/4 < 1/2$, so that it is better to attack later even though `xs` might have been altered by P. This scenario confirms that in fact for some priors, the program in Fig. 5 cannot be regarded as secure.

XI. *HMM*'S AND THE DALENIUS DESIDERATUM

Our abstracting from initial-state correlations allows a semantics for programs' *final* states alone. Sometimes however leakage from the *initial* state is important, even if that state is overwritten by the markov part of the *HMM*: what the initial state *was* might reveal information about what some other correlated state still *is*, even if that other state is not mentioned in the program at all. This general concern was raised wrt. statistical databases by Dalenius [10] who argued that it is inescapable; Dwork later gave a proof of this [11].

With our constructions here, we are able to see the Dalenius effect in programming terms. Normally if a program does not refer at all to some variable, then its meaning is independent of that variable: the variable is just “carried through”. In terms of weakest preconditions for terminating programs, this is a combination of conjunctvitivy and the fact that $\mathrm{wp}.prog.\Phi$ is just Φ whenever *prog* does not refer to Φ.

We believe that the Dalenius effect is, in programing terms, the question of compositionality wrt. unreferenced global variables, and we show here how to deal with it.

Consider a “constant” markov $M^{\mathsf{x}}_{x,x'} = \mathit{1\ if\ } x'{=}\mathsf{x} \mathit{\ else\ 0}$ for some fixed $\mathsf{x}\colon\mathcal{X}$. Then $[\![C{:}M^{\mathsf{x}}]\!] = [\![{:}M^{\mathsf{x}}]\!]$ for any channel C — any leaking by C of the initial state is ignored, because that state is overwritten by x. We now adapt our framework so that it is *not* ignored.

[27] For example Shannon entropy requires infinite I, and the encoding is then related to minimising the Kullback-Leibler divergence.

Let $C{:}\,\mathcal{X}\!\to\!\triangleright\mathcal{Y}$ be a channel and $M{:}\,\mathcal{X}\!\to\!\triangleright\mathcal{X}$ a markov, with $\mathcal{Z}$ fresh. Write $C_{\times\mathcal{Z}}$ in $(\mathcal{X}\times\mathcal{Z})\!\to\!\triangleright\mathcal{Y}$ for the expanded channel

$$(C_{\times\mathcal{Z}})_{(x,z),y} \quad := \quad C_{x,y}\ .$$

Similarly $M_{\times\mathcal{Z}}{:}\,(\mathcal{X}\times\mathcal{Z})\!\to\!\triangleright(\mathcal{X}\times\mathcal{Z})$ is given by

$$(M_{\times\mathcal{Z}})_{(x,z),(x',z')} \quad := \quad M_{x,x'} \text{ if } z{=}z' \text{ else } 0\ .$$

These definitions ensure that for $\pi{:}\,\mathbb{D}(\mathcal{X}\times\mathcal{Z})$ neither $\pi\triangleright(C_{\times\mathcal{Z}})$ nor $\pi\triangleright(M_{\times\mathcal{Z}})$ depends on the $\mathcal{Z}$ component. Take for example $\mathcal{Z}{:=}\{z_0, z_1\}$ consider C, M as below:

$$C \;=\; \begin{array}{c} \\ x_0{:} \\ x_1{:} \end{array}\!\!\begin{array}{c} \begin{array}{cc} y_0 & y_1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1/4 & 3/4 \end{pmatrix} \end{array} \qquad M \;=\; \begin{array}{c} \\ x_0{:} \\ x_1{:} \end{array}\!\!\begin{array}{c} \begin{array}{cc} x_0 & x_1 \end{array} \\ \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \end{array}$$

$$C_{\times\mathcal{Z}} \;=\; \begin{array}{l} \\ (x_0,z_0){:} \\ (x_0,z_1){:} \\ (x_1,z_0){:} \\ (x_1,z_1){:} \end{array}\!\!\begin{array}{c} \begin{array}{cc} y_0 & y_1 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1/4 & 3/4 \\ 1/4 & 3/4 \end{pmatrix} \end{array}$$

$$M_{\times\mathcal{Z}} \;=\; \begin{array}{l} \\ x_0z_0{:} \\ x_0z_1{:} \\ x_1z_0{:} \\ x_1z_1{:} \end{array}\!\!\begin{array}{c} \begin{array}{cccc} x_0z_0 & x_0z_1 & x_1z_0 & x_1z_1 \end{array} \\ \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 \end{pmatrix} \end{array}\ .$$

The definitions above show that in $C_{\times\mathcal{Z}}$ the rows of the original C are each repeated $2 = \#\mathcal{Z}$ times; and the subsequent update by $M_{\times\mathcal{Z}}$ leaves $\mathcal{Z}$ unchanged. Observe that these definitions now account for information flows with respect to initial distributions $\mathbb{D}(\mathcal{X}\times\mathcal{Z})$ where, crucially, the $\mathcal{Z}$ component is merely "carried along". But it captures the Dalenius effect mentioned, as in the following scenario.

Consider an initial distribution $\pi{:}\,\mathbb{D}(\mathcal{X}\times\mathcal{Z})$ such that $\pi_{x_i,z_j} = 1$ if and only if $i{=}j$. We see that, even though $\mathcal{Z}$ is not accessed by the program at all, if ever y_1 is observed then the $\mathcal{Z}$ component must certainly be z_1, and if y_0 is observed then it is 4 times more likely to be z_0 than z_1.

Although $\mathcal{Z}$ is arbitrary, it can be shown that this Dalenius effect on any $\mathcal{Z}$ can be determined by the *HMM* semantics *specifically in the case where* $\mathcal{X}{=}\mathcal{Z}$. That is, we do not have to consider "all $\mathcal{Z}$'s", which would be impractical. By projecting onto the $\mathcal{Z}$ component, $[\![C_{\times\mathcal{Z}}{:}M_{\times\mathcal{Z}}]\!]$ as a whole acts as a pure channel, and if $\#\mathcal{Z}{\geq}\#\mathcal{X}$ then the component matrices C and M can be completely recovered from observations made only on the composition $[\![C_{\times\mathcal{Z}}{:}M_{\times\mathcal{Z}}]\!]$.

XII. Related work

There is great diversity in approaches to information flow in (probabilistic) programs, which we have surveyed in our own earlier work [4]–[7]. Here we concentrate on general techniques for semantic constructions, in particular those based on monads, duality and refinement.

Refinement of probabilistic programs appeared in [23] where evaluations were used to construct a powerdomain for probabilistic but possibly non-terminating computations; this was extended to include demonic choice in the discrete case in [19], [21], and was significantly generalised in [24]. Our "uncertainty refinement" that combines information flow with functional properties first appeared for information flow in straight-line programs in [4], was extended to general measure spaces [5] and appeared independently for the specific case of channels [8]. Whereas Jones and Plotkin began with an underlying partial order over which to construct a probability space, our uncertainty-refinement order begins "one level up", using hyper-distributions $\mathbb{D}^2\mathcal{X}$ to encode an "attack model" that accounts for information flow.

Doberkat defines *stochastic relations* that correspond to forward-semantic functions of type $\mathcal{X}\!\to\!\mathbb{D}\mathcal{X}$ for Markov processes: these are what we generalise by going "one level up". The converse of those stochastic relations [13] might improve the presention of our Def. 5, where a hyper is extracted from a channel and a prior, i.e. from a joint distribution.

Dual models for program semantics include [17], then for probabilistic programs [18], [20] in the purely probabilistic case. Subsequently [19] added demonic choice. And [24], [25] study dual models for probability and nondeterminism and, in particular, use a version of Riesz's representation theorem.

An alternative approach to combining probability and nondeterminism is the work of Goubault-Larrecq [25]. It uses general denotations for probabilistic programs in which nondeterminism is introduced at the level of measures (by weakening the modularity law) rather than as healthy sets of measures [19], [21], [24]. That leads naturally to a backward semantics of probabilistic demonic programs because nondeterminism is captured within integration. There is thus a strong analogy between our *UM*-transformers and Goubault-Larrecq's "previsions" because both are continuous functionals that act on some set of tests (bounded continuous functions). The main difference is that our *UM*-transformers are specifically tailored to capture security semantics, which is what leads to concavity on our set of uncertainty measures. Notice moreover that Goubault-Larrecq encounters a difficulty similar to our composition of *HMM*'s, that the decomposition $[\![C{:}M]\!]$ (resp. collinearity) is not preserved by Giry composition. Indeed, both difficulties are resolved by working in a larger space, namely, the space of abstract *HMM*'s (resp. not-necessarily-collinear continuous previsions).

In [26] a dual model for Markov processes is used to prove properties about approximations of finite behaviours, and in [27] it is shown how expectation transformers relate to explicit program models described by Markov processes.

Recently Jacobs and Hasuo have explored a general categorical construction of a backward transformer semantics from a forward monadic model of probabilistic computations (discrete, continuous and quantum) [28], [29]. Their construction uses measurability as the underlying feature of "predicates", while the stronger condition of continuity is crucial for our uncertainty measures. It would be interesting to see whether an instantiation of that categorical derivation can provide more structure for what we have done.

The concave functions advocated here for analysing information-flow properties have appeared in [5], [8] and have been identified in [30] as an ingredient in privacy analysis.

XIII. Conclusions and prospects

Our principal objective was to provide an abstract setting for *HMM*'s based on well understood principles of semantic spaces. We did that using Giry's general monadic framework applied at the level of $\mathbb{D}\mathcal{X}$ (rather than $\mathcal{X}$); the resulting structures include a refinement order which is sensitive to both functional *and* information-flow properties, and they lead to a dual, transformer space supported by theorems demonstrating the duality.

More abstractly (recall §I-B), we aimed to profit by joining two ideas: the established use of *HMM*'s as descriptions of probabilistic mechanisms having hidden state, and the established use of monads for modelling computations. Our novel use of $\mathbb{D}\mathcal{X}$ in the monad, rather than the state $\mathcal{X}$ itself, is the principal innovation that allowed this; and the synthesised hyper-distribution space that results leads to other advantages (†'s below).

An immediate benefit accrues because, in monad-enabled programming languages, probabilistic-programming packages can be built very quickly and e.g. [31] is just one of many examples. Indeed the translation into real programs is almost elementary because of the powerful and general structures available: a prototype has very recently been constructed by Schrijvers [32], independently veryifying the examples in Figs. 3–6. (See App. E for an overview.)

More importantly, any monad brings with it both general equational properties and specific properties applying to the monad in question (such as those in [2]). These conceptual tools allow reasoning about the structures modelled (*HMM*'s in this case) in ways that would be obscured by their more direct operational representation (e.g. as matrices).

† The other advantages of hypers are several: one is that they abstract from differences between entropies in a way that allows all of the entropies to be used uniformly. For example, a hyper contains all the information necessary to calculate the information leakage of a particular program fragment (typically, in the security literature, a pure channel §III-D), as shown in [6], and furthermore the Kantorovich-metric structure of $\mathbb{D}\mathcal{X}$ we used earlier for channels [7] now carries over to *HMM*'s.

† Another advantage of hypers is that their partial-order enables semantics for "looping *HMM*'s" in the standard way (least fixed-point) for computer science, rather than a direct ad-hoc definition based on matrices. Indeed a typical use of *HMM*'s is to run a single *HMM*-step (§III-A) repeatedly and then to make statistical deductions about its hidden features: sophisticated mathematical tools are available for this special case [14]. Via abstract *HMM*'s we can however, in principle, handle complex, heterogeneous systems beyond (what amounts to, in the special case just above) a single loop containing just a single statement.

Our more concrete aim (again §I-B) was to allow source-level reasoning about probabilistic programs with hidden state. Historically *at the source level* this works best with backwards reasoning based on predicates (or similar) that can be embedded between program statements rather than forwards reasoning which, here, would be calculations using $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ directly.

Here our "predicates" are *UM*'s, which in this paper however are mathematical objects unsuitable for embedding directly in program texts (see App. P, last paragraph) As remarked in §IX-A, however, any *UM* can be expressed as U_l for some loss-function l which function –crucially– is indeed an expression based on program variables [22]. The added complexity introduced by the hidden state is that the program-logic based on that observation must represent the index-set (I) of the loss function; that would most likely be done by adding a special-purpose quantifier (since the loss-function index must be a bound variable within the assertion, not appearing in the program proper).

Exploiting this opportunity for a source-level quantitative logic of probabilistic hidden state is planned for future work.

Acknowledgements We're grateful for advice from Franck van Breugel, James Worrell, Tom Schrijvers and other members of IFIP WG 2.1, and for inspiration and insight from the INRIA Princess team. And we thank the referees particularly for their suggestions of related work and possible improvements. We acknowledge support from the Australian Research Council's grant DP120101413 and the INRIA équipe associée Princess; and Morgan acknowledges the support of NICTA, which is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre-of-Excellence Program.

References

[1] E. Moggi, "Computational lambda-calculus and monads," in *Proc. 4th IEEE Symp. LiCS*, 1989, pp. 14–23.

[2] M. Giry, "A categorical approach to probability theory," in *Categorical Aspects of Topology and Analysis*, ser. Lecture Notes in Mathematics. Springer, 1981, vol. 915, pp. 68–85.

[3] A. McIver, L. Meinicke, and C. Morgan, "Hidden-Markov program algebra with iteration," *Mathematical Structures in Computer Science*, 2014.

[4] ——, "Compositional closure for Bayes risk in probabilistic noninterference," in *Proc. 37th Int. Colloq. ICALP 2010, Part II*, 2010, pp. 223–235.

[5] ——, "A Kantorovich-monadic powerdomain for information hiding, with probability and nondeterminism," in *Proc. 27th Symp. LiCS*, 2012, pp. 460–70.

[6] A. McIver, C. Morgan, G. Smith, B. Espinoza, and L. Meinicke, "Abstract channels and their robust information-leakage ordering," in *Proc. 3rd Conf. PoST (ETAPS)*, ser. Lecture Notes in Computer Science, M. Abadi and S. Kremer, Eds., vol. 8414. Springer, 2014, pp. 83–102.

[7] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Additive and multiplicative notions of leakage, and their capacities," in *Proc 27th IEEE Symp. CSF*. IEEE, 2014, pp. 308–322.

[8] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Proc. 25th IEEE Symp. CSF*, Jun. 2012, pp. 265–79.

[9] G. Smith, "On the foundations of quantitative information flow," in *Proc. 12th Conf. FoSSaCS (ETAPS)*, ser. Lecture Notes in Computer Science, L. de Alfaro, Ed., vol. 5504, 2009, pp. 288–302.

[10] T. Dalenius, "Towards a methodology for statistical disclosure control," *Statistik Tidskrift*, vol. 15, pp. 429–44, 1977.

[11] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Colloq. ICALP*, 2006, pp. 1–12.

[12] D. Fremlin, *Measure Theory*. Torres Fremlin, 2000.

[13] E. Doberkat, "The converse of a stochastic relation," in *Proc. 6th Conf. FoSSaCS (ETAPS)*, ser. LNCS, A. Gordon, Ed., vol. 2620. Springer-Verlag, 2003, pp. 233–49.

[14] D. Jurafsky and J. Martin, *Speech and Language Processing*. Prentice Hall International, 2000.

[15] J. Landauer and T. Redmond, "A lattice of information," in *Proc. 6th IEEE CSFW'93*, Jun. 1993, pp. 65–70.

[16] F. van Breugel, "The metric monad for probabilistic nondeterminism," 2005, `www.cse.yorku.ca/~franck/research/drafts/monad.pdf`.

[17] E. Dijkstra, *A Discipline of Programming*. Prentice-Hall, 1976.

[18] D. Kozen, "A probabilistic PDL," in *Proc. 15th ACM Symp. Theory of Computing*. ACM, 1983, pp. 291–7.

[19] C. Morgan, A. McIver, and K. Seidel, "Probabilistic predicate transformers," *ACM Trans Prog Lang Sys*, vol. 18, no. 3, pp. 325–53, 1996.

[20] C. Jones, "Probabilistic nondeterminism," Edinburgh University, Monograph ECS-LFCS-90-105, 1990, (Ph.D. Thesis).

[21] A. McIver and C. Morgan, *Abstraction, Refinement and Proof for Probabilistic Systems*, ser. Tech Mono Comp Sci. Springer, 2005.

[22] K. Chatzikokolakis, Private communications, 2014.

[23] C. Jones and G. Plotkin, "A probabilistic powerdomain of evaluations," in *Proc. 4th IEEE Symp. LiCS*, 1989, pp. 186–95.

[24] R. Tix, K. Keimel, and G. Plotkin, "Semantic domains for combining probability and non-determinism," *Electron. Notes Theor. Comput. Sci.*, vol. 222, pp. 3–99, 2009.

[25] J. Goubault-Larrecq, "Continuous previsions," in *Proc. 16th EACSL*, ser. Lecture Notes in Computer Science, vol. 4646. Springer, 2007, pp. 542–57.

[26] P. Chaput, V. Danos, P. Panangaden, and G. D. Plotkin, "Approximating Markov processes by averaging," *J. ACM*, vol. 61, no. 1, 2014.

[27] F. Gretz, J. Katoen, and A. McIver, "Operational versus weakest pre-expectation semantics for the probabilistic guarded command language," *Perform. Eval.*, vol. 73, pp. 110–132, 2014.

[28] B. Jacobs, "Measurable spaces and their effect logic," in *Proc. 28th LiCS*, 2013, pp. 83–92.

[29] I. Hasuo, "Generic weakest precondition semantics from monads enriched with order," in *Proc. CMCS*, ser. LNCS, M. Bonsangue, Ed., vol. 8446. Springer, 2014, pp. 10–32.

[30] D. Kifer and B.-R. Lin, "Towards an axiomatization of statistical privacy and utility," 2010, Penn State Technical report: CSE-10-002.

[31] M. Erwig and S. Kollmansberger, "Probabilistic functional programming in Haskell," *Journal of Functional Programming*, vol. 16, pp. 21–34, 2006.

[32] T. Schrijvers, "A monadic model for computations that leak secrets," 2015, `www.cse.unsw.edu.au/~carrollm/LiCS-TS.html`.

[33] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.

[34] D. Blackwell, "The comparison of experiments," in *Proc. 2nd Berkely Symp. Mathematical Statistics and Probability*. Univ. California Press, 1951, pp. 93–102.

[35] R. G. Bartle, *The Elements of Integration*. John Wiley & Sons, Inc., 1966.

[36] J. R. Munkres, *Topology*. Prentice Hall, 1999.

[37] M. H. Stone, "The generalized Weierstrass approximation theorem," *Math Magazine*, vol. 21, no. 4, pp. 167–184, March 1948.

[38] M. Bačák and J. M. Browein, "On difference convexity of locally Lipschitz functions," *Optimization: A Journal of Math Prog and Oper Research*, vol. 60, no. 8-9, pp. 961–978, 2011.

[39] L. H. Loomis and S. Sternberg, *Advanced Calculus*. Jones and Bartlett Publishers, 1990.

[40] K. R. Parthasarathy, *Probability Measures on Metric Spaces*. AMS Chelsea Publishing, 1967.

[41] R. Ranga Rao, "Relations between weak and uniform convergence of measures with applications," *Annals of Mathematical Statistics*, vol. 33, no. 2, pp. 659–680, January 1962.

APPENDIX

A. Summary of notations

These entries list in first-use order the points at which notation is introduced during the exposition: a detailed explanation of each is given there.

$-^\dagger$	Kleisli extension	p.1
$f.x$ vs. $f(x)$	Function application is ".", i.e. a dot.	p.2
$\mathcal{X} \rightarrowtriangle \mathcal{Y}$	Type of matrix.	p.2
$C_{x,y}, C_{-,y}$ etc.	Elements of vectors and matrixes by index; whole rows/columns.	p.2
$\overrightarrow{\mathcal{X}}$	Type of vector.	p.2
$(\cdot)$	Matrix multiplication: vectors automatically taken as row- or column- for conformity.	p.2
$(:)$ vs. $(\in)$	Declaration vs. property.	p.2
u.c. Roman letter	Matrices: C for channels; M for transformers; H for *HMM*'s	p.2
$\mathcal{X}$	Finite set of states.	p.2
$\mathcal{Y}$	Finite set of observations.	p.2
l.c. Greek letter	Vectors, usually distributions over $\mathcal{X}$: π for priors; ρ for posteriors; δ for others.	p.2
$\sum()$	Weight (sum of elements) of vector or matrix.	p.2
$\pi \triangleright C$	Channel applied to prior.	p.2
nrm	Normalisation of distribution.	p.2
similar	wrt. columns of joint matrix.	p.3
$y_{1,2}$ vs. y_1, y_2	Former abbreviates latter.	p.3
$\mathbb{D}$	Discrete-distribution type constructor, a functor.	p.3
$\mathbb{D}^2$	Distribution-of-distributions.	p.3
hyper	Abbreviation of "hyper-distribution".	p.3
$\mathbb{D}f$	Push-forward of f.	p.3
$[\![\cdot]\!]$	Semantic function for *HMM*'s.	p.3
$J^\downarrow$	Reduced matrix.	p.3
avg	Average (of hyper); multiply in monad.	p.3
Δ	Upper-case Greek for hypers.	p.3
channel	The emission part of an *HMM*-step.	p.4
markov	The transition part of an *HMM*-step.	p.4
$(C{:}M), (C{:}), (:M)$	One-step *HMM* defined by channel and markov.	p.4
nc	Channel that releases no information.	p.4
$x_p \oplus x'$	The two-point distribution "x with probabiity p and x' with probability $1-p$".	p.4
$[\cdot]$	Point distribution.	p.4
id	Identity (Markov) transform.	p.4
"inner" distributions	Posteriors to which hyper assigns probabilities.	p.5
"outer" probabilities	Probabilities assigned by hyper to posteriors.	p.5
@	Notation for specific hyper-distributions	p.5
$(;)$	Sequential composition of *HMM*'s.	p.5
H vs. h	Matrix- vs. abstract *HMM*'s.	p.5
$(\sqsubseteq)$	Refinement relation between hypers.	p.6
$\underline{\Delta}$	A distribution of hypers.	p.6
$\mathcal{E}_\delta\, f$	Expected value.	p.6
Π, μ, η	Giry/Lawvere functor etc.	p.6
$\mathcal{B}$	Metric-monad functor.	p.6
d_K	Kantorovich metric.	p.6
d_1	Discrete metric.	p.6
d_M	Manhattan metric.	p.6
$({}_p+)$	Weighted sum of distributions.	p.7
$\mathbb{H}\mathcal{X}$	Abstract *HMM*'s on $\mathcal{X}$.	p.7
$\mathbb{U}\mathcal{X}$, *UM*'s	Uncertainty measures on $\mathcal{X}$.	p.7
wp.h	Uncertainty transformer (determined by h.)	p.8
$\mathbf{1}$	The everywhere-one function.	p.8
U_l	Uncertainty measure defined by loss-function l.	p.8
$M_{\times\mathcal{Z}}$	Dalenius $\mathcal{Z}$-extension (of *HMM* H).	p.10
$(\cdot \| \cdot)$	Parallel composition of channels.	p.14
$(\circ)$	Functional composition.	p.14
$p(x), p(X\|y)$ etc.	Conventional notations for joint distributions.	p.15
$\lceil \delta \rceil$	The support of a distribution.	p.20
$d_M(-,-)$	Manhattan distance.	p.21
$\mathbb{C}\mathcal{X}$	Continuous functions from $\mathbb{D}\mathcal{X}$ to $\mathbb{R}$.	p.23
$\| \cdot - \cdot \|_\infty$	Uniform metric on $\mathbb{U}\mathcal{X}$ and $\mathbb{C}\mathcal{X}$.	p.23
$(\lambda \cdots)$	Lambda abstraction.	p.28
$l \triangleleft \pi$	π-skewed loss function.	p.30

B. Pure channels and pure markovs [§IV-A]

In §IV-A it was remarked that the single, uniform definition of $(;)$ for *HMM*'s "essentially" treats pure markovs and pure channels differently. This is because markovs and channels are encoded differently within *HMM*'s. First we give the details for classical *HMM*'s; then further below we address abstract *HMM*'s.

1) Composition of pure markovs: The usual composition of Markov matrices $M^{1,2}\colon \mathcal{X}\!\to\!\mathcal{X}$ is via matrix multiplication $M^1\cdot M^2$, and the result is of the same type $\mathcal{X}\!\to\!\mathcal{X}$. If we do it at the *HMM*-level, we find

$$
\begin{array}{ll}
& (:M^1);(:M^2)\quad {}_{x,(y_1,y_2),x'} \\
= & \textstyle\sum_{x''}(:M^1)_{x,y_1,x''}(:M_2)_{x'',y_2,x'} \\
= & \qquad\qquad \text{"Recall from §III-C that channel } \mathsf{nc} \text{ reveals nothing."} \\
& \mathsf{nc}_{x,(y_1,y_2)}\textstyle\sum_{x''} M^1_{x,x''}M^2_{x'',x'} \\
= & (:\, M^1\cdot M^2)\quad {}_{x,(y_1,y_2),x'}\ ,
\end{array}
$$

so that $(:M^1);(:M^2) = (:\, M^1\cdot M^2)$.

2) Composition of pure channels: Concatenation of channels, which we write $C^1\|C^2$, models applying *both* channels to the same input and observing both outputs. Thus

$$C^1\|C^2\quad {}_{x,(y_1,y_2)} \quad = \quad C^1_{x,y_1}\times C^2_{x,y_2}\ .$$

This is different from channel *cascading*, which applies the second channel C^2 to the observations of the first channel C^1 via matrix multiplication. A striking distinction is that $C^1\cdot C^2$ releases no more information that C^1 alone (the Data-processing Inequality [33]), whereas $C^1\|C^2$ releases no *less* information that either of $C^{1,2}$ alone. In this case we find

$$
\begin{array}{ll}
& (C^1:);(C^2:)\quad {}_{x,(y_1,y_2),x'} \\
= & \textstyle\sum_{x''}(C^1:)_{x,y_1,x''}(C^2:)_{x'',y_2,x'} \\
= & \qquad\qquad \text{"Recall from §III-D markov } \mathsf{id} \text{ is the identity."} \\
& \textstyle\sum_{x''} C^1_{x,y_1}\mathsf{id}_{x,x''}C^2_{x'',y_2}\mathsf{id}_{x'',x'} \\
= & C^1_{x,y_1}C^2_{x,y_2}\mathsf{id}_{x,x'} \\
= & (C^1\|C^2)_{x,(y_1,y_2)}\mathsf{id}_{x,x'} \\
= & (C^1\|C^2\ :)\quad {}_{x,(y_1,y_2),x'}\ ,
\end{array}
$$

so that $(C^1:);(C^2:) = (C^1\|C^2\ :)$.

3) Pure channel followed by pure markov: Finally, note that a general *HMM*-step (§III-A) is a pure channel followed by a pure markov. We assume assume that $\mathsf{nc}_{x,y_2}\!=\!1$ if $y_2\!=\!\hat{y}$ else 0 for some fixed $\hat{y}$ in $\mathcal{Y}$, and calculate

$$
\begin{array}{ll}
& (C:);(:M)\quad {}_{x,(y_1,y_2),x'} \\
= & \textstyle\sum_{x''}(C:)_{x,y_1,x''}(:M)_{x'',y_2,x'} \\
= & \textstyle\sum_{x''} C_{x,y_1}\mathsf{id}_{x,x''}\mathsf{nc}_{x'',y_2}M_{x'',x'} \\
= & C_{x,y_1}M_{x,x'}\mathsf{nc}_{x,y_2} \qquad\qquad \text{"}\mathsf{id}_{x'',x'}\text{, 1-point rule"} \\
= & C_{x,y_1}M_{x,x'} \text{ if } y_2\!=\!\hat{y} \text{ else } 0 \qquad\qquad \text{"assumption above"} \\
= & (C{:}M)_{x,(y_1,y_2),x'}\ ,
\end{array}
$$

so that $(C:)\,;(:M) = (C{:}M)$.[28]

4) Pure markov followed by pure channel: This cannot in general be reduced to a single *HMM*-step. In $(:M);(C:)$ let both C, M be the identity. Then the observations and final state will be perfectly correlated, something that is not possible for single *HMM*-step.

5) Pure abstract markovs: Since a pure markov reveals nothing, a pure abstract markov $h\colon \mathbb{H}\mathcal{X}$ should produce only point hypers, i.e. have for all $\pi\colon \mathbb{D}\mathcal{X}$ that $h.\pi = [\rho]$ for some ρ (depending on π).

From that we can deduce that for any pure abstract markov h the effect of $\mathsf{avg}\circ h$ (on some π) is matrix multiplication by some M (independent of π). That is, for any $0\!\le\!p\!\le\!1$ we have

$$(\mathsf{avg}\circ h).(\pi_1{}_p{+}\pi_2) \quad = \quad (\mathsf{avg}\circ h).\pi_1{}_p{+}\,(\mathsf{avg}\circ h).\pi_2\ , \quad (7)$$

which property characterises matrix multiplication. This is because $h.(\pi_1{}_p{+}\pi_2){=}[\rho]$ and $h.\pi_{1,2}{=}[\rho_{1,2}]$ resp. for some $\rho, \rho_{1,2}$, together with Lem. 19, gives

$$[\rho_1]_p{+}[\rho_2] \quad \sqsubseteq \quad [\rho]\ ,$$

and the only way that can hold is if $\rho = \rho_1{}_p{+}\rho_2$, which is precisely the claim made at (7) just above.

6) Pure abstract channels: A pure channel is one that releases information about the distribution on $\mathcal{X}$ but does not change it: one can think of the transformation part as the identity matrix. Thus (7) above suggests that we should have that $\mathsf{avg}\circ h$ is the identity for a pure channel, i.e. that $\mathsf{avg}.(h.\pi){=}\pi$. This is necessary, but turns out not to be sufficient: we explore a fuller characterisation of channels later (App. Q).

[28] The reason that $(C:);(:M)$ and $(:M);(C:)$ differ in general is that in the (mathematical) definition of an *HMM*-step (e.g. Fig. 2) the emissions are determined by the input, initial state (rather than the output, final state). Had that original definition been the other way around, then we'd have had $(:M);(C:)$ as an *HMM*-step.

C. Proof of Thm. 12 for composition [§IV-C]

Having reached §IV-C we have two formulations of *HMM*'s: the original, matrix style (§III-A); and the "abstract" hyper-distribution style (§III-B), and there is a semantic function $[\![\cdot]\!]$ that takes the first to the second (§III-B). Each has its own definition of a sequential composition operator (Secs. IV-A,IV-B). It's a routine but inescapable obligation to show that the semantic function commutes with the two definitions of that composition.

Theorem 12: Composition faithfully denoted
Let $H^{1,2}\!:\mathcal{X}\!\to\!\mathcal{Y}\!\times\!\mathcal{X}$ be *HMM*'s. Then

$$[\![H^1;H^2]\!] \quad = \quad [\![H^1]\!];[\![H^2]\!] \ ,$$

where (2) is used for (;) on the left and Def. 11 for (;) on the right.

Proof: We use conventional $p(\cdot,\cdot,\cdots)$-style joint-distribution notations for this.

Write $p_1(x,y_1,x'')$ for the probability $H^1_{x,y_1,x''}$ that the joint distribution H^1 assigns to the event "x is input, and y,x'' are output"; write $p_2(x'',y_2,x')$ similarly for H^2; and write $p(x,y_1,y_2,x')$ for the (matrix) composition $H^1;H^2$ as a whole.

We follow the usual notational conventions for these p's wrt. marginal- and conditional distributions, e.g. that $p_1(y)$ is the marginal probability of the event $Y\!=\!y$ induced by H_1 and that $p_1(x|y_1,x'')$ is the conditional probability of $X\!=\!x$ given $Y_1\!=\!y_1 \wedge X''\!=\!x''$. By e.g. $p_1(Y)$ we mean the y-marginal distribution as a whole, and by $p_1(X|y_1)$ we mean the *a-posteriori* distribution on the input induced by having observed some y_1 (and having marginalised by summing over all x'').

We avoid potential divisions by zero (i.e. due to a $\mathcal{Y}$ marginal's being zero at some y) by using *wlog* reduced matrices (Def. 5).

Now fix arbitrary $\pi\!:\ \overrightarrow{\mathcal{X}}$. The first step $[\![H^1]\!].\pi$ produces hyper $p_1(X''|y_1)@p_1(y_1)$ with y_1 varying over $\mathcal{Y}$, where $p_1(x,y_1,x'') = \pi_x H^1_{x,y,x''}$.[29] By "$y_1$ varying over $\mathcal{Y}$" we mean e.g. a distribution presented in the style of (1) for all y_1 in $\mathcal{Y}$.

And from an arbitrary π'', the second step would produce $p_2(X'|y_2)@p_2(y_2)$ with y_2 (also) varying over $\mathcal{Y}$, and $p_2(x'',y_2,x') = \pi''_{x''}H^2_{x'',y_2,x'}$. To carry out the Kleisli composition, we must let the second-step prior π'' range over the inners $p_1(X''|y_1)$ from the first step.

Thus letting π'' be some $p_1(X''|y_1)$ from the first step, we get from Def. 9 that

$$\begin{array}{lll} & p_{2_{y_1}}(x'',y_2,x') & \\ = & \pi''_{x''}H^2_{x'',y_2,x'} & \\ = & p_1(X''|y_1)_{x''}H^2_{x'',y_2,x'} & \text{“set } \pi''\!:=\!p_1(X''|y_1)\text{”} \\ = & p_1(x''|y_1)H^2_{x'',y_2,x'} \ , & \end{array}$$

where the y_1-subscript in the $p_{2_{y_1}}$, which we are defining on the left, captures its dependence on p_2.

[29]Note that the omitted x in $p_1(X''|y_1)$ indicates an implicit $\sum_x$, following the usual conventions for the $p()$ notations.

With that, we have that Kleilsi composition $[\![H^1]\!];[\![H^2]\!]$ applied to π is

$$p_{2_{y_1}}(X'|y_2) \quad @\ p_1(y_1)p_{2_{y_1}}(y_2) \ , \tag{8}$$

with $y_{1,2}$ varying over $\mathcal{Y}^2$.

What we would like to know is whether this Kleisli-generated (8) is the same hyper

$$p(X'|y_1,y_2) \quad @\ p(y_1,y_2) \ , \tag{9}$$

with $y_{1,2}$ varying over $\mathcal{Y}^2$, that would result from Def. 9 applied to $\pi\triangleright(H^1;H^2)$ directly. Thus we now prove for all $y_{1,2}$ the two equalities

$$\begin{array}{rcll} \underline{\text{Kleisli-generated (8)}} & & \underline{\text{Def. 9 generated (9)}} & \\ p_1(y_1)p_{2_{y_1}}(y_2) & = & p(y_1,y_2) & (10) \\ p_{2_{y_1}}(X'|y_2) & = & p(X'|y_1,y_2) & (11) \end{array}$$

that together will establish the equality of the two hypers. Beginning with (10) we calculate

$$\begin{array}{lll} & p_1(y_1)p_{2_{y_1}}(y_2) & \\ = & p_1(y_1)\sum_{x'} p_{2_{y_1}}(y_2,x') & \\ = & \sum_{x'} p_1(y_1)p_{2_{y_1}}(y_2,x') & \\ = & \sum_{x'} p(y_1,y_2,x') & \text{“Lem. 33 below”} \\ = & p(y_1,y_2) \ . & \end{array}$$

For (11) we prove that $p_{2_{y_1}}(x'|y_2) = p(x'|y_1,y_2)$ for arbitrary x', calculating

$$\begin{array}{lll} & p_{2_{y_1}}(x'|y_2) & \\ = & p_{2_{y_1}}(y_2,x')/p_{2_{y_1}}(y_2) & \\ = & p(y_1,y_2,x')/p_1(y_1)\ /p_{2_{y_1}}(y_2) & \text{“Lem. 33”} \\ = & p(y_1,y_2,x')/p(y_1,y_2) & \text{“(10)”} \\ = & p(x'|y_1,y_2) \ . & \end{array}$$

And with those two equalities we have our result. □

The following technical lemma, used in the proof of Thm. 12, simplifies the relationship between p_1,p_2 and p:

Lemma 33: **Technical lemma** Continuing with the notations of above, we show that

$$p_1(y_1)p_{2_{y_1}}(y_2,x') \quad = \quad p(y_1,y_2,x') \ .$$

Proof:

$$\begin{array}{lll} & p_1(y_1)p_{2_{y_1}}(y_2,x') & \\ = & p_1(y_1)\sum_{x''} p_{2_{y_1}}(x'',y_2,x') & \\ = & p_1(y_1)\sum_{x''} p_1(x''|y_1)H^2_{x'',y_2,x'} & \text{“calculation above”} \\ = & \sum_{x''} p_1(y_1,x'')H^2_{x'',y_2,x'} & \\ = & \sum_{x,x''} p_1(x,y_1,x'')H^2_{x'',y_2,x'} & \\ = & \sum_{x,x''} \pi_x H^1_{x,y_1,x''}H^2_{x'',y_2,x'} & \\ = & \sum_x \pi_x \sum_{x''} H^1_{x,y_1,x''}H^2_{x'',y_2,x'} & \\ = & \sum_x p(x,y_1,y_2,x') & \\ = & p(y_1,y_2,x') \ . & \end{array}$$

□

```
// xs is set uniformly from {01,10,11}.
print xs[0] 1/2⊕ xs[1] ;
xs:= xs 1/2⊕ -xs
```

This system is as in Fig. 5 except that the prior initial distribution differs: at least one bit of xs is known to be 1.

Fig. 6. Simple-channel program excluding xs=00 initially

D. Example program with non-uniform prior [§IV-D]

In Fig. 5 we gave a small program, on a two-bit state space, that illustrated an information leak (via a print statement) followed by a probabilistic update. In the case of a uniform input (prior distribution), the program's output turned out to have the same distribution as had the input: that is, although information had been leaked by the print, that leak was rendered useless, made out-of-date by the subsesequent probabilitsic update.

The intuitive explanation for this was that e.g. a printed 0 would tell us that *at that point* the state could not be 11 (as in Fig. 4 from §III-D). But the subsequent update could convert a 11 into a 00, and so the final state could after all be 00.

We then cautioned that the conclusion that the program was (wrt. the final state) "leak free" was unjustified in general, since with a different prior there could well be proper leakage.

Here we give an example showing that to be so. In Fig. 6 the initial hyper is "skewed", that is

$$(0, 1/3, 1/3, 1/3) \qquad @\,1\ , \tag{12}$$

so that with certainty (@1) it is known that the initial distribution is $(0, 1/3, 1/3, 1/3)$. Via the first statement print xs[0] $1/2\oplus$ xs[1] an attacker will wprob. $1/3$ (resp. $2/3$) observe 0 (resp. 1) and revise his belief of xs's distribution as in the first (resp. second) row here:

$$\begin{array}{ll} (0, 1/2, 1/2, 0) & @\,1/3 \\ (0, 1/4, 1/4, 1/2) & @\,2/3 \end{array}$$

And after the second statement xs:= xs $1/2\oplus$ -xs the hyper for the *current* (and final) distribution of xs will have become

$$\begin{array}{ll} (0, 1/2, 1/2, 0) & @\,1/3 \\ (1/4, 1/4, 1/4, 1/4) & @\,2/3\ , \end{array} \tag{13}$$

where in the $1/3$-case he is better off finally than initially (since he knows xs cannot be 00 or 11), but in the other case he is worse off (since xs=00 has become possible). Thus if the attacker's choice is either to guess xs's initial value or to run the program and guess xs's final value, he can use these hypers to help make up his mind depending on his own criteria for the utility of his planned theft, the social context in which he is operating. [30]

For example, the Shannon entropy of xs is initially $\lg(3)\sim 1.6$, but finally it is conditionally $1/3\times 1 + 2/3\times 2 = 2/3 > 1.6$: if the attacker is using Shannon entropy to make his decision, he should act sooner rather than later.

On the other hand, the one-guess probability (Rényi min-entropy) of xs is initially $1/3$; and finally it is the same, at $1/3\times 1/2 + 2/3\times 1/4 = 1/3$. If the attacker is using this criterion, it does not matter when he acts.

In either case, the hypers (12) and (13) contain all the information necessary for his decision: the bit-values printed are themselves not important *for his decision*, which is why we can quotient our semantics by abstracting from them. (He does however need those values when he *makes* his attack, if he decides "later".)

These calculations are confirmed in App. E.

[30] Compare a thief's two alternatives for stealing a credit card: *"Steal it now, since the wallet is just sitting there."* and *"Steal it after the card is used at an ATM where he can see some digit of the PIN, but there is a risk his victim will notice and choose a new PIN."*

E. Overview of Haskell-monadic prototype [§IV-D,§XII]

Schrijvers has constructed a Haskell prototype of our hyper-based monadic model for discrete, finite *HMM*'s, and has applied it to our examples of Figs. 3–6 [32]. We give a brief summary here. [31]

A discrete probabilitiy distribution on a set $\mathcal{X}$ is modelled as a monadic type `Dist x` that is effectively a list `[(x,Rational)]` of elements from $\mathcal{X}$ and their associated probabilities. The type of (discrete) hypers $\mathbb{D}^2\mathcal{X}$ is then `Dist(Dist x)`.

A Markov "matrix" on $\mathcal{X}$ is of type `x->Dist x`, in fact encoding the matrix as a function from row-indices to distributions $\mathbb{D}\mathcal{X}$; a channel matrix is of type `x->Dist y` for any type $\mathcal{Y}$ of observations whatever.

The mini- programming language has two elementary statements: to use markov `mm` we have `update mm` that updates the state according to `mm`. To use channel `cm` we have `reveal cm` that emits (e.g. prints) the channel's output wrt. the hidden state at that point: the state is not changed and, in particular, the output is not assigned to anything. It is merely observed. Both of these statements are of type `Dist x->Dist(Dist x)`, modelling our $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$. In fact they are in $\mathbb{H}\mathcal{X}$, as Lemmas 17,19 show.

Sequential composition (;) of programs is the Kleisli composition `>=>` provided by Haskell's conventions for monads, in this case the monad `Dist`. Using that, and relying on §IV-C and App. B3, we can define an elementary *HMM*-step (§III-A) as

```
    hmmStep cm mm
=   reveal cm >=> update mm  .
```

This is why `hmmStep` does not have to be primitive.

Our example space $\mathcal{X}$ is `(Bool,Bool)`, representing the bit-pair `xs`, and our two example (input) priors are

```
uniform      =  [  ((False,False),1%4),
                   ((False,True),1%4),
                   ((True,False),1%4),
                   ((True,True),1%4)
                ]
```

and from App. B

```
skewed       =  [  ((False,False),0),
                   ((False,True),1%3),
                   ((True,False),1%3),
                   ((True,True),1%3)
                ]
```

Our example observation space $\mathcal{Y}$ is `Bool`.

With suitable definitions typed as above for channel `oneBit` that outputs one of `xs`'s two bits, uniformly at random, and `invert` that either inverts `xs` or does not, again uniformly at random, the four programs are then

```
fig3  =       update invert
fig4  =       reveal oneBit
fig5  =       -- Uniform prior.
              reveal oneBit
        >=>   update invert
fig6  =       -- Skewed prior.
              reveal oneBit
        >=>   update invert
```

The programs are run using the function

```
    runOn prior prog
=   pretty (prog prior)
```

where `pretty` is an output-formatting function that prints hypers in a readable way. [32]

The results of running the programs are as follows, where the third column gives the outer probabilities of the resulting hyper, and the first two columns give the corresponding inner distributions. We print `True`,`False` as `1`,`0` respectively: [33]

```
runOn uniform fig3 =
 00  0.25  1.0    point hyper
 01  0.25
 10  0.25
 11  0.25

runOn uniform fig4 =
 01  0.25  0.5    Half the time...
 10  0.25
 11  0.5          11 is most likely, and
 00  0.5   0.5    the other half it's 00.
 01  0.25
 10  0.25

runOn uniform fig5 =
 00  0.25  1.0
 01  0.25
 10  0.25
 11  0.25

runOn skewed fig6 =
 00  0.25  0.67
 01  0.25
 10  0.25
 11  0.25

 01  0.5   0.33
 10  0.5
```

The prototpe contains also a `repeat` feature: for example `repeat 10 (reveal oneBit)` is a program that reveals a random bit of `xs` 10 times independently. With the uniform prior we would expect that the resulting hyper would have three inners: one of them, occurring with probability approximately $1/2$, would correspond to the case where the input

[31] We have altered some of Schrijvers' variable names and types, in order better to correspond with our presentation above.

[32] Ironically, the pretty-printing function's definition is the longest of any individual function in the prototype.

[33] The prototype prints probabiities as fractions; but here they are printed as reals, for neatness.

bits of `xs` differed, in which case with overall probability $^{1023}/_{1024}$ there would be two different revelations among the 10 instances — thus showing that indeed the bits differed. But we would still have no (more) information about whether the input was `01` or `10`.

The remaining $^1/_{1024}$ would be split between two cases: bit `xs[0]` was revealed every time, or `xs[1]` was; and those outcomes would contribute to the other two inners.

Those other two inners would have probability approximately $^1/_4$ each, corresponding to input `00` or `11` where the two bits are the same. The program confirms this, giving [34]

```
runOn uniform
(repeat 10 (reveal oneBit)) =
 01   1/1026    513/2048   outer is about 1/4
 10   1/1026
 11   512/513   inner is "almost certainly 11"
 01   1/2       511/1024   about 1/2
 10   1/2
 00   512/513   513/2048   about 1/4
 10   1/1026
 11   1/1026
```

The small perturbations away from $^1/_4$ etc. reflect the small chance, mentioned above, that even when the inputs differ the random `oneBit` reveals the same bit 10 times in a row.

Finally, if we run the same program but with the final probabilistic inversion included, we get

```
runOn uniform
(repeat 10 (reveal oneBit)
>=> update invert) =
 00   256/513   513/1024   two inners merged
 01   1/1026
 10   1/1026
 11   256/513   about 1/2
 01   1/2       511/1024   about 1/2
 10   1/2
```

in which the two "bits equal" inners from just above have merged: although the probabilistic inversion preserves the information concerning whether the bits are equal, it conceals in the equals case whether they were both `00` or both `11`.

[34]This time we preserve the fractions.

F. Equivalent presentations of refinement: Lem. 14 [§V]

Lem. 14 concerned two definitions of uncertainty refinement, showing them to be equivalent: one was formulated for joint distributions (defined at (3) within the lemma), suitable for discrete reasoning; and the other was formulated for hypers (Def. 13), suitable for extension to more general reasoning (e.g. measures). We sketch the proof of that equivalence in App. G immediately below.

In this section we present an example, two hypers $\Delta_{S,I}$ shown to satisfy $\Delta_S \sqsubseteq \Delta_I$ in both presentations (Def. 13, Lem. 14), with an explanation of how to move from one presentation to the other.

As in (1) of §III-D, we use the following notation for discrete distribution where specific values in the support are named: we write

$$\begin{array}{l} x_1 \mathbin{@} p_1 \\ x_2 \mathbin{@} p_2 \\ \textit{etc...} \end{array} \tag{14}$$

If these are laid out horizontally, we enclose them in double set-brackets $\{\{\cdots\}\}$ separated by commas: thus $\{\{H@^2/_3, T@^1/_3\}\}$ describes a coin twice as likely to give heads as tails. If the double brackets are used without probabilities (and thus also without @'s) then the intended distribution is uniform, so that $\{\{H, T\}\}$ describes a fair coin; a convenient special case of that is $\{\{H\}\}$ for the point distribution on H, the coin that gives heads every time. [35]

Let $\mathcal{X}$ be the set $\{H, T\}$ of coin-flip results. We choose our two hypers as follows, presenting them as at (14):

$$\Delta_S \;=\; \left[\begin{array}{ll} H_{2/3}\oplus T & @\ ^1/_2 \\ H_{1/3}\oplus T & @\ ^1/_2 \end{array}\right]$$

$$\Delta_I \;=\; \left[\begin{array}{ll} H_{2/3}\oplus T & @\ ^1/_3 \\ H_{1/2}\oplus T & @\ ^1/_3 \\ H_{1/3}\oplus T & @\ ^1/_3\ . \end{array}\right]$$

The first hyper Δ_S represents choosing fairly between two biased coins and having the chosen one secretly flipped: we know which coin was flipped, but we are not allowed to see the outcome of the flip. In Δ_I however we choose fairly between *three* coins: the two biased coins from before, and a fair one. Again the chosen one is secretly flipped; again we are not allowed to see the outcome.

We argue that in any reasonable measure of secrecy, it should in the second case Δ_I be harder to guess which of H, T was flipped than in the first case Δ_S. And it is precisely that non-specific "in *any* reasonable measure" that uncertainty refinement $\Delta_S \sqsubseteq \Delta_I$ attempts to capture. [36]

In this case, and informally speaking, Δ_I is more secure than Δ_S because there is now a third possible case that acts as a linear combination of the existing two. That is, some of the separation between the inners $H_{2/3}\oplus T$ and $H_{1/3}\oplus T$ in the

[35]In the semantic space we write $[x]$ for that: here we are syntactic.

[36]Furthermore, the powerful "Coriaceous" completeness property (Lem. 23) shows the dual result: if some Δ_S, Δ_I are *not* in the refinement relation, that is $\Delta_S \not\sqsubseteq \Delta_I$, then there is *guaranteed* to be a uncertainty measure wrt. to which Δ_I is *not* more secure than Δ_S.

support of Δ_S has been merged together to become a single inner $H_{1/2}\oplus T$ in the support of Δ_I — and what makes the observer more uncertain is that he doesn't know how to pull that single inner apart again.

Two (reduced) joint matrices $J_{S,I}$ that give $\Delta_{S,I}$ resp. are

$$J_S \;=\; \begin{matrix} \\ H\colon \\ T\colon \end{matrix} \begin{matrix} a & b \end{matrix} \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}$$

$$J_I \;=\; \begin{matrix} \\ H\colon \\ T\colon \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 2/9 & 1/6 & 1/6 \\ 1/9 & 1/6 & 2/9 \end{pmatrix}$$

where the observation spaces are $\mathcal{Y}_S{=}\{a,b\}$ and $\mathcal{Y}_I{=}\{c,d,e\}$ respectively. Now the refinement matrix that establishes (according to Lem. 14) that $\Delta_S \sqsubseteq \Delta_I$ is $R\colon \mathcal{Y}_S \twoheadrightarrow \mathcal{Y}_I$ given by

$$R \;=\; \begin{matrix} \\ a\colon \\ b\colon \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 2/3 & 1/3 & 0 \\ 0 & 1/3 & 2/3 \end{pmatrix}$$

which, read columnwise, says in its column c that to make Column c of J_I you take $2/3$ of Column a of J_S and none of Column b of J_S. The middle column d of R is where the actual refinement lies, that Column d of J_I is made by adding $1/6$ of each of Columns a, b of J_S together. This is where J_I (equiv. Δ_I) reveals less than J_S (equiv. Δ_S) does about the distribution on $\mathcal{X}{=}\{H,T\}$. And, as the lemma suggests, we indeed have $J_S{\cdot}R = J_I$.

In the more abstract terms of Def. 13, the $\underline{\Delta}$ we are looking for, that establishes $\Delta_S \sqsubseteq \Delta_I$ at the hyper-level directly, can be given as (the denotation of) a joint distribution $J\colon \mathbb{D}\mathcal{X} \twoheadrightarrow \mathcal{Y}_I$ itself: we will have $\underline{\Delta} := [\![J]\!]$ which, because J's source type is $\mathbb{D}\mathcal{X}$, will have type $\mathbb{D}^2(\mathbb{D}\mathcal{X}) = \mathbb{D}^3\mathcal{X}$ as we expect from $[\![\cdot]\!]$. The rows of J will be labelled by the support of Δ_S, so that it will have only two rows; thus we have

$$J \;=\; \begin{matrix} H_{2/3}\oplus T\colon \\ H_{1/3}\oplus T\colon \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 1/3 & 1/6 & 0 \\ 0 & 1/6 & 1/3 \end{pmatrix}. \qquad (15)$$

If on the other hand we were to write $\underline{\Delta}{=}[\![J]\!]$ as a hyper directly (performing the various normalisations etc.) we would have

$$\underline{\Delta} \;=\; \begin{bmatrix} [\, H_{2/3}\oplus T \,] & @\, 1/3 \\ (H_{2/3}\oplus T)_{1/2}\oplus(H_{1/3}\oplus T) & @\, 1/3 \\ [\, H_{1/3}\oplus T \,] & @\, 1/3 \end{bmatrix} .$$

Now $\mathsf{avg}.\underline{\Delta}$ is given by the calculation

$$\begin{array}{l} [\, H_{2/3}\oplus T \,]{\times}1/3 \\ +\ (H_{2/3}\oplus T)_{1/2}\oplus(H_{1/3}\oplus T){\times}1/3 \\ +\ [\, H_{1/3}\oplus T \,]{\times}1/3 \end{array}$$

$$\begin{array}{ll} = & H_{2/3}\oplus T)_{1/2}\oplus(H_{1/3}\oplus T) \\ = & \Delta_S\ . \end{array}$$

This can also be seen (indeed is easier to see) if we simply take the left-marginal of J, for which you add the columns together: you get

$$\begin{matrix} \\ H_{2/3}\oplus T\colon \\ H_{1/3}\oplus T\colon \end{matrix} \begin{matrix} c{+}d{+}e \\ =1 \end{matrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} ,$$

which is again Δ_S.

For the other direction we obtain $(\mathbb{D}\mathsf{avg}).\underline{\Delta}$ by avg'ing each inner of $\underline{\Delta}$ while preserving the (outer) probablities. That gives

$$(\mathbb{D}\mathsf{avg}).\underline{\Delta} \;=\; \begin{bmatrix} H_{2/3}\oplus T & @\, 1/3 \\ H_{1/2}\oplus T & @\, 1/3 \\ H_{1/3}\oplus T & @\, 1/3 \end{bmatrix} ,$$

because

$$\begin{array}{rcl} \mathsf{avg}.[\, H_{2/3}\oplus T \,] & = & H_{2/3}\oplus T \\ \mathsf{avg}.(H_{2/3}\oplus T)_{1/2}\oplus(H_{1/3}\oplus T) & = & H_{1/2}\oplus T \\ \mathsf{avg}.[\, H_{1/3}\oplus T \,] & = & H_{1/3}\oplus T \end{array}$$

And so that the remaining question is "How do we get that $\underline{\Delta}$ from some R?"

Remember that the support of Δ_S is $\{H_{2/3}\oplus T, H_{1/3}\oplus T\}$. Make a distribution π_S by mapping those (inner) distributions of Δ_S onto the labels in $\mathcal{Y}_S$ associated uniquely with them in J_S. (The association is unique because J_S is reduced.) That gives us that π_S is of type $\mathbb{D}\mathcal{Y}_S$ and has value $a_{1/2}\oplus b$.

Now form the joint-distribution matrix $\pi_S \triangleright R$, i.e.

$$\begin{matrix} a\colon \\ b\colon \end{matrix} \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \triangleright \begin{matrix} \\ \\ \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 2/3 & 2/3 & 0 \\ 0 & 1/3 & 2/3 \end{pmatrix} = \begin{matrix} \\ a\colon \\ b\colon \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 1/3 & 1/6 & 0 \\ 0 & 1/6 & 1/3 \end{pmatrix}$$

which (like R itself) is of type $\mathcal{Y}_S \twoheadrightarrow \mathcal{Y}_I$. (But note that R is a channel matrix, whereas $\pi_S \triangleright R$ is a joint-distribution matrix.)

Now use the relabelling in the reverse direction on the rows of the joint distribution above (as "new row-labels" at right above) to get a matrix with the same contents but now of type $\mathbb{D}\mathcal{X} \twoheadrightarrow \mathcal{Y}_I$. It is

$$\begin{matrix} \\ H_{2/3}\oplus T\colon \\ H_{1/3}\oplus T\colon \end{matrix} \begin{matrix} c & d & e \end{matrix} \begin{pmatrix} 1/3 & 1/6 & 0 \\ 0 & 1/6 & 1/3 \end{pmatrix}$$

which is exactly the J we had at (15) above, and as above we get $\underline{\Delta}$ via $\underline{\Delta}{=}[\![J]\!]$.

Thus in this example we have illustrated how one might move between the two equivalent definitions of refinement. Each one has a witness: in the hyper-formulation it is the distribution on hypers $\underline{\Delta}$; and in the matrix formulation is is a post-processing "refinement matrix" R. The sketch proof (App. G) shows how to obtain each from the other in general.

G. Monadic vs. matrix presentations of refinement [§V]

In App. F we gave an example of the two equivalent presentations of refinement; here we give a proof (sketch) that it can always be done.

Definition 34: **Support of a distribution** Given discrete distribution $\delta\colon \mathbb{D}\mathcal{Z}$, we write $\lceil\delta\rceil$ for the *support* of δ, the set of elements $z\colon\mathcal{Z}$ for which $\delta.z$, the probability assigned by δ to z, is not zero. Obviously $\delta\in\mathbb{D}\mathcal{Z}$ implies $\lceil\delta\rceil\subseteq\mathcal{Z}$; if in fact $\lceil\delta\rceil=\mathcal{Z}$ then we say that δ is *full support*. □

Lemma 14: Refinement of joint-distribution matrices Let $J_S\colon\mathcal{X}'\!\!\twoheadrightarrow\!\!\mathcal{Y}_S$ and $J_I\colon\mathcal{X}'\!\!\twoheadrightarrow\!\!\mathcal{Y}_I$ be joint-distribution matrices, both of them *reduced* in the sense of Def. 5, such that $[\![J_{S,I}]\!]=\Delta_{S,I}$ resp. In this section only we use $\mathcal{X}'$ as a reminder that the *input* side of these J's, their row-indices, is actually the *output* side of the *HMM*'s from which they are derived, i.e. that $J_{x',y}=\sum_x H_{x,y,x'}$ as in Def. 9.

We prove the equivalence

$$\Delta_S \sqsubseteq \Delta_I \quad\text{iff}\quad J_S\cdot R = J_I \quad\text{for some } R$$

where R is a stochastic *refinement matrix* of type $\mathcal{Y}_S\!\!\twoheadrightarrow\!\!\mathcal{Y}_I$ (i.e. such that $\sum R_{y,-}=1$ for each $y\colon\mathcal{Y}_S$).

Proof: First we note that for any reduced joint distribution matrix $J\colon\mathcal{Z}\!\!\twoheadrightarrow\!\!\mathcal{Z}'$ there is a one-one correspondence between J's column labels, i.e. elements of $\mathcal{Z}'$, and the support of the hyper $\Delta=[\![J]\!]$ that J defines: it is the function $j\colon\mathcal{Z}'\overset{\text{1-1}}{\longleftrightarrow}\lceil\Delta\rceil$ from Def. 5, injective into $\mathbb{D}\mathcal{Z}$ because J is reduced. We write $(\overset{\text{1-1}}{\longleftrightarrow})$ to emphasise our one-one use of it below.

$\underline{R \text{ makes } \underline{\Delta}}$: We show first that for $J_{S,I},\Delta_{S,I}$ and R as above we can construct a suitable $\underline{\Delta}$. Let the relabelling associated with J_S be $j_S\colon Y_S\overset{\text{1-1}}{\longleftrightarrow}\lceil\Delta_S\rceil$. Relabel Δ_S so that it is a distribution of support $\mathbb{D}Y_S$, so that we can use Def. 5 to define $\underline{\Delta}:=[\![\Delta_S\triangleright R]\!]$, noting that the types of (relabelled) $\Delta_S\in\mathbb{D}\mathcal{Y}_S$ and of $R\in\mathcal{Y}_S\!\!\twoheadrightarrow\!\!\mathcal{Y}_I$ are precisely what Def. 5 requires to produce a result of type $\mathbb{D}^2\mathcal{Y}_S$. Now relabel this (back again) to make an element of $\mathbb{D}^2\lceil\Delta_S\rceil$, that is of $\mathbb{D}^3\mathcal{X}$ because $\lceil\Delta_S\rceil\subseteq\mathbb{D}\mathcal{X}$.

We have $\Delta_S=\mathsf{avg}.\underline{\Delta}$ immediately, from the remark following Def. 6.

For $(\mathbb{D}\mathsf{avg}).\underline{\Delta}=\Delta_I$ we first calculate

$$\begin{array}{ll} & (\mathbb{D}\mathsf{avg}).[\![\Delta_S\triangleright R]\!] \\ = & [\![M\cdot(\Delta_S\triangleright R)]\!]\ , \qquad \text{“Set } \mathcal{D}:=\lceil\Delta_S\rceil \text{ in Lem. 35 below”} \\ & \quad \text{where } M_{x,\rho}:=\rho.x \text{ for } x\colon\mathcal{X} \text{ and } \rho\colon\lceil\Delta_S\rceil. \end{array}$$

Now for arbitrary $x\colon\mathcal{X}$ and $y_I\colon\mathcal{Y}_I$ we continue

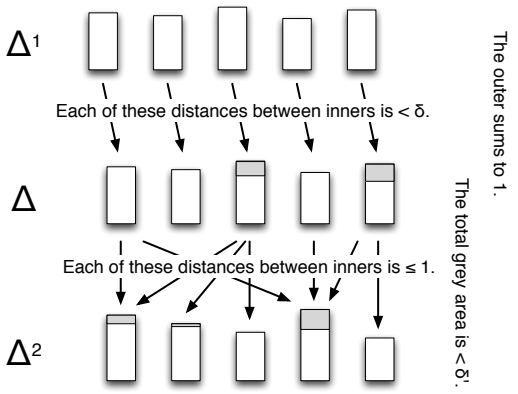
$$\begin{array}{lll} & (M\cdot(\Delta_S\triangleright R))_{x,y_I} & \\ = & \sum_{\rho\colon\mathcal{D}} M_{x,\rho}\,(\Delta_S)_\rho\,R_{\rho,y_I} & \\ = & \sum_{\rho\colon\lceil\Delta_S\rceil} \rho.x\,(\Delta_S)_\rho\,R_{\rho,y_I} & \text{“Defn. } M;\ \mathcal{D}=\lceil\Delta_S\rceil\text{”} \\ = & \sum_{y_S\colon\mathcal{Y}_S} (J_S)_{x,y_S}\,R_{\rho,y_I} & \text{“}\lceil\Delta_S\rceil=\mathcal{Y}_S;\ \Delta_S=[\![J_S]\!]\text{”} \\ = & J_I\ , & \text{“}J_I=J_S\cdot R\text{”} \end{array}$$

whence $(\mathbb{D}\mathsf{avg}).\underline{\Delta}=[\![J_I]\!]=\Delta_I$ as required.

$\underline{\underline{\Delta} \text{ makes } R}$: To show that from $\underline{\Delta}$ we can construct a suitable R, we do similar calculations to the above, but in the reverse direction. □

Lemma 35: **Technical lemma** Let $\mathcal{D}\subseteq\mathbb{D}\mathcal{X}$ be some set of distributions on $\mathcal{X}$, and let $J\colon\mathcal{D}\!\!\twoheadrightarrow\!\!\mathcal{Y}$ be a joint-distribution matrix between (those) *distributions* on $\mathcal{X}$ and some observation space $\mathcal{Y}$. Then $\mathbb{D}\mathsf{avg}.[\![J]\!]=[\![M\cdot J]\!]$, where $M\colon\mathcal{X}\!\!\twoheadrightarrow\!\!\mathcal{D}$ is defined $M_{x,\delta}:=\delta.x$ for $x\colon\mathcal{X}$ and $\delta\colon\mathcal{D}$.

Proof: Arithmetic, and careful index-wrangling. □



We move from Δ^1 to Δ^2 in two steps.
The first-step distance $d_K(\Delta^1,\Delta)$ is no more than the distance between corresponding inners times the total weight moved: the former is $<\delta$ and the latter is exactly 1.
The second-step distance $d_K(\Delta,\Delta^2)$ is no more than the maximum distance between any two inners times the total weight moved: the former is ≤ 1 and the latter is $<\delta'$.

Fig. 7. Illustration of the proof of Lem. 17.

H. Proofs that $[\![H]\!]\in\mathbb{H}\mathcal{X}$ for classical H [§VII-A]

Sometimes called "healthiness conditions", these are essentially technical results giving general properties that are to hold for the denotation of any program. They are used for the proof of other, more specific properties of programs. Here we use them to define a subset $\mathbb{H}\mathcal{X}$ of $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$, and we prove that $[\![H]\!]\in\mathbb{H}\mathcal{X}$ for all classical HMM's H.

Lemma 17: Denotations of HMM's are continuous For all $H\colon\mathcal{X}\to\!\!\triangleright\mathcal{Y}\times\mathcal{X}$ we have that $[\![H]\!]$ is a continuous function in $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ wrt. the Kantorovich metric on both sides.

Proof: We recall from Def. 9 that $[\![H]\!].\pi = [\![\pi\triangleright J]\!]$ where $J_{x',y} = \sum_x H_{x,y,x'}$. We consider this as the composition of the two functions $\pi\mapsto(\pi\triangleright J)$ and $[\![\cdot]\!]$, with the metric on the intermediate space $\mathcal{X}\to\!\!\triangleright\mathcal{Y}$ (of matrices) being the Manhattan metric d_M on $\mathbb{D}(\mathcal{X}\times\mathcal{Y})$. Furthermore, since the first step comprises only elementary arithmetic operations, and the Manhattan metric is topologically equivalent to the Euclidean, the continuity is trivial. Thus we concentrate on the continuity of $[\![\cdot]\!]$ alone.

Within this proof we write $d_M(J^1,J^2)$ for the Manhattan distance between $J^{1,2}\colon\mathcal{X}\to\!\!\triangleright\mathcal{Y}$ and $d_K(\Delta^1,\Delta^2)$ for the Kantorovich distance between $\Delta^{1,2}\colon\mathbb{D}^2\mathcal{X}$. Also, temporarily we use δ as in the ε-and-δ of continuity (i.e. as a real number, not a distribution).

We begin by recalling that nrm of type $\overrightarrow{\mathcal{X}}\to\mathbb{D}\mathcal{X}$ takes (possibly un-normalised) vectors to (normalised) distributions, and that it is continuous. Choose arbitrary $\delta>0$; because $\mathcal{Y}$ is finite, there is a single $\varepsilon>0$ such that for all J^2 we have

$$d_M(J^1,J^2)<\varepsilon \quad\text{implies}\quad d_K(\mathsf{nrm}.J^1_{-,y},\mathsf{nrm}.J^2_{-,y})<\delta \text{ for all } y\colon\mathcal{Y}, \tag{16}$$

since $d_M(J^1_{-,y},J^2_{-,y})<d_M(J^1,J^2)<\varepsilon$ for all $y\colon\mathcal{Y}$.

Similarly for any $\delta'>0$ there is an $\varepsilon'>0$ such that for all J^2 we have

$$d_M(J^1,J^2)<\varepsilon' \quad\text{implies}\quad d_K(Y^1,Y^2)<\delta' , \tag{17}$$

where $Y^{1,2}$ are the $\mathcal{Y}$-marginals $\sum J^{1,2}_{-,y}$ induced by $J^{1,2}$ resp. (In fact $\varepsilon'=\delta'$ suffices.)

Fixing J^1, choose any J^2 such that $d_M(J^1,J^2)<\varepsilon\min\varepsilon'$, and let $\Delta^{1,2}$ be $[\![J^{1,2}]\!]$ resp. We define a hyper Δ that is "one the way" from Δ^1 to Δ^2 as follows: for each y we replace the y-generated inner ρ^1_y of Δ^1 by the corresponding y-generated inner ρ^2_y of Δ^2; but we do not (yet) change the outer probabilities in Δ^1. Thus Δ is a "hybrid", having the inners of Δ^2 but retaining the outer of Δ^1. From (16) we have $d_K(\Delta^1,\Delta)<\delta$.

Now the intermediate Δ has the inners of Δ^2 but still the outer of Δ^1: so we complete the journey from Δ^1 to Δ^2 by changing Δ's outer to Δ^2's outer. By (17) we have $d_K(\Delta,\Delta^2)<\delta'$, since the distance between *any* two inners cannot exceed 1.

So by the triangle inequality for $d_K(\cdot,\cdot)$ we have for arbitrary $\delta,\delta'>0$ that $d_K(\Delta^1,\Delta^2)\leq d_K(\Delta^1,\Delta)+d_K(\Delta,\Delta^2)<\delta+\delta'$ provided $d_M(J^1,J^2)<\varepsilon\min\varepsilon'$, which suffices for continuity of $[\![\cdot]\!]$. □

Lemma 19: Denotations of HMM's are super-linear
For all $H\colon\mathcal{X}\to\!\!\triangleright\mathcal{Y}\times\mathcal{X}$ we have

$$[\![H]\!].\pi_1\ {}_p\!+\![\![H]\!].\pi_2 \quad\sqsubseteq\quad [\![H]\!].(\pi_1\ {}_p\!+\pi_2)\ , \tag{18}$$

where $(\sqsubseteq)$ is refinement as defined in Def. 13.

Proof: Take any reduced $J^{1,2}\colon\mathcal{X}\to\!\!\triangleright\mathcal{Y}$, and argue first that for any $0\leq p\leq 1$ we have

$$[\![J^1]\!]\ {}_p\!+\![\![J^2]\!] \quad\sqsubseteq\quad [\![J^1\ {}_p\!+\ J^2]\!]\ , \tag{19}$$

since the horizontal concatenation J of the two (scaled) matrices $p\times J^1$ and $(1-p)\times J^2$ satisfies $[\![J]\!]=[\![J^1]\!]_p+[\![J^2]\!]$, and J itself is refined to $J_1\ {}_p\!+\ J_2$ (in the sense of Lem. 14) by the refinement matrix

$$R \quad:= \quad \begin{pmatrix} 1 & 0 & \cdots \\ 0 & 1 & \cdots \\ \vdots & \vdots & \vdots \\ 1 & 0 & \cdots \\ 0 & 1 & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{array}{l} \longleftarrow \text{ corresp. to first col. of } J^1 \\ \longleftarrow \text{ corresp. to snd. col. of } J^1 \\ \\ \\ \longleftarrow \text{ corresp. to first col. of } J^2 \\ \longleftarrow \text{ corresp. to snd. col. of } J^2 \\ \\ \end{array}$$

that simply sums corresponding columns.

Now we observe that

$$\begin{array}{ll} & [\![H]\!](\pi_1\ {}_p\!+\ \pi_2) \\ = & [\![(\pi_1\ {}_p\!+\ \pi_2)\triangleright H)]\!] \\ = & [\![\pi_1\triangleright H\ {}_p\!+\ \pi_2\triangleright H]\!] \\ \sqsupseteq & [\![\pi_1\triangleright H]\!]\ {}_p\!+\ [\![\pi_2\triangleright H]\!] \qquad \text{“(19) just above”} \\ = & [\![H]\!].\pi_1\ {}_p\!+\ [\![H]\!].\pi_2\ , \end{array}$$

as required. □

I. Properties of the refinement order ($\sqsubseteq$)

1) Abstract HMM's are ($\sqsubseteq$)*-monotonic:* [§VII-A]
Super-linearity (Lem. 19) is equivalently ($\sqsubseteq$)-monotonicity of the Kleisli-extension $h^\dagger$ of any $h\colon \mathbb{H}\mathcal{X}$; that is, it is equivalent to the more general $\Delta_1 \sqsubseteq \Delta_2 \Rightarrow h^\dagger.\Delta_1 \sqsubseteq h^\dagger.\Delta_2$. Assuming ($\sqsubseteq$)-monotonicity and recalling that $[\cdot]$ is the point distribution, we have trivially $[\pi_1]_p{+}[\pi_2] \sqsubseteq [\pi_{1p}{+}\pi_2]$ and so

$$
\begin{array}{lll}
 & h.\pi_{1\,p}{+}\,h.\pi_2 & \\
= & h^\dagger.[\pi_1]_{\,p}{+}\,h^\dagger.[\pi_2] & \text{“defn. } h^\dagger\text{”} \\
= & h^\dagger.([\pi_1]_p{+}[\pi_2]) & \text{“}h^\dagger \text{ linear”} \\
\sqsubseteq & h^\dagger.[\pi_{1p}{+}\pi_2] & \text{“}[\pi_1]_p{+}[\pi_1] \sqsubseteq [\pi_{1p}{+}\pi_2]\text{;} \\
 & & \text{assumption that } h^\dagger \text{ is monotonic”} \\
= & h.(\pi_{1p}{+}\pi_2)\ . & \text{“defn. } h^\dagger\text{”}
\end{array}
$$

For the other direction (sketch), in the discrete case we note that a proof of $\Delta_1 \sqsubseteq \Delta_2$ can be broken down into a succession of column-merges (in the matrix representation), each of them being of the form “replace $[\pi_1]_p{+}[\pi_2]$ by $[\pi_{1p}{+}\pi_2]$”.

2) Composition of abstract HMM's respects the refinement order: [§VII-A]
We show that sequential composition of abstract *HMM*'s respects the refinement order ($\sqsubseteq$) on both sides, i.e. that for $h, h_{1,2}\colon \mathbb{H}\mathcal{X}$ we have both

$$h_1 \sqsubseteq h \quad\Rightarrow\quad h_1; h_2 \sqsubseteq h; h_2 \tag{20}$$

$$\text{and} \qquad h_2 \sqsubseteq h \quad\Rightarrow\quad h_1; h_2 \sqsubseteq h_1; h\ . \tag{21}$$

Although this can be argued directly in terms of abstract *HMM*'s, it is easier if we use the *UM*'s defined later (§VIII). For (20) we have

$$
\begin{array}{ll}
 & h_1 \sqsubseteq h \\
\text{iff} & \text{wp}.h_1 \leq \text{wp}.h \qquad \text{“App. I3 just below”} \\
\text{implies} & \\
 & \text{wp}.h_1 \circ \text{wp}.h_2 \ \leq\ \text{wp}.h \circ \text{wp}.h_2 \\
\text{iff} & \text{wp}.(h_1; h_2) \leq \text{wp}.(h_1; h) \qquad \text{“Cor. 30 in App. O”} \\
\text{iff} & h_1; h_2 \sqsubseteq h_1; h\ . \qquad \text{“App. I3”}
\end{array}
$$

And for (21) we have

$$
\begin{array}{ll}
 & h_2 \sqsubseteq h \\
\text{iff} & \text{wp}.h_2 \leq \text{wp}.h \qquad \text{“App. I3 just below”} \\
 & \\
\text{implies} & \qquad \text{“wp}.h_1 \text{ is } (\leq)\text{-monotonic, Lem. 26(2)”} \\
 & \text{wp}.h_1 \circ \text{wp}.h_2 \ \leq\ \text{wp}.h_1 \circ \text{wp}.h \\
 & \\
\text{iff} & \text{wp}.(h_1; h_2) \leq \text{wp}.(h_1; h) \qquad \text{“Cor. 30 in App. O”} \\
\text{iff} & h_1; h_2 \sqsubseteq h_1; h\ . \qquad \text{“App. I3”}
\end{array}
$$

3) Refinement of transformers: [§VIII-D]
Here we prove the correspondence between the forwards- and the backwards manifestations of refinement ($\sqsubseteq$), i.e. that we have

$$h_1 \sqsubseteq h_2 \quad \text{iff} \quad \text{wp}.h_1 \leq \text{wp}.h_2\ ,$$

where on the *rhs* we have extended ($\leq$) pointwise, i.e. meaning $\text{wp}.h_1.u.\pi \leq \text{wp}.h_1.u.\pi$ for all $u\colon \mathbb{U}\mathcal{X}$ and $\pi\colon \mathbb{D}\mathcal{X}$. We reason

$$
\begin{array}{ll}
 & h_1 \sqsubseteq h_2 \\
\text{iff} & \qquad \text{“pointwise extension } (\sqsubseteq)\text{”} \\
 & h_1.\pi \sqsubseteq h_2.\pi \quad \text{for all } \pi\colon \mathbb{D}\mathcal{X} \\
 & \\
\text{iff} & \qquad \text{“Lem. 23, soundness and completeness”} \\
 & \mathcal{E}_{h_1.\pi}\, u \leq \mathcal{E}_{h_2.\pi}\, u \qquad \text{for all } \pi\colon \mathbb{D}\mathcal{X} \text{ and all } u\colon \mathbb{U}\mathcal{X} \\
 & \\
\text{iff} & \qquad \text{“defn. wp.()”} \\
 & \text{wp}.h_1.u.\pi \leq \text{wp}.h_2.u.\pi \qquad \text{for all } \pi\colon \mathbb{D}\mathcal{X} \\
 & \qquad\qquad\qquad\qquad\ \ \text{and for all } u\colon \mathbb{U}\mathcal{X} \\
 & \\
\text{iff} & \text{wp}.h_1 \leq \text{wp}.h_2\ . \qquad \text{“pointwise extension”}
\end{array}
$$

4) Composition respects transformer refinement: [§VIII-D]
For $t_{1,2}\colon \mathbb{U}\mathcal{X}$ we have defined $t_1 \sqsubseteq t_2$ to be simply that $t_1.u \leq t_2.u$ for all $u\colon \mathbb{U}\mathcal{X}$. Here we show that functional composition of transformers respects that refinement order ($\sqsubseteq$) on both sides, i.e. that for $t, t_{1,2}\colon \mathbb{T}\mathcal{X}$ we have both $t_1 \sqsubseteq t \Rightarrow t_1 \circ t_2 \sqsubseteq t \circ t_2$ and $t_2 \sqsubseteq t \Rightarrow t_1 \circ t_2 \sqsubseteq t_1 \circ t$.

In fact it is trivial from the property (imposed by $\mathbb{T}\mathcal{X}$) that transformers are ($\leq$)-monotonic, that is Lem. 26(2).

J. Soundness and completeness: Lem. 23 [§VIII]

We mention soundness and completeness in this paper because it provides an important justification for our definition and use of the general uncertainty measures and, in particular, their transformers.

The soundness part of Lem. 23 is related to the Data-Processing Inequality, the *DPI* [33] , which concerns two channels $C\colon \mathcal{X}\to\!\!\!\!\triangleright\,\mathcal{Y}$ and $R\colon \mathcal{Y}\to\!\!\!\!\triangleright\,\mathcal{Z}$. (Note that the channel R here takes the *observations* $\mathcal{Y}$ of Channel C as its input. Our *HMM*'s do not take observations as input.)

Informally stated, the *cascade* of C and R is the channel given by the matrix multiplication $C{\cdot}R$, and the *DPI* states that the information leakage from $C{\cdot}R$ cannot be more than the leakage from C alone: adding another child to the game "Chinese Whispers" cannot make the eventual ouput less ridiculous.

We call this *soundness* because it states that a no-less-secure hyper wrt. our uncertainty refinement order indeed cannot be less uncertain when tested with *any* uncertainty measure. This result is proved in in [6], [8].

The completeness part of Lem. 23 is related to the "Coriaceous Conjecture" partially proved in [8], which became the Coriaceous Property (*CP*) in [6] when its proof, for channels, was presented in complete form based on its earlier, complete proof in [4] for hypers.[37] In [6] terms, the *CP* is that if there *is no* R such that $C_1{\cdot}R{=}C_2$ then there is a "gain function" (for us here, a loss function, which determines a special form of uncertainty measure in our terms) that is witness to the non-existence of such an R. The importance of the *CP* for quantitative information-flow security was explained in [8], and it was proved there to hold for many interesting special cases of $C_{1,2}$. But not for all of them.

The *CP* was proved to extend beyond the discrete case, to proper measure spaces, in [5].

[37] Geoffrey Smith has since told us that it follows from a result of Blackwell [34].

K. Proof of Lem. 25 [§VIII-B]

This technical lemma assures the well definedness of our dual space: we have defined our uncertainty measures $\mathbb{U}\mathcal{X}$ as functions in $\mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$ with certain properties; and we have stated that $\mathrm{wp}.h.u$ is also an uncertainty measure. Thus we must show that $\mathrm{wp}.h.u \in \mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$ and that it satisfies the properties for membership of $\mathbb{U}\mathcal{X}$.

Lemma 25: Well-definedness of Def. 24
If $h\colon \mathbb{H}\mathcal{X}$ is an abstract *HMM* and $u\colon \mathbb{U}\mathcal{X}$ is a *UM*, then $\mathrm{wp}.h.u$ is in $\mathbb{U}\mathcal{X}$.

Proof: Since $h{\in}\mathbb{H}\mathcal{X}$, we have $h\colon \mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ satisfying Lems. 17,19. We must show for any $u\colon \mathbb{U}\mathcal{X}$ and $\delta\colon \mathbb{D}\mathcal{X}$ that $\delta{\mapsto}\mathcal{E}_{h.\delta}\,u$ is in $\mathbb{U}\mathcal{X}$, i.e. that it is in $\mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$, is concave and is continuous (Def. 22).

Membership of $\delta{\mapsto}\mathcal{E}_{h.\delta}\,u$ in $\mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$ is trivial.

For concavity: Because $u{\in}\mathbb{U}\mathcal{X}$ we know it is itself concave; and we have that h satisfies the properties in Def. 20. We now reason

$$
\begin{array}{lll}
 & \mathrm{wp}.h.u.(\pi_1\ {}_p{+}\ \pi_2) & \\
= & \mathcal{E}_{h.(\pi_1 {}_p{+} \pi_2)}\,u & \text{“Def. 24”} \\
 & & \\
\geq & & \text{“ Def. 20, hence } (h.\pi_1)_p{+}(h.\pi_2) \sqsubseteq h.(\pi_1{}_p{+}\pi_2) \\
 & & \quad u \text{ concave, hence Lem. 16 (non-strict) applies ”} \\
 & \mathcal{E}_{(h.\pi_1)_p+(h.\pi_2)}\,u & \\
 & & \\
= & \mathcal{E}_{h.\pi_1}\,u\ {}_p{+}\ \mathcal{E}_{h.\pi_2}\,u & \text{“}\mathcal{E}\text{ is linear”} \\
= & \mathrm{wp}.h.u.\pi_1\ {}_p{+}\ \mathrm{wp}.h.u.\pi_2\ , & \text{“Def. 24”}
\end{array}
$$

as required.

For continuity: We must show that $\mathrm{wp}.h.u$ is continuous, given that both u, h are themselves continuous. Because h itself is continuous, we need only show that in general the function $\Delta{\mapsto}\mathcal{E}_\Delta\,u$ is continuous wrt. Kantorovich on the left, in Δ for fixed continuous u. This is proved in Lem. 37 just below. It in turn requires several supporting theorems, given in App. L immediately following. □

We now prove the following lemma showing that $\Delta{\mapsto}\mathcal{E}_\Delta\,u$ is continuous whenever u is a continuous function. That is, it holds for a more general class of functions rather than just for uncertainty measures $u\colon \mathbb{U}\mathcal{X}$.

Definition 36: **Space of continuous functions** We define $\mathbb{C}\mathcal{X}$ to be the set of all *continuous functions* from $\mathbb{D}\mathcal{X}$ (with the Kantorovich metric) to $\mathbb{R}$ (with the ordinary metric). This set is endowed with the *uniform metric* $\|\cdot - \cdot\|_\infty$, defined

$$\|u_1{-}u_2\|_\infty \quad := \quad \sup_{\delta\colon \mathbb{D}\mathcal{X}} |u_1.\delta - u_2.\delta| \ , \tag{22}$$

that turns $\mathbb{C}\mathcal{X}$ into a complete metric space. □
In the sequel, we write $\|u\|_\infty{:=}\|u{-}\mathbf{0}\|_\infty$.

Lemma 37: **Continuity lemma** Let $u\colon \mathbb{D}\mathcal{X}\to\mathbb{R}^{\geq}$ be continuous (wrt. the Kantorovich metric on $\mathbb{D}\mathcal{X}$). Then the function $F.\Delta{:=}\mathcal{E}_\Delta\,u$ is also continuous.

Proof: Let $\Delta, \Delta'\colon \mathbb{D}^2\mathcal{X}$ be two hypers. We make first the stronger assumption that u is a k-Lipschitz function, for some $k{>}0$; and to establish the connection with the conventional

presentation of the Kantorovich metric we write in this section only $\int \cdots \mathrm{d}\Delta$ rather than $\mathcal{E}_\Delta \cdots$.

We have

$$\begin{array}{lll} & \left|\int u\,\mathrm{d}\Delta - \int u\,\mathrm{d}\Delta'\right| & \\ = & k\left|\int {}^1\!/_k\, u\,\mathrm{d}\Delta - \int {}^1\!/_k\, u\,\mathrm{d}\Delta'\right| & \text{“}{}^1\!/_k f \text{ is 1-Lipschitz”} \\ \leq & k\, d_K(\Delta,\Delta')\ , & \text{“Definition of } d_K(\cdot,\cdot)\text{”} \end{array}$$

and hence also the map F is k-Lipschitz in this case.

Now we relax our assumption, letting u be an arbitrary continuous function in $\mathbb{C}\mathcal{X}$, i.e. no longer necessarily 1-Lipschitz. By the density of Lipschitz functions Thm. 42, there exists a sequence u_n of Lipschitz functions such that

$$\lim_{n\to\infty} \|u - u_n\|_\infty \quad = \quad 0\ . \tag{23}$$

That sequence u_n generates a sequence $F_n : \mathbb{D}^2\mathcal{X}\to\mathbb{R}^{\geq}$ defined by $F_n.\Delta = \int u_n \mathrm{d}\Delta$. Each function F_n is continuous because u_n is k-Lipschitz for some k (depending on n).

Now for every $n\colon\mathbb{N}$ and $\pi\colon\mathbb{D}\mathcal{X}$ we have that $|u_n.\pi| \leq \sup_n \|u_n\|_\infty$, where $\sup_n \|u_n\|_\infty$ is finite because

- $\|u_n\|_\infty$ converges to $\|u\|_\infty$ (continuous image of a convergent sequence),
- and so there exists $N\colon\mathbb{N}$ such that $|\|u_k\|_\infty - \|u\|_\infty| \leq 1$ for every $k{\geq}N$. That is, we have $\sup_n \|u_n\|_\infty \leq \max(\max_{n\leq N} \|u_n\|_\infty, \|u\|_\infty + 1)$.

Therefore, by the Dominated Convergence Theorem [35, Thm. 5.6], we know that u is Δ-integrable and that the sequence F_n converges to F such that $F.\Delta = \int u\,\mathrm{d}\Delta$ for every $\Delta \in \mathbb{D}^2 S$ (and this convergence is pointwise). Let us show that F is necessarily continuous by showing that F_n converges uniformly to F.

Let $\varepsilon > 0$. For every $\Delta\colon\mathbb{D}^2\mathcal{X}$, we have

$$\begin{array}{lll} & |F_n.\Delta - F.\Delta| & \\ = & \left|\int u_n\,\mathrm{d}\Delta - \int u\,\mathrm{d}\Delta\right| & \text{“Definition of } F_n \text{ and } F\text{”} \\ \leq & \int |u_n{-}u|\,\mathrm{d}\Delta & \text{“}\left|\int g\mathrm{d}\Delta\right| \leq \int |g|\mathrm{d}\Delta\text{”} \\ \leq & \|u_n{-}u\|_\infty\ . & \text{“}|u_n{-}u| \leq \|u_n{-}u\|_\infty \text{ and } \int \mathbf{1}\mathrm{d}\Delta = 1\text{”} \end{array}$$

Since $\lim_{n\to\infty} \|u_n{-}u\|_\infty = 0$, there exists $N\colon\mathbb{N}$ (that depends on ε only) such that $|F_n.\Delta{-}F.\Delta| < \varepsilon$ for every $n{\geq}N$. By the Uniform Limit Theorem [36, Thm. 21.6], the uniform limit of a sequence of continuous functions is continuous: that is F is continuous, which is what we had to prove. □

Remarkably, it is now easy to show that $\mathrm{wp.}()$ is an injection over all of (measurable) $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$, a fact we will use later (in App. O); we have

Lemma 38: **$\mathrm{wp.}()$ is an injection on $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$**

If $\mathrm{wp.}h_1{=}\mathrm{wp.}h_2$ for some (measurable) $h_{1,2}\colon\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$, then $h_1{=}h_2$.

Proof: We reason

$$\begin{array}{lll} & h_1 \neq h_2 & \\ \Rightarrow & h_1.\pi \neq h_2.\pi & \text{“for some } \pi\colon\mathbb{D}\mathcal{X}\text{”} \\ \Rightarrow & h_1.\pi \not\sqsubseteq h_2.\pi & \text{“}\textit{wlog}\text{; } (\sqsubseteq)\text{-antisymmetry from §V”} \\ \Rightarrow & \mathcal{E}_{h_2.\pi}\, u < \mathcal{E}_{h_1.\pi}\, u & \text{“Lem. 23 completeness (}\textit{Coriaceous}\text{), for some } u\colon\mathbb{U}\mathcal{X}\text{”} \\ \Leftrightarrow & \mathrm{wp.}h_2.u.\pi < \mathrm{wp.}h_1.u.\pi & \text{“Def. 24”} \\ \Rightarrow & \mathrm{wp.}h_1 \neq \mathrm{wp.}h_2\ . & \end{array}$$

L. Lipschitz Density Theorem [Lem. 37 in §K]

This section shows that the set $\mathbb{L}\mathcal{X}$ of Lipschitz functions from $\mathbb{D}\mathcal{X}$ to $\mathbb{R}$ is dense in $\mathbb{C}\mathcal{X}$ wrt. the uniform metric $\|\cdot-\cdot\|_\infty$. The proof relies on the Stone-Weierstrass Density Theorem [37, Thm. 5], repeated here:

Theorem 39: **Stone-Weierstrass Density**
Given the compact metric space $\mathbb{D}\mathcal{X}$ and a subalgebra $\mathcal{A}$ of $\mathbb{C}\mathcal{X}$, the set $\mathcal{A}$ is dense in $\mathbb{C}\mathcal{X}$ (wrt. $\|\cdot-\cdot\|_\infty$) if and only if it separates distributions in $\mathbb{D}\mathcal{X}$ and vanishes nowhere. □

Lemma 40: **Sub-algebra** $\mathbb{L}\mathcal{X}$ is a subalgebra of $\mathbb{C}\mathcal{X}$.

Proof: The constant functions $\mathbf{0}, \mathbf{1}$ are in $\mathbb{L}\mathcal{X}$. Let $f, g\colon \mathbb{L}\mathcal{X}$ and $c\colon \mathbb{R}$. We will show that $cf, f+g$, fg and f/g (where $g > 0$) are in $\mathbb{L}\mathcal{X}$ (where the operations are defined pointwise).

Scaling: We have

$$|c(f.\delta)-c(f.\delta')| \;=\; |c||f.\delta-f.\delta'| \;\leq\; |c|n(d_K(\delta,\delta'))\ ,$$

for some $n\colon \mathbb{R}$.

Addition: We have

$$|(f+g).\delta-(f+g).\delta'| \;\leq\; |f.\delta-f.\delta'|+|g.\delta-g.\delta'| \leq (n+m)d_K(\delta,\delta')\ ,$$

for some $m, n\colon \mathbb{R}$.

Multiplication: We have

$$\begin{array}{lll} & |fg.\delta - fg.\delta'| & \\ = & |f.\delta g.\delta - f.\delta g.\delta' + f.\delta g.\delta' - f.\delta' g.\delta'| & \\ \leq & |f.\delta||g.\delta - g.\delta'| + |g.\delta'||f.\delta - f.\delta'| & \text{“}|x+y| \leq |x| + |y|\text{”} \\ \leq & \leq (\|f\|_\infty m + \|g\|_\infty n) d_K(\delta,\delta') & \text{“}|f.\delta| \leq \|f\|_\infty\text{”} \end{array}$$

where $\|f\|_\infty$ and $\|g\|_\infty$ are finite because f, g are continuous on the compact metric space $\mathbb{D}\mathcal{X}$. □

Lemma 41: **Point separation** $\mathbb{L}\mathcal{X}$ separates distributions in $\mathbb{D}\mathcal{X}$ and vanishes nowhere.

Proof: That $\mathbb{L}\mathcal{X}$ vanishes nowhere is clear since $\mathbf{1}\in\mathbb{L}\mathcal{X}$. Let us prove that $\mathbb{L}\mathcal{X}$ separates points. Let $\delta_1, \delta_2\colon \mathbb{D}\mathcal{X}$ such that $\delta_1 \neq \delta_2$. We consider the function $f\colon \mathbb{D}\mathcal{X} \to \mathbb{R}$ such that $f.\delta := d_K(\delta, \delta_1)$. The function f is indeed 1-Lipschitz because, by the triangular inequality, we have

$$|f.\delta - f.\delta'| \;=\; |d_K(\delta,\delta_1) - d_K(\delta',\delta_1)| \;\leq\; d_K(\delta,\delta')$$

and $f.\delta_2 > 0 = f.\delta_1$ because $\delta_2 \neq \delta_1$ and $d_K(\cdot,\cdot)$ is a metric. □

Theorem 42: **Lipschitz Density** $\mathbb{L}\mathcal{X}$ is dense in $\mathbb{C}\mathcal{X}$.

Proof: This follows from Lem. 41, Lem. 40 and the Stone-Weierstrass Thm. 39. □

M. Proof of Lem. 27 [§VIII-C]

Lemma 27: Uncertainty transformers are 1-Lipschitz Take $h\colon \mathbb{H}\mathcal{X}$ and define $t := \mathrm{wp}.h$. Let $|\cdot|$ be absolute value. Then t is 1-Lipschitz in the sense that

$$\sup_{\delta\colon \mathbb{D}\mathcal{X}} |t.u_1.\delta - t.u_2.\delta| \quad\leq\quad \sup_{\delta\colon \mathbb{D}\mathcal{X}} |u_1.\delta - u_2.\delta|\ .$$

Proof: Consider arbitrary $u_{1,2}\colon \mathbb{U}\mathcal{X}$. We reason

$$\begin{array}{lll} & \sup_{\delta\colon \mathbb{D}\mathcal{X}} |\mathrm{wp}.h.u_1.\delta - \mathrm{wp}.h.u_2.\delta| & \\ = & \sup_{\delta\colon \mathbb{D}\mathcal{X}} |\mathcal{E}_{h.\delta}\, u_1 - \mathcal{E}_{h.\delta}\, u_2| & \\ \leq & \sup_{\delta\colon \mathbb{D}\mathcal{X}}\ \mathcal{E}_{h.\delta}\, |u_1 - u_2| & \text{“property of } |\cdot|\text{”} \\ \leq & \sup_{\delta\colon \mathbb{D}\mathcal{X}} & \text{“}\mathcal{E} \text{ monotonic”} \\ & \quad \mathcal{E}_{h.\delta}\, (\sup_{\delta'\colon \mathbb{D}\mathcal{X}} |u_1.\delta' - u_2.\delta'|) & \\ = & \sup_{\delta\colon \mathbb{D}\mathcal{X}} |u_1.\delta - u_2.\delta|\ . & \text{“rename } \delta' \text{ to } \delta\text{”} \end{array}$$

□

N. Proof of Thm. 29 [§VIII-D]

1) Overview:

The proof first establishes a non-trivial version of the Riesz representation for the current setting, with the main argument showing that our definitions extend more generally (i.e. outside of $\mathbb{U}\mathcal{X}$) thus allowing appeal to established mathematical results on dualities. The constructed h is then shown to have the characteristic properties set out at Lem. 17 and Lem. 19.

Theorem 29: Characterisation of transformers
Take $t\colon \mathbb{T}\mathcal{X}$. That is, t is

1) *linear*: for $a_{1,2}\colon \mathbb{R}^{\geq}$ and $u_{1,2}\colon \mathbb{U}\mathcal{X}$ we have
$$t.(a_1u_1 + a_2u_2) \quad = \quad a_1t.u_1 + a_2t.u_2 \ ,$$
2) *monotonic*: $t.u_1.\delta \geq t.u_2.\delta$ for every $u_1 \geq u_2$, where $u_{1,2}\colon \mathbb{U}\mathcal{X}$ and $\delta\colon \mathbb{D}\mathcal{X}$. (We use pointwise lifting of $\geq$ on $\mathbb{U}\mathcal{X}$.)
3) *total*: $t.\mathbf{1}{=}\mathbf{1}$ where $\mathbf{1}.\delta{:=}1$ for all $\delta\colon \mathbb{D}\mathcal{X}$,
4) *1-Lipschitz* on $\mathbb{U}\mathcal{X}$ wrt. the uniform metric $\|-\cdot-\|_\infty$ as defined by (22) (and used e.g. in Lem. 27).

Then there is a unique function $h\colon \mathbb{H}\mathcal{X}$ such that $t = \mathrm{wp}.h$.

Proof: We give the proof in App. N3, after some technical lemmas. □

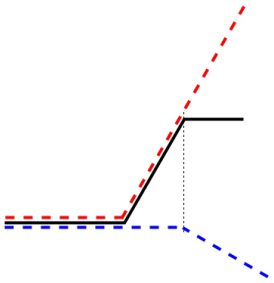
2) Technical preparation:
The core ingredient in the proof of this theorem is the Riesz Representation Theorem for linear functionals (linear maps from a normed vector space to $\mathbb{R}$). A difficulty however originates from the fact that the representation theorem is stated on the space of all continuous functions $\mathbb{C}\mathcal{X}$, but our linear function t is defined only from $\mathbb{U}\mathcal{X}$ to itself.

Yet $\mathbb{U}\mathcal{X}$ is a sub-metric space of $\mathbb{C}\mathcal{X}$ under the uniform metric $\|-\cdot-\|_\infty$. More importantly, we prove that the vector space generated by $\mathbb{U}\mathcal{X}$ is dense in $\mathbb{C}\mathcal{X}$. (See Lem. 43 and Fig. 8.) This is essential to ensure that if t extends to a continuous linear function over $\mathbb{C}\mathcal{X}$, then such an extension is necessarily unique. We will show in Thm. 45 that such an extension always exists.

Lemma 43: **Concave density** The vector space generated by $\mathbb{U}\mathcal{X}$ is dense in $\mathbb{C}\mathcal{X}$ wrt. $\|-\cdot-\|_\infty$.

Proof: This result essentially follows from [38, Pro. 2.2]. We give the proof here for completeness.

Let $\langle\mathbb{U}\mathcal{X}\rangle$ be the set of functions that can be written as the difference of two positive concave functions from $\mathbb{D}\mathcal{X}$ to $\mathbb{R}$. Then $\langle\mathbb{U}\mathcal{X}\rangle$ coincides with the real vector space generated by $\mathbb{U}\mathcal{X}$ (by grouping positively and negatively weighted components). Equivalently, every function in $\langle\mathbb{U}\mathcal{X}\rangle$ is the difference of two positive continuous convex functions: if $f = u_1 - u_2$ for $u_{1,2}\colon \mathbb{U}\mathcal{X}$, then
$$f = (-u_2 - c) - (-u_1 - c)$$
where $c = \min(\inf_{\delta\colon \mathbb{D}\mathcal{X}} -u_1.\delta, \inf_{\delta\colon \mathbb{D}\mathcal{X}} -u_2.\delta)$. The constant c is finite because $\mathbb{D}\mathcal{X}$ is compact. The functions $-u_{1,2} - c$ are positive, continuous and convex functions.



Every continuous piecewise-linear function can be expressed as the sum of finitely many convex and concave functions using the construction shown in this figure. This provides a geometrical view of the Concave Density Lem. 43.

Fig. 8. The sum of convex (red, upper) and concave (blue, lower) functions gives a zig-zag (black, middle).

Now let us apply the Stone-Weierstrass Thm. 39 on $\langle\mathbb{U}\mathcal{X}\rangle$ which is a subset of $\mathbb{C}\mathcal{X}$.

To do that we need first to show that $\langle\mathbb{U}\mathcal{X}\rangle$ is an algebra (i.e. has a zero and unit, is closed under scalar multiplication and addition and pointwise multiplication of f's). In addition $\langle\mathbb{U}\mathcal{X}\rangle$ must "vanish nowhere" on $\mathbb{D}\mathcal{X}$ and "separate points". (See below for explanations of those properties.)

$\langle\mathbb{U}\mathcal{X}\rangle$ is an algebra: Since $\langle\mathbb{U}\mathcal{X}\rangle$ is a vector space, the constant functions $\mathbf{0}, \mathbf{1}$ (identically 0 and 1 resp.) and the functions $cf, f+g$ are in $\langle\mathbb{U}\mathcal{X}\rangle$ for every $c\colon \mathbb{R}$ and $f, g\colon \langle\mathbb{U}\mathcal{X}\rangle$.

Let $f, g\colon \langle\mathbb{U}\mathcal{X}\rangle$ be such that $f = u_1{-}u_2$ and $g = v_1{-}v_2$ where $u_{1,2}, v_{1,2}$ are positive continuous convex functions. Notice that
$$f^2 \quad = \quad 2(u_1^2 + u_2^2) - (u_1 + u_2)^2$$
where $u_{1,2}^2$ and $(u_1{+}u_2)^2$ are positive convex functions (because the square of a non-negative convex function is convex). That is, we have $f^2 \in \langle\mathbb{U}\mathcal{X}\rangle$. Now
$$fg \quad = \quad (f+g)^2 - (f^2 + g^2)$$
and thus $fg{\in}\langle\mathbb{U}\mathcal{X}\rangle$, because we have just shown that all of $(f+g)^2, f^2, g^2$ are in $\langle\mathbb{U}\mathcal{X}\rangle$.

$\langle\mathbb{U}\mathcal{X}\rangle$ vanishes nowhere: We must show that for each $\delta\colon \mathbb{D}\mathcal{X}$ there is some $f\colon \langle\mathbb{U}\mathcal{X}\rangle$ such that $f\delta \neq 0$. But this is immediate since $\mathbf{1}.\delta \neq 0$ for every $\delta\colon \mathbb{D}\mathcal{X}$ and $\mathbf{1}{\in}\langle\mathbb{U}\mathcal{X}\rangle$.

$\langle\mathbb{U}\mathcal{X}\rangle$ separates points: We must show that for every pair $\delta \neq \delta'$ there is some $f \in \langle\mathbb{U}\mathcal{X}\rangle$ such that $f.\delta \neq f.\delta'$. We argue as follows.

Given $\delta\colon \mathbb{D}\mathcal{X}$ (fixed) and define $f^\delta.\delta' := d_K(\delta', \delta)$ for $\delta'\colon \mathbb{D}\mathcal{X}$. Observe that for $\delta' \neq \delta$ we have that $0 = f^\delta.\delta < d_K(\delta', \delta) = f^\delta.\delta'$. Thus it suffices to show that $f^\delta{\in}\langle\mathbb{U}\mathcal{X}\rangle$, and we continue as follows.

For every $\delta_{1,2}\colon \mathbb{D}\mathcal{X}$, we have
$f^\delta.(\delta_{1p}{+}\delta_2)$

$=$ "d_K is Kantorovich distance; Definition of f^δ" $d_K((\delta_1{}_p{+}\delta_2),\delta)$

$\leq$ $d_K(\delta_1,\delta)_p{+}d_K(\delta_2,\delta)$ "$d_K(\delta',\delta)=\frac{1}{2}d_M(\delta',\delta)$ is convex, for fixed δ"

$=$ $f^\delta.\delta_1{}_p{+}f^\delta.\delta_2$ "Definition of f^δ"

That is $\mathbf{1}-f^\delta\in\mathbb{U}\mathcal{X}$ and so, by Thm. 39, we have $\langle\mathbb{U}\mathcal{X}\rangle$ dense in $\mathbb{C}\mathcal{X}$. □

The extension of a transformer $t{:}\,\mathbb{U}\mathcal{X}{\to}\mathbb{U}\mathcal{X}$ to a continuous linear function from $\mathbb{C}\mathcal{X}$ to itself is done in two stages. Firstly, t is extended *linearly* to a continuous linear function $t'{:}\,\langle\mathbb{U}\mathcal{X}\rangle{\to}\mathbb{C}\mathcal{X}$. This step is justified in Thm. 45. Secondly, t' is extended *continuously* to a continuous linear function $\tilde{t}{:}\,\mathbb{C}\mathcal{X}{\to}\mathbb{C}\mathcal{X}$. This step uses the density proven in Lem. 43 and is shown in Lem. 44 below.

Lemma 44: **Extension from $\langle\mathbb{U}\boldsymbol{\mathcal{X}}\rangle$ to $\mathbb{C}\boldsymbol{\mathcal{X}}$** Every continuous linear function t from $\langle\mathbb{U}\mathcal{X}\rangle$ to $\mathbb{C}\mathcal{X}$ extends uniquely to a continuous linear function $\tilde{t}$ from $\mathbb{C}\mathcal{X}$ to itself.

Proof: This result follows from the Continuous Linear Extension Theorem [39, Ch. 4 Thm. 10.1].

All we need to show is that the (Cauchy) completion of $\langle\mathbb{U}\mathcal{X}\rangle$ is $\mathbb{C}\mathcal{X}$, which follows from the fact that $\langle\mathbb{U}\mathcal{X}\rangle$ is dense in $\mathbb{C}\mathcal{X}$ (Lem. 43) and that $\mathbb{C}\mathcal{X}$ is a complete normed vector space when endowed with the uniform norm $\|f\|_\infty{:=}\|f{-}\mathbf{0}\|_\infty$. □

Theorem 45: **Extension from $\mathbb{U}\boldsymbol{\mathcal{X}}$ to $\mathbb{C}\boldsymbol{\mathcal{X}}$** Every transformer extends uniquely to a positive continuous linear function from $\mathbb{C}\mathcal{X}$ to itself.

Proof: Let $t{:}\,\mathbb{T}\mathcal{X}$ be a transformer. It suffices to prove that t has a positive continuous extension t' on the sub-vector space $\langle\mathbb{U}\mathcal{X}\rangle$. If such a t' exists then a unique extension $\tilde{t}{:}\,\mathbb{C}\mathcal{X}{\to}\mathbb{C}\mathcal{X}$, which is positive [38] and continuous, can be deduced using Lem. 44.

Let $f{:}\,\langle\mathbb{U}\mathcal{X}\rangle$, there exists $u_{1,2}\in\mathbb{U}\mathcal{X}$ such that $f{=}u_1{-}u_2$. We define $t'.f{=}t.u_1-t.u_2$.

t' is well-defined: We must show that $t'.f$ is independent of how f is written as the difference of two uncertainty measures. Firstly, notice that if $u_1{-}u_2\in\mathbb{U}\mathcal{X}$ for some $u_{1,2}\in\mathbb{U}\mathcal{X}$ then $t.(u_1{-}u_2)=t.u_1-t.u_2$. Secondly, let $f=u_1{-}u_2=v_1{-}v_2$. Then $(u_1{+}v_2){-}(u_2{+}v_1)=\mathbf{0}$, which is in $\mathbb{U}\mathcal{X}$. Therefore, we have $t.(u_1+v_2)-t.(u_2+v_1)=\mathbf{0}$, and that implies $t.u_1-t.u_2=t.v_1-t.v_2$ by linearity of t.

t' is linear and unique: Linearity is clear and it implies uniqueness of the extension t' over $\langle\mathbb{U}\mathcal{X}\rangle$.

t' is 1-Lipschitz: Let $f,g{:}\,\langle\mathbb{U}\mathcal{X}\rangle$ be such that we have $f=u_1{-}u_2$ and $g=v_1{-}v_2$. Then

$\|t'.f{-}t'.g\|_\infty$
$=$ $\|(t.u_1-t.u_2){-}(t.v_1-t.v_2)\|_\infty$ "Definition of t'"
$=$ $\|t.(u_1+v_2){-}t.(v_1+u_2)\|_\infty$ "t is linear, $u_i+v_j{\in}\mathbb{U}\mathcal{X}$"
$\leq$ $\|(u_1+v_2){-}(v_1+u_2)\|_\infty$ "t is 1-Lipschitz"
$=$ $\|f{-}g\|_\infty$. "Definition of f,g"

Therefore, t' is also continuous.

t' is positive: (i.e. it maps non-negative functions to non-negative functions). This follows from monotonicity of t.

By Lem. 44, the extension t' further extends into a continuous positive linear function $\tilde{t}{:}\,\mathbb{C}\mathcal{X}{\to}\mathbb{C}\mathcal{X}$ with $\tilde{t}.u=t.u$ for every $u{:}\,\mathbb{U}\mathcal{X}$. □

Before we can finally establish Thm. 29, we prove one last technical lemma that provides uniform convergence (used in the Kantorovich metric on $\mathbb{D}^2\mathcal{X}$) from weak convergence.

A family $\mathcal{A}$ of continuous functions from $\mathbb{D}\mathcal{X}$ to $\mathbb{R}$ is said to be *uniformly bounded* if there exists a real number M such that for every $f{:}\,\mathcal{A}$ and $\delta{:}\,\mathbb{D}\mathcal{X}$ we have $|f.\delta|\leq M$.

The family $\mathcal{A}$ is said to be *equicontinuous at* $\delta{:}\,\mathbb{D}\mathcal{X}$ (with respect to $d_K(,)$) if for each $\varepsilon{>}0$ there exists $\gamma{>}0$ such that for every $\delta'{:}\,\mathbb{D}\mathcal{X}$ with $d_K(\delta,\delta'){<}\gamma$ we have

$$\sup_{f:\,\mathcal{A}}|f.\delta{-}f.\delta'|{\leq}\varepsilon\ .$$

The family $\mathcal{A}$ is (simply) *equicontinuous* if it is equicontinuous at every δ. Intuitively, all functions in an equicontinuous family have the same "degree of continuity" ([40, p. 50]).

Lemma 46: **Technical lemma** For any $\delta{:}\,\mathbb{D}\mathcal{X}$ the family of 1-Lipschitz functions vanishing at δ, that is

$$\mathbb{L}_1^\delta\mathcal{X}\quad:=\quad\{f{:}\,\mathbb{C}\mathcal{X}\mid f.\delta=0\text{ and }f\text{ 1-Lipschitz}\}\ ,$$

is equicontinuous and uniformly bounded.

Proof:

Uniformly bounded by 1: Take arbitrary $f{:}\,\mathbb{L}_1^\delta\mathcal{X}$. For every $\delta'{:}\,\mathbb{D}\mathcal{X}$ we have

$$|f.\delta'|\ =\ |f.\delta'-0|\ =\ |f.\delta'{-}f.\delta|\ \leq\ d_K(\delta',\delta)\ \leq\ 1\ ,$$

that is that $\mathbb{L}_1^\delta\mathcal{X}$ is uniformly bounded.

Equicontinuous: The equicontinuity also follows from Lipschitzness. For $\varepsilon{>}0$ take $\gamma{=}\varepsilon$. For every $f{:}\,\mathbb{L}_1^\delta\mathcal{X}$ and $\delta'{:}\,\mathbb{D}\mathcal{X}$ such that $d_K(\delta,\delta'){<}\gamma$, we have $|f.\delta{-}f.\delta'|\leq d_K(\delta,\delta')<\gamma=\varepsilon$. □

We can now finally give the proof of Thm. 29, using the following version of Riesz Representation and Ranga's result [41], both stated next.

Theorem 47: **Riesz Representation [40, Ch. 2 Thm. 5.8]** Let $\mathbb{D}\mathcal{X}$ be a compact metric space and let l be a positive linear functional on $\mathbb{C}\mathcal{X}$ such that $l.\mathbf{1}=1$. Then there exists a unique probability measure Δ on the Borel algebra of $\mathbb{D}\mathcal{X}$ such that $l.f=\mathcal{E}_\Delta\,f$, for all $f{:}\,\mathbb{C}\mathcal{X}$. □

Theorem 48: **Weak to uniform convergence: separability strengthened to compactness (Ranga) [41, Thm. 3.1].** Let $\mathbb{D}\mathcal{X}$ be a compact metric space. A sequence Δ_n of measures over $\mathbb{D}\mathcal{X}$ converges weakly to Δ [39] iff for each equicontinuous and uniformly bounded class of functions $\mathcal{A}{\subseteq}\mathbb{C}\mathcal{X}$, we have

$$\lim_{n\to\infty}\sup_{f\in\mathcal{A}}|\mathcal{E}_{\Delta_n}f-\mathcal{E}_\Delta f|=0\ .$$

□

[38] For the positiveness of the continuous extension, if f is a positive continuous function that is the uniform limit of a sequence of f_n's in $\langle\mathbb{U}\mathcal{X}\rangle$, then the sequence of positive continuous functions $\max(\mathbf{0},f_n){\in}\langle\mathbb{U}\mathcal{X}\rangle$ also converges to f wrt. the uniform metric. The reason is $|f.\delta{-}\max(0,f_n.\delta)|\leq|f.\delta-f_n.\delta|$, for every $\delta{:}\,\mathbb{D}\mathcal{X}$ and positive f. Thus $t.f$ has to be positive.

[39] That is, the expected values $\mathcal{E}_{\Delta_n}f$ converge to $\mathcal{E}_\Delta f$, for each $f{:}\,\mathbb{C}\mathcal{X}$.

3) Proof of Thm. 29, correspondence of $\mathbb{H}\mathcal{X}$ and $\mathbb{T}\mathcal{X}$: Now we are ready to prove Thm. 29.

Let t be a transformer that satisfies the properties stated in Thm. 29; we shall construct an h such that $\text{wp}.h = t$.

By Lem. 45, there exists a unique positive and continuous extension $\tilde{t}\colon \mathbb{C}\mathcal{X}\to\mathbb{C}\mathcal{X}$ of t. Fix $\delta\colon \mathbb{D}\mathcal{X}$. The function

$$f \quad \mapsto \quad \tilde{t}.f.\delta$$

maps each continuous function $f\colon \mathbb{C}\mathcal{X}$ to $\tilde{t}.f.\delta$ is a positive linear functional on $\mathbb{C}\mathcal{X}$; moreover $\tilde{t}.\mathbf{1}.\delta = t.\mathbf{1}.\delta = 1$, thus $\mathbf{1} \mapsto 1$. Therefore, Thm. 47 implies that there exists a unique Borel probability measure Δ_δ on $\mathbb{D}\mathcal{X}$ such that $\tilde{t}.f.\delta = \mathcal{E}_{\Delta_\delta}\, f$, for every $f\colon \mathbb{C}\mathcal{X}$.

Define $h.\delta := \Delta_\delta$ for each $\delta\colon \mathbb{D}\mathcal{X}$. We now check that h has the required properties demanded by Lemmas 17,19.

Continuity: For the continuity assumption in Lem. 17, we let δ_n be a sequence of distributions in $\mathbb{D}\mathcal{X}$ converging to $\delta\colon \mathbb{D}\mathcal{X}$ with respect to the Kantorovich metric on $\mathbb{D}\mathcal{X}$. It suffices to show that the limit of $d_K(h.\delta_n, h.\delta)$ is 0, as n goes to infinity.

First we rewrite $d_K(h.\delta_n, h.\delta)$ into a form for which Thm. 48 applies. Take $\delta'\colon \mathbb{D}\mathcal{X}$ and recall that $\mathbb{L}_1^{\delta'}\mathcal{X}$ is the set of 1-Lipschitz functions from $\mathbb{D}\mathcal{X}$ to $\mathbb{R}$ that evaluate to 0 at δ'. We have

$$d_K(h.\delta_n, h.\delta) \quad = \quad \sup_{f\in\mathbb{L}_1^{\delta'}\mathcal{X}} |\mathcal{E}_{h.\delta_n}\, f - \mathcal{E}_{h.\delta}\, f| \qquad (24)$$

because $d_K(-,-)$ is invariant by translation (of the f's). By Lem. 46, we know that $\mathbb{L}_1^{\delta'}\mathcal{X}$ is equicontinuous and uniformly bounded.

Finally, in order to apply Thm. 48 to the *rhs* of (24), we must show that the sequence $h.\delta_n$ converges weakly to $h.\delta$. But this is the same as saying that the mapping

$$\delta \quad \mapsto \quad \mathcal{E}_{h.\delta}\, f \ ,$$

is continuous as a function of δ. This follows because $\mathcal{E}_{h.\delta}\, f = \tilde{t}.f.\delta$, by construction, and furthermore $\tilde{t}.f$ is continuous in δ.

Thus Thm. 48 applies and, in this case, says that the *rhs* of (24) converges to 0 for n sufficiently large. That is enough to show that h is continuous.

Super linear: For the super-linearity assumption in Lem. 19, suppose that t=wp.h is in $\mathbb{T}\mathcal{X}$ and take arbitrary $\delta_{1,2}\colon \mathbb{D}\mathcal{X}$. Then we reason

$$\begin{array}{lll} & h.\delta_{1p}{+}h.\delta_2 \ \sqsubseteq\ h.(\delta_{1p}{+}\delta_2) & \\ \text{if} & \mathcal{E}_{(h.\delta_{1p}+h.\delta_2)}\, u \ \leq\ \mathcal{E}_{h.(\delta_{1p}+\delta_2)}\, u & \text{“for all } u\colon \mathbb{U}\mathcal{X} \text{ Lem. 23 Coriaceous”} \\ \text{if} & \quad \text{wp}.h.u.\delta_{1p}{+}\text{wp}.h.u.\delta_2 & \text{“Defn. wp.()”} \\ & \leq\ \text{wp}.h.u.(\delta_{1p}{+}\delta_2) & \\ \text{if} & \text{wp}.h.u \in \mathbb{U}\mathcal{X}\ , & \text{“Defn. } \mathbb{U}\mathcal{X}\text{”} \end{array}$$

which was our assumption.

Finally uniqueness follows directly from Lem. 38 in App. K.

O. Proof of Cor. 30 [§VIII-D]

This proof is made easier by operating in a slightly more general space than $\mathbb{H}\mathcal{X}$, i.e. the measurable subset of $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$, not taking advantage of the stronger conditions that characterise $\mathbb{H}\mathcal{X}$ within it. In this section only we write $\overline{\text{wp}}.$ for the function defined as at Def. 24 but over the larger space.

Lemma 49: **Transformer composition**
For any (measurable) $h_{1,2}\colon \mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ we have that $\overline{\text{wp}}.(h_1; h_2) = \overline{\text{wp}}.h_1 \circ \overline{\text{wp}}.h_2$.

Proof:

$$\begin{array}{lll} & \overline{\text{wp}}.(h_1; h_2).u.\pi & \\ = & \mathcal{E}_{(h_1;h_2).\pi}\, u & \text{“Def. 24 extended to } \overline{\text{wp}}.()\text{”} \\ = & \mathcal{E}_{\mathsf{avg}.(\mathbb{D}h_2.(h_1.\pi))}\, u & \text{“}h_1; h_2 \text{ is Kleisli composition”} \\ = & & \text{“}\mathcal{E}_{\mathsf{avg}.\underline{\Delta}}\, u = \mathcal{E}_{\underline{\Delta}}\, (\lambda\Delta\cdot\mathcal{E}_\Delta\, u) \text{ from (†) below} \\ & \mathcal{E}_{\mathbb{D}h_2.(h_1.\pi)}\, (\lambda\Delta\cdot\mathcal{E}_\Delta\, u) & \lambda \text{ is lambda-abstraction”} \\ = & & \text{“}\mathcal{E}_{\mathbb{D}h_2.\Delta}\, F = \mathcal{E}_\Delta\, (F{\circ}h_2) \text{ from (‡) below”} \\ & \mathcal{E}_{h_1.\pi}\, ((\lambda\Delta\cdot\mathcal{E}_\Delta\, u)\circ h_2) & \\ = & & \text{“make } \pi' \text{ explicit”} \\ & \mathcal{E}_{h_1.\pi}\, (\lambda\pi'\cdot((\lambda\Delta\cdot\mathcal{E}_\Delta\, u)\circ h_2).\pi') & \\ = & \mathcal{E}_{h_1.\pi}\, (\lambda\pi'\cdot\mathcal{E}_{h_2.\pi'}\, u) & \text{“}\Delta{:=}\, h_2.\pi'\text{”} \\ = & \overline{\text{wp}}.h_1.(\lambda\pi'\cdot\mathcal{E}_{h_2.\pi'}\, u).\pi & \text{“Def. 24”} \\ = & \overline{\text{wp}}.h_1.(\overline{\text{wp}}.h_2.u).\pi\ . & \text{“Def. 24”} \end{array}$$

The identities (†) and (‡) were proven by Giry ([2, Sec. 3 p.70]). In (‡), F maps every hyper Δ to $\mathcal{E}_\Delta\, u$. □

Our next step is to use Thm. 29 to show that indeed $h_1; h_2 \in \mathbb{H}\mathcal{X}$, so that $\overline{\text{wp}}.$ can be replaced by wp.() in Lem. 49 just above. We have

Lemma 50: **$\mathbb{H}\mathcal{X}$ closed under composition**
For $h_{1,2}\colon \mathbb{H}\mathcal{X}$ we have $h_1; h_2 \in \mathbb{H}\mathcal{X}$.

Proof: If $h_{1,2}\colon \mathbb{H}\mathcal{X}$ then wp.$h_{1,2}\in\mathbb{T}\mathcal{X}$ from Lems. 26,27; and since those properties are closed under composition, we have that wp.$h_1\circ$wp.$h_2\in\mathbb{T}\mathcal{X}$ as well.

From Thm. 29 there is then a unique $h\colon \mathbb{H}\mathcal{X}$ such that wp.h = wp.$h_1\circ$wp.h_2; but examination of Lem. 38 shows membership of $\mathbb{H}\mathcal{X}$ is not necessary for that uniqueness: it applies to the whole of (measurable) $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$. That is, there no other measurable h in all of $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$ such that $\overline{\text{wp}}.h = t$.

But from Lem. 49 we know that $\overline{\text{wp}}.(h_1; h_2) = t$, and so we must have $h_1; h_2 = h \in \mathbb{H}\mathcal{X}$. □

Thus we can conclude

Corollary 30: Transformer composition
For any $h_{1,2}\colon \mathbb{H}\mathcal{X}$ we have that also $h_1; h_2 \in \mathbb{H}\mathcal{X}$, and furthermore wp.$(h_1; h_2)$ = wp.$h_1\circ$wp.h_2.

Proof: Lemmas 49,50 just above. □

P. Calculation of wp.$[\![P]\!]$ [§X-B]

In §X-B a sample analysis was done on a very small program to show how, if the post-uncertainty is fixed, a pre-uncertainty can be calculated once and for all; and that then that pre-uncertainty can be used to investigate the security implications of a number of different priors, without having to re-analyse the program for each one.

Here we give the calculations for wp.$(\cdot)$ in §X-B. We note below however that ideally the pre-uncertainty would be calculated by *source-level* reasoning; but that is not what we do here. (See also our "more concrete aim" in §XIII concerning source-level reasoning.)

Let P be the program set out in Fig. 5 (and also Fig. 6 from App. D). As usual for weakest preconditions, we work from post- to pre-. Let u be the *UM* from §X-A, reflecting the circumstances of an attacker whose principal concern is whether the two bits of $\mathtt{xs}$ are the same.

Beginning with the second statement, since with transformers we work from the back towards the front, we expect informally that wp.$[\![\mathtt{xs:=\ xs}_{1/2}\oplus\mathtt{-xs}]\!].u$ is just u again — since the assignment does not affect $\mathtt{xs}[0]{=}\mathtt{xs}[1]$, whichever branch is taken. Calculation confirms that: for abritrary π we have

$$
\begin{array}{ll}
& \text{wp.}[\![\mathtt{xs:=\ xs}_{1/2}\oplus\mathtt{-xs}]\!].u.\pi \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“semantics of } \mathtt{xs:=\ xs}_{1/2}\oplus\mathtt{-xs}\text{”} \\
& \mathcal{E}_{\,[(\ (\pi_{00}+\pi_{11})/2,(\pi_{01}+\pi_{10})/2,\quad u} \\
& \qquad\ (\pi_{10}+\pi_{01})/2,(\pi_{11}+\pi_{00})/2\ )] \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“expectation over point hyper”} \\
& u.\ (\ (\pi_{00}+\pi_{11})/2,(\pi_{01}+\pi_{10})/2, \\
& \qquad\ (\pi_{10}+\pi_{01})/2,(\pi_{11}+\pi_{00})/2\ ) \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“definition } u \text{ from §X-B”} \\
& \qquad\quad (\pi_{00}+\pi_{11})/2+(\pi_{11}+\pi_{00})/2 \\
& \min\quad (\pi_{01}+\pi_{10})/2+(\pi_{10}+\pi_{01})/2 \\
= & \ (\pi_{00}+\pi_{11})\ \min\ (\pi_{01}+\pi_{10}) \\
= & \ u\ , \qquad\qquad\qquad\qquad\qquad\qquad \text{“definition } u \text{ again”}
\end{array}
$$

as we expected.

Continuing towards the front of the program we now calculate again for arbitrary π, but from just above able to use the same u that we started with, that

$$
\begin{array}{ll}
& \text{wp.}[\![\mathtt{print\ xs}[0]\ {}_{1/2}\!\oplus \mathtt{xs}[1]]\!].u.\pi \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“ semantics of } \mathtt{print\ xs}[0]\ {}_{1/2}\!\oplus \mathtt{xs}[1] \\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{define } s_0{:=}\pi_{00}+(\pi_{01}+\pi_{10})/2 \\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad\ \ s_1{:=}(\pi_{01}+\pi_{10})/2+\pi_{11}\ \text{”} \\
& \mathcal{E}_{\ \ (\pi_{00}/s_0,\pi_{01}/2s_0,\pi_{10}/2s_0,0)\quad u} \\
& \ {}_{s_0}\!\oplus\ (0,\pi_{01}/2s_1,\pi_{10}/2s_1,\pi_{11}/s_1) \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“}\mathcal{E} \text{ linear, applied to two-point hyper (Def. 10)”} \\
& \qquad\quad u.(\pi_{00}/s_0,\pi_{01}/2s_0,\pi_{10}/2s_0,0) \\
& {}_{s_0}\!\oplus\quad u.(0,\pi_{01}/2s_1,\pi_{10}/2s_1,\pi_{11}/s_1) \\
= & \qquad\qquad\qquad\qquad\qquad\qquad \text{“definition } u \text{ from previous calculation”} \\
& \qquad\quad \pi_{00}\min(\pi_{01}{+}\pi_{10})/2+(\pi_{01}{+}\pi_{10})/2\min\pi_{11}\ ,
\end{array}
$$

as claimed in §X-B.

We stress that calculating wp.() this way for any but the smallest programs is *not practical at all*. For a practical calculus, instead the formulation of uncertainties as loss functions would be used to write them as expressions at the source level, i.e. over program variables, and then using formal manipulations in a quantitative program logic (extending e.g. [18], [19]).

The issue of source-level reasoning is discussed further in the conclusion §XIII.

Q. Using loss functions to characterise pure channels

With uncertainty transformers, we can be more precise about the properties satisfied by pure-(abstract) channel *HMM*'s specifically. As with markovs the mechanism by which information is released is independent of the (probability) values associated with the prior; in fact it only depends on the underlying state value, that is $\mathcal{X}$. This property can be described neatly in terms of a "multiplicative property" on transformers which, in addition, provides a characterisation of transformers which correspond to channels. We begin with a motivating example.

Take $\mathcal{X}{=}\{0,1\}$. It's easy to construct an $h{:}\,\mathbb{H}\mathcal{X}$ with the property that for all $\pi{:}\,\mathbb{D}\mathcal{X}$ we have $\mathsf{avg}.(h.\pi) = \pi$, which is to say that its markov is the identity, but it is still not a pure channel: we simply "cheat" by using a different channel for each prior. Take for example the π-indexed channels given by the matrix

$$C^\pi \quad := \quad \begin{pmatrix} \pi_0 & \pi_1 \\ 0 & 1 \end{pmatrix} .$$

The function defined $f.\pi{:=}[\![\pi{\triangleright}C^\pi]\!]$ does not satisfy $f{=}[\![C{:}]\!]$ for any *single* fixed C, and this example provides the insight for characterising pure channels: they have a simple multiplicative property, which we express using loss functions as follows.

Definition 51: **Multiplicativity of transformers** For loss-function $l{:}\,I{\to}\mathcal{X}{\to}\mathbb{R}^{\geq}$ and $\pi{:}\,\mathbb{D}\mathcal{X}$ define a π-skewed loss function $(l{\triangleleft}\pi).i.x{:=}l.i.x{\times}\pi.x$. We then say that transformer $t{:}\,\mathbb{T}\mathcal{X}$ is *multiplicative* if for any $\pi_{1,2}{:}\,\mathbb{D}\mathcal{X}$ and loss function l we have $t.(U_{l\triangleleft\pi_1}).\pi_2 = t.(U_{l\triangleleft\pi_2}).\pi_1$.[40] □

Lemma 52: **Channels are multiplicative** Let $C{:}\,\mathcal{X}{\to\!\!\bullet}\mathcal{Y}$ be a channel matrix. Then $\mathrm{wp}.[\![C{:}]\!]$ is multiplicative.

Proof: This follows because the identity transformer is multiplicative, i.e. $(U_{l\triangleleft\pi_1}).\pi_2 = (U_{l\triangleleft\pi_2}).\pi_1$, and that $\mathrm{wp}.()$ applied to a pure channel maps any given loss function to a sum of loss functions "scaled" by the columns. □

The following fact shows that this multiplicative property in fact characterises channels.

Lemma 53: Let $f{:}\,\mathbb{D}\mathcal{X}{\to}\mathbb{D}^2\mathcal{X}$ be such that $f.\pi$ has finite support for every $\pi{:}\,\mathbb{D}\mathcal{X}$; assume it satisfies the pure-channel property from §B6; and assume that $\mathrm{wp}.f$ is multiplicative as just above. Then there is some set of observations $\mathcal{Y}$ and channel $C{:}\,\mathcal{X}{\to\!\!\bullet}\mathcal{Y}$ such that $f = [\![C{:}]\!]$.

Proof: Let N be the size of $\mathcal{X}$ and let υ be the uniform distribution on $\mathcal{X}$.[41] Define $\Delta{:=}f.\upsilon$ and let $\mathcal{Y}$ be the support of Δ, a finite set of distributions that will be used as column indices. Then define $C{:}\,\mathcal{X}{\to\!\!\bullet}\mathcal{Y}$ by

$$C_{x,y} \quad := \quad N \times \Delta.y.x \ ,$$

so that $f.\upsilon = \Delta = [\![C{:}]\!].\upsilon$. We now show that in fact $f.\pi = [\![C{:}]\!].\pi$ for all $\pi{:}\,\mathbb{D}\mathcal{X}$.

We have for any loss function l that

$$\begin{array}{lll}
& \mathcal{E}_{f.\pi}\, U_l & \\
= & \mathrm{wp}.f.U_l.\pi & \\
= & \mathrm{wp}.f.U_{l'\triangleleft\upsilon}.\pi & \text{“define } l'{:=}N{\times}l\text{”} \\
= & \mathrm{wp}.f.U_{l'\triangleleft\pi}.\upsilon & \text{“assumption wp.}f\text{ multiplicative”} \\
= & \mathcal{E}_{f.\upsilon}\,(U_{l'\triangleleft\pi}) & \\
= & \mathcal{E}_{[\![C{:}]\!].\upsilon}\,(U_{l'\triangleleft\pi}) & \text{“defn. } C\text{”} \\
= & \mathcal{E}_{[\![C{:}]\!].\pi}\, U_l \ , & \text{“reverse steps above; wp.}[\![C{:}]\!]\text{ multiplicative”}
\end{array}$$

so $f.\pi{=}[\![C_\Delta{:}]\!].\pi$ since hypers are determined by loss functions [4], [6], thus $f{=}[\![C_\Delta{:}]\!]$ because π was arbitrary.

□

[40] This notation is by analogy with $\pi{\triangleright}C$ that "multiplies π in" from the $\mathcal{X}$ side of a matrix; in $C{\triangleleft}\pi$ the π is multiplied in from the other side.

[41] It is *upsilon* for "uniform".