BRICS

Basic Research in Computer Science

BRICS DS-03-14 D. Varacca: Two Denotational Models for Probabilistic Computation

Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation

Daniele Varacca

BRICS Dissertation Series DS-03-14

ISSN 1396-7002 November 2003

See back inner page for a list of recent BRICS Dissertation Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `DS/03/14/`

Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation

Daniele Varacca

PhD Dissertation

SOLIDUM PETIT IN PROFUNDIS · UNIVERSITAS ARHUSIENSIS

BRICS

BRICS PhD School
Department of Computer Science
University of Aarhus
Denmark

Probability, Nondeterminism and Concurrency: Two denotational models for probabilistic computation

A Dissertation
Presented to the Faculty of Science
of the University of Aarhus
in Partial Fulfillment of the Requirements for the
PhD Degree

by
Daniele Varacca
August 29, 2003

Abstract

Nondeterminism is modelled in domain theory by the notion of a powerdomain, while probability is modelled by that of the probabilistic powerdomain. Some problems arise when we want to combine them in order to model computation in which both nondeterminism and probability are present. In particular there is no categorical *distributive law* between them. We introduce the *powerdomain of indexed valuations* which modifies the usual probabilistic powerdomain to take more detailed account of where probabilistic choices are made. We show the existence of a distributive law between the powerdomain of indexed valuations and the nondeterministic powerdomain. By means of an equational theory we give an alternative characterisation of indexed valuations and the distributive law. We study the relation between valuations and indexed valuations. Finally we use indexed valuations to give a semantics to a programming language. This semantics reveals the computational intuition lying behind the mathematics.

In the second part of the thesis we provide an operational reading of continuous valuations on certain domains (the distributive concrete domains of Kahn and Plotkin) through the model of *probabilistic event structures.* Event structures are a model for concurrent computation that account for causal relations between events. We propose a way of adding probabilities to confusion free event structures, defining the notion of probabilistic event structure. This leads to various ideas of a run for probabilistic event structures. We show a confluence theorem for such runs. Configurations of a confusion free event structure form a distributive concrete domain. We give a representation theorem which characterises completely the powerdomain of valuations of such concrete domains in terms of probabilistic event structures.

Acknowledgements

Ok so apparently I did it. I wrote a PhD thesis. I want to thank Vincent Danos and Mike Mislove for carefully reading it and for being part of my evaluation committee.

Of course I did not do it all by myself. Many people helped me.

First of all I want to thank my supervisor Glynn Winskel. He gave me lots of good advice and contributed with experience, intuition and good ideas to this thesis, but he also introduced me to the pleasures of Vindaloo. My relation with Glynn has been open and honest, and it is for this honesty that I want to thank him most.

Although this thesis is mainly the joint work of Glynn and myself, many important conversations contributed in a way or another, and I want to acknowledge Mario Cáccamo, Andrzej Filinski, Marcelo Fiore, Thomas Hildebrandt, Michael Mislove, Gordon Plotkin, Luigi Santocanale, Peter Sewell, Zhe Yang. Special thanks go to Achim Jung, with whom I had many fruitful discussions and who made me like Birmingham.

Special thanks also to Mogens Nielsen. BRICS has been an amazing environment and if a person can represent it, this is Mogens. I want to thank him for support, encouragement and for discussing with me ideas that led to the work on event structures. All teaching staff at BRICS are wonderful, but I want to mention in particular Olivier Danvy, for his support, his memorable advice on how to give talks, and for P.D.Q. Bach. BRICS is also made of non-teaching staff and I have to thank Janne Christensen, whose smiling email was the first contact I had with my new life in Denmark, Karen Møller, Lene Kjeldsteen, Uffe Engberg and Ingrid Larsen. Last and most importantly, BRICS gave me the opportunity to meet other PhD students and researchers from all over the world. Too many friends to name them all, but I want at least to mention: the 1999 foreign students Maciej Koprowski, Paulo Oliva, Jirka Srba and Dan Hernest, for everything we have shared, from the dirty dorms to the complexity theory homeworks; Bernd Grobauer, for unveiling the German in me; Federico Crazzolara and Giuseppe Milicia, for reminding me of my Italian roots; Oliver Möller, for the conversations we had over a game of cards; Zhe Yang, for all the passions we have in common, from espresso to Mozart; Frank Valencia, despite everything we don't have in common; Marcin Jurdziński and Riko Jacob, for their enthusiasm toward climbing and life.

Århus is a beautiful city, and I am grateful to have had the privilege of living there for two years. The long summer evenings, the brisk and clear winter days, the botanic garden, the beaches, the forests. I will miss it.

For the second half of my studies I lived in Cambridge. This city grew on me and I will miss it too. I have to thank the Computer Laboratory, for having me as long term guest. Thanks in particular to Marcelo Fiore, Peter Sewell, and to all the students in the Theory group. Special thanks to Mario Cáccamo with whom I spent much time talking about science and sharing PhD depression. Thanks go to Emmanuel College, in particular to Neil Dodgson, for having me as guest. Thanks to all the people at Emmanuel Boat Club, in particular my captains Joe, Dave and Alex, for allowing me to experience the thrill of the Bumps. I also want to thank my housemates John and Cordula and their cat Lotti for two wonderful years in Lizzie Way.

I spent lots of free time climbing (or lots of time free climbing): thanks to Århus Klatreklubben, in particular Dorthe, Erik and Dieter, for introducing me to climbing. Thanks to the Cambridge University Mountaineering Club, in particular to Trond, James, Kat, Barbara, Moira, Crispin and Sibe, for all the trips to the Peak District and beyond. Special thanks to Andrei Serjantov both as fellow PhD student and as trustworthy climbing partner.

Many other people have been supporting me with their friendship. In particular I want to thank Benedetta for the Greek myths, Magda and Wiola for Częstochowa, Claudia for the couscous, Rossana, Francesca, Giovanni and Andrea for being always there, and Alina for the second movement of the 7th Symphony, the Waltz in A Major and the cheesecake (yummy).

The road toward the PhD did not start four years ago, but 26 years ago. Since this is the end of my life as a student (at last!), I want to thank all the teachers that contributed to this. E siccome sono quasi tutti italiani, in italiano li ringrazio. In particolare il mio pensiero va a Marta Saccani, Patrizia Copelli, Alberto Fontana, Claudia Termini, Alberto Arosio, Pino Rosolini e Marco Forti.

My family has been always supportive of my studies, although I know that in recent years my mum would have liked to see me more often. Papà, Mamma, Toe, I love you.

Daniele Varacca
Århus and Cambridge, August 2003.

Contents

Part I

Introduction

Chapter 1

The Nature of Computation

Computing machines are a fundamental component of the human environment. Our cars, mobile phones, banks, supermarkets, and therefore our whole everyday life, rely on computers. It is extremely important that computers do what we expect from them. The need to understand computation is vital.

In the era of the Internet, computers perform concurrent activities, communicating continuously with each other. Complex systems of several communicating machines are extremely difficult to understand. On top of the inherent complexity of the system, we have to deal with the possibility of failure of each component, with failing and noisy channels of communication and with malicious intrusions.

Often we cannot reasonably hope to understand the *exact* behaviour of a complex system, but we may still have enough information to determine the *probability* that some behaviour is observed.

Sometimes we can also *exploit* a probabilistic behaviour to our advantage. All modern security protocols, for instance, involve the use of randomness, and it is clear how widespread is the use of computer security, from e-commerce to military applications.

It is therefore important to understand the nature of computation involving probability and concurrency. To obtain this, we make use of mathematical structures, called *models*. It is crucial that the models are *abstract* enough to ignore small details that may hinder our comprehension and yet have enough *complexity* to provide useful information.

An essential tool for modelling computation is the mathematical notion of nondeterminism. Nondeterminism is not a feature of real systems, but is used to obtain higher levels of abstraction and to compose simple models in order to build more complex ones.

This thesis is a step towards a better understanding of computation involving probability, nondeterminism and concurrency.

1.1 Semantics of computation

Semantics is the art of giving computer programs mathematical meaning. At the most concrete level, programs can be just seen as strings of bits. They are given as input to computers that perform different actions depending on which string has been given. A first rough notion of meaning for a program is simply the behaviour of the computer. However, just as we cannot understand the psychology of a person by studying the chemical reactions in their body, we find the need of higher levels of *abstraction*, in order to really understand what programs do.

To obtain this, we give semantics to programs, that is we build mathematical models, we associate every program to some entity in a model, and then we study the model. Often we give different semantics to the same program and we study the relations between them.

The study of semantical models has various aims. First of all, semantical models can give us information about a specific program. If we want to know what a program does, we can study its semantics in some model, and obtain the information we are looking for.

A second use of semantical models is to help us in designing better programming languages. While we give semantics we may realise that, if the language were organised in a different way, giving semantics would be easier, or clearer. A programming language with a neat semantics can be easier to reason about.

Finally we can use semantical models to understand the nature of computation. In some cases we may realise that giving semantics would be easier, if the *model* were structured in a different way. Or we may realise that it is impossible to give semantics using the model we have. Sometimes models seem appropriate at the first sight, but a more thorough study reveals unexpected problems. This tells us that our understanding of what we want to model is flawed or incomplete, and must be reviewed. On the other hand, if we can formalise mathematically some intuition we have, we strengthen our belief in that intuition.

This thesis mainly fits the last approach. We study two mathematical models for computation involving probability. The first model is an example of how a pure mathematical notion (the notion of distributive law in category theory) enhances our understanding of the nature of computation. The second is a formal model for probabilistic concurrent computation that confirms some intuitions we have.

In both cases the main part of the work is purely mathematical, with definitions, lemmas, theorems. However, some computational intuition will be provided along the way.

1.1.1 Operational versus denotational

Among the different semantical models, it is customary to make distinctions between *operational* models, and *denotational* models. Operational models tend to be built upon some notion of an abstract computer, while denotational models use other mathematical frameworks. Denotational models are usually characterised by *compositionality*: the denotation of a complex program is obtained by composing (in some rigorous mathematical sense) the denotations of the simpler parts of the program.

Sometimes, though, operational semantics is compositional too. We feel that the distinction between the two kinds of semantics is blurred, but such, rather philosophical, discussion is beyond the scope of this thesis. For an introduction to the topic, we refer to Winskel's book [Win93].

1.1.2 Denotational semantics

Denotational semantics originated in the sixties from the work of Dana Scott and Christopher Strachey [SS71]. Domain theory [Plo83, AJ94, G$^+$03] was mainly developed as a mathematical foundation for denotational semantics. Domain theoretic models were used to give semantics to functional programming languages, from the untyped lambda-calculus [Sco72] to PCF [Plo77, Mil77]. It was soon discovered that in domain theory it was difficult to model the notion of sequential evaluation. More complex tools were invented, see for example [Ong95, HO00].

Labelled transition systems (see [WN95]) are the standard models for concurrent computation, but they are usually considered to belong to the class of operational models. More denotational in flavour is the notion of event structure [Win80, Win87]. Event structures and domain theory are related [NPW81]. More recently Winskel proposed the categorical notion of *presheaf* as suitably rich denotational model for concurrency [NW03].

1.1.3 Nondeterminism and Concurrency

Nondeterminism is an important semantical concept, often used for abstracting away from details. When we model a program using nondeterminism, we want to model the fact that the program *can* perform different actions, without recording any further information on which action *will be* actually performed. As pointed out by previous authors (see for instance [Seg95, dA97, Sto02]), this feature is useful in the following situations

- **implementation** The high level description of a language abstracts away from implementation details. Different implementations produce different behaviours. To account for them, nondeterminism is used. Often one aims to showing that the behaviour is essentially independent from the implementation details.

- **scheduling** In concurrent systems, sometimes one wants to abstract away from the order in which independent components perform their computations. This can be done using nondeterminism.

- **communication with the environment** In the compositional semantics of concurrent systems, one wants to model different components separately and combine them. When modelling one component we have to account for the possibility of communicating with other components. Before the system is put together we do not know which communications will actually happen. The possibility of performing different communications is dealt with using nondeterminism.

- **competition for communication** Even when a concurrent system is put together, different components may compete for communication. This competition may be resolved by the scheduling of the components or may depend on details we want to abstract away from.

- **lack of probabilistic information** When different behaviours are possible one can sometimes estimate the relative probabilities. Sometimes, though, one cannot do this, because the information available is not enough to produce a meaningful estimate, and so one must resort to nondeterminism

Most labelled transition systems use nondeterminism to model concurrency. Such models are called *interleaving* models. A computation is a linear sequence of event. When two events are concurrent, they can happen in any order nondeterministically. Event structures instead provide a non-linear notion of computation. Such models are called *causal* models, because the order in which two events happen is recorded only in so far as there is some causal relation between them. Both event structures and transitions systems still feature nondeterminism.

There are also various possible views of nondeterminism.

1. A nondeterministic system can show different behaviours and sometimes we have simply to account for all the possibilities.

2. If we consider the various possibilities as a menu from which we can choose the action we like, we may ignore bad behaviours.

3. If the choice is not under our control but it is made by some external agent, we usually imagine it as malicious as possible, and we consider only the worst cases.

The notion of powerdomain [Plo76, Smy78] was introduced to model nondeterminism in domain theory. The three different views of nondeterminism described above correspond to three different notion of powerdomain, the Plotkin, the Hoare and the Smyth powerdomain respectively [Win83].

1.1.4 Probabilistic computation

There are various applications of probability theory to computer science. We can design algorithms which take advantage of random choices during the computation [Sri95]. We can design cryptographic protocols using random choices to increase security [GM99]. In a distributed setting we can use random choices to break symmetries [Lyn96, HP00]. On the other hand probabilistic models allow us to consider phenomena (noise, malfunction, intrusion) which in the real world can affect computations [Han91]. Probability theory is also a fundamental ingredient in the theory of quantum computation [Bra, NC00].

Sequential programming languages can feature probabilistic choice explicitly as a constructor [Jon90], or via random assignment (if the language is state-based) [Koz81]. The explicit approach is usually followed for concurrent languages [vG^{+}90, GJS90, BK00, Low93, Sei95].

Probability has been modelled in domain theory through probabilistic powerdomains [SD80, Jon90, JT98, Eda95a]. Various kinds of probabilistic transition systems exist [vG^{+}90, LS91]. Often such transition systems model both probability and nondeterminism [SL95]. The only probabilistic notion of event structure we are aware of is the one presented in Katoen's thesis [Kat96].

Sometimes probabilistic choice is considered as a refined version of nondeterministic choice. When different behaviours are available, we might decide to choose by flipping a coin in order to mislead some kind of "adversary". This adversary could be the one that finds the worst case for an algorithm, could be an eavesdropper, or could be a scheduling policy that maintains an undesirable symmetry. Alternatively, we might know some information on how the choice is made by the external agent. In both cases we can tell with which probability each possible action is performed.

Once “atomic” actions are endowed with probabilities, we can evaluate the overall probability that some observable event happens. We might want to know with which probability a value is output, a final state is attained, or a sequence of actions is performed. Due to well known limitations of measure theory, we cannot always assign probabilities to all possible observations, and sometimes we have to restrict to the “measurable” ones.

In some computational models we can keep track of how long it takes to perform an action. These durations can sometimes be expressed by random variables. We will not deal with this kind of models in this thesis.

1.1.5 Probability and nondeterminism

Models have been studied which deal with both nondeterminism and probability, especially in concurrent systems. Even when we consider probabilistic choice as refinement of nondeterministic choice, we still want to abstract away from scheduling policies and we need to model communication. As observed above, nondeterminism is required for this. There are extensions of CCS [Han91], CSP [$\mathrm{M}^{+}94$] , and π-calculus [HP00] which combine probability and nondeterminism. There are various operational models of this kind [BSdV03]. One of the leading models is that of probabilistic automata [SL95, Sto02]. On the domain theoretic side, some work had already been done to combine the probabilistic powerdomain with the nondeterministic one [Mis00, Tix99]. We will discuss this work in Chapter 4.

1.2 Contents and structure of the thesis

I outline here the structure of the thesis. There are three parts: an introductory part, and one part for each model we study. The last two parts are logically independent of each other, and can be read in any order.

In Chapter 2 we provide a quick overview of the preliminary notions we need for the main body of the thesis. There is little original work there, although the section on domain theory contains some definitions of mine.

In Part II we study the notion of indexed valuation, as a denotational model for probabilistic computation. This model arises from the need of combining probabilities and nondeterminism. The probabilistic powerdomain and the nondeterministic powerdomain do not combine nicely. In technical terms, there is no distributive law between the two monads. We face this mathematical problem discovering where the core of the problem lies and we propose our solution which amounts to a modification of the probabilistic powerdomain. First, we perform our constructions simply using sets. This provides already most of the intuitions and is also preparatory to the more involved and technical construction of domain theory. Finally we use our construction to give denotations to a simple programming language.

Chapter 3 begins by showing the failure of the distributive law. We then define indexed valuations in the category of sets. We show that they form a monad. We show the existence of a distributive law between the indexed valuation monad and the finite nonempty powerset monad. We provide an equational characterisation of such construction. Finally we provide a construction for a different equational theory, inspired by the work of Tix and Mislove. In Chapter 4 we define indexed valuations in the category of continuous domains. We show the relation between indexed valuations and continuous valuations. We show the existence of the distributive law between indexed valuations and the Hoare powerdomain. We propose alternative definitions of indexed valuations and discuss their merits and limitations. In Chapter 5 we give an operational and denotational semantics to a programming language with probabilistic primitive. We study the language both with and without recursion, and show adequacy theorems. Then we add nondeterministic choice to the language. We give operational semantics in terms of probabilistic automata, and we give denotational semantics both in terms of indexed valuations and in terms of the convex powerdomain of Tix and Mislove. Different adequacy theorems show the computational meaning of the two models.

In part III we study the notion of probabilistic event structures, as a causal model for probabilistic concurrent computation. We discuss the problems arising when one wants to add probabilities to event structures. This leads us to study the restricted notion of confusion free event structure. In this restricted setting we are able to extend the classic theory of event structures. We show connections between probabilistic event structures and domain theory. We also show connections between probabilistic event structures and the standard interleaving models.

In Chapter 6 we define the notion of confusion free event structure. We then define the notion of valuations on a confusion free event structure. We first define a notion that assumes probabilistic independence of all choices and then we generalise it removing this assumption. We show the relation between valuations on the event structure and continuous valuations on the domain of configurations. We define a further generalised notion of valuation on a confusion free event structure that uses subprobability distributions. This allows us to characterise completely continuous valuations in terms of valuations on event structures. Finally we provide a categorical view of probabilistic event structures. In Chapter 7 we define various notions of a run of an event structure. We first define a linear notion, similar to the corresponding notion in the interleaving models. Then we define a notion more sensitive to the causal nature of event structures. We discuss some of the properties of the latter. In particular we show a confluence theorem for such runs. Using runs, we give an alternative characterisation of valuations on an event structure. Finally we characterise the maximal valuations on the domain of configurations.

1.3 Acknowledgements of collaboration

I acknowledge here the external contributions to my work. Of course, I take responsibility for the many errors and inelegances still contained in this thesis.

In autumn 2000 I was having difficulties in combining the probabilistic and nondeterministic monads. Luigi Santocanale had suggested to me to use a distributive law, but I could not prove the obvious definition correct. Andrzej Filinski showed me how he had managed to combine the two monads in ML.

I noticed that his implementation of the probabilistic monad in terms of lists did not satisfy the law $A +_p A = A$ and I decided to exploit this observation. At the same time, after an email correspondence, Gordon Plotkin proved that there is no categorical distributive law between the nondeterministic monad and the usual probabilistic monad. At the beginning I was trying to describe the new construction in terms of multisets, but Glynn Winskel suggested the better idea of using indices. We were then able to show the existence of the distributive law between indexed valuations and powerset in the category of sets. Later I found out about Gautam's theorem (3.1.1), which implies the existence of the distributive law. I worked a lot in the category of continuous domains, but the lack of concrete representation made it difficult to prove the existence of the distributive law. Achim Jung suggested the idea of exploiting the bases, although I'm still not able to produce a nice, short, elegant proof based on that. Later I had the idea of using Beck's theorem instead.

At the beginning of 2002 I was trying to study a probabilistic version of the language SPL of Federico Crazzolara and Glynn Winskel [CW01]. This language has a semantics in terms of Petri nets, so I was looking for a suitable notion of probabilistic Petri nets. I could not find any, therefore in the spring 2002 I talked with Mogens Nielsen about it, we came up with a nice definition and we produced a paper [VN03]. Mogens observed that confusion free Petri nets were particularly well behaved and he had the idea that led to the first formulation of the confluence theorem in terms of Mazurkievicz equivalence. The original proof I gave of that theorem was extremely technical and unfortunately the referees were not impressed. The notion of probabilistic Petri nets we present in [VN03] corresponds to that of event structure together with a local valuation. At the same time Glynn suggested the idea of building probabilistic models out of a simple process language and defined the notion of global valuation with independence. This notion was later generalised, removing the independence assumption.

My original notion of run was the one that is now called 'inductive test'. Later Glynn saw that the right definition for a run would be the one of finitary test, and suggested the characterisation of global valuations in terms of tests (Theorem 7.4.1) I had the idea that every global valuation with independence would extend to a continuous valuation and I proved it. The statements of the two combinatorial lemmas are mine. The proof of the first is due to Glynn while the BSV lemma was proved by Moritz Becker, Gareth Stoyle and myself, during a hot afternoon in room FE22 of the William Gates Building.

Chapter 2

Preliminary Notions

This chapter introduces the notions we need in the next two parts. Most of the material is not original. We assume little previous knowledge from the reader: basically some set theory and some category theory. For everything else we aim at giving a self contained, although quick, introduction to the subject matter. In the first section we fix some notational convention. Section 2.2 is on category theory, mainly monads and adjunctions. We also define the notion of distributive law and present the notion of monad as model for computational effects. Section 2.3 is on domain theory, mainly continuous domains, and it contains some definitions of mine, that I did not find in the literature. Section 2.4 introduces some notions of universal algebra and the notion of nondeterministic powerdomain. Section 2.5 presents the relevant notions of measure theory, with particular focus on the notion of valuation for a topology. Section 2.6 introduces various notions of transition systems, with particular focus on probabilistic automata.

All sections introduce the relevant notation. The reader can find pointers to the most important symbols and key words in the index at the end of the thesis.

2.1 Notation

In this thesis we will refer to more entities than there are letters in the Greek and Latin alphabets combined. Some overloading of the notation is therefore inevitable. We will try to respect the following general guidelines.

Capital letters A, B, C, X, Y, Z usually denote sets. They are sometimes decorated with indices, dashes, tildes. Elements of such sets are denoted by lowercase letters a, b, c, x, y, z, possibly decorated. When a set is used to index a family, it is denoted by letters from the middle of the alphabet I, J, K. Its elements are denoted by i, j, k. Sometimes we will use calligraphic letters $\mathcal{S}, \mathcal{T}$ to denote sets of sets.

If $\Phi(x)$ is some predicate and X is a set, we write $\{x \in X \mid \Phi(x)\}$ to denote the set of elements of X which satisfy Φ. We write $\{x \mid \Phi(x)\}$ when X is understood to be some kind of "universe". We will use $|X|$ to denote the cardinality of X.

When $f : X \to Y$ is a function, and $A \subseteq X$ we denote by $f(A)$ the image of A under f, i.e. the set $\{y \in Y \mid \exists a \in A.f(a) = y\}$. If $B \subseteq Y$, we denote by $f^{-1}(B)$ the inverse image of B under f, i.e. the set $\{x \in X \mid \exists b \in B.f(x) = b\}$. If f is a partial function, the *domain* of f is the subset of X where f is defined.

If $\equiv$ is an equivalence relation on a set X, we denote the set of its equivalence classes by $X/\equiv$.

If X is a set, the powerset of X will be denoted by $\mathcal{P}(X)$. The finite and nonempty powerset will be denoted by $P(X)$, while the finite powerset (including the empty set) will be denoted by $P_\perp(X)$. When Y is a finite set and $Y \subseteq X$ we write $Y \subseteq_{fin} X$. If I is a set, functions $f : I \to X$ are sometimes called *families* indexed by I. A family indexed by I is sometimes denoted by $(x_i)_{i\in I}$ or simply (x_i). Families indexed by the set of natural numbers are called *sequences.* Sometimes, we use the word 'family' also to denote a set of sets.

As usual $X \cup Y$ denotes union, $X \cap Y$ denotes intersection and $X \setminus Y$ denotes the set $\{x \in X \mid x \notin Y\}$. If $\mathcal{S}$ is a set of sets, we write $\bigcup \mathcal{S}$ to denote $\{x \mid \exists X \in \mathcal{S}.x \in X\}$ and $\bigcap \mathcal{S}$ to denote $\{x \mid \forall X \in \mathcal{S}.x \in X\}$. If $(X_i)_{i\in I}$ is a family of sets, we write $\bigcup_{i\in I} X_i$ to denote $\{x \mid \exists i \in I.x \in X_i\}$ and $\bigcap_{i\in I} X_i$ to denote $\{x \mid \forall i \in I.x \in X_i\}$. We write $X \uplus Y$ to denote $X \cup Y$ when $X \cap Y = \emptyset$.

Sometimes we use the lambda notation to denote functions. If $exp(x)$ is an expression such that for every $z \in X$, $exp(z) \in Y$ we write $\lambda x.exp(x)$ to denote the function $f : X \to Y$ such that $f(z) = exp(z)$.

When we want to define a term t using an expression *exp* we use the "assignment" notation $t := exp$.

We will use the symbol $\mathbb{N}$ to denote the set of natural numbers $\{0, 1, 2, \ldots\}$. The cardinality of $\mathbb{N}$ is denoted by $\aleph_0$. A set of cardinality $\aleph_0$ is called *countable*. With the notation I_n we denote the set $\{1, 2, \ldots, n\} \subseteq \mathbb{N}$. Note that in this thesis I_n begins with 1.

The symbol $\mathbb{Q}$ denotes the rational numbers, while $\mathbb{R}$ denotes the real numbers. We use the usual interval notation for real numbers, with $[x, y[$ for example denoting the set $\{z \in \mathbb{R} \mid x \leq z < y\}$. With $\mathbb{R}^+$ we denote the set of *nonnegative* real numbers. With $\overline{\mathbb{R}^+}$ we denote the set $\mathbb{R}^+ \cup \{+\infty\}$ where $+\infty$ is a new element and the order relation is extended so as to make $+\infty$ the maximum. Addition and multiplication are extended to $\overline{\mathbb{R}^+}$ by

$$\infty + x = \infty; \quad \infty \cdot x = \begin{cases} \infty & \text{if } x > 0; \\ 0 & \text{if } x = 0. \end{cases}$$

Let X be a set and let $f : X \to \overline{\mathbb{R}^+}$. For every $Y \subseteq X$ we define

$$f[Y] := \sum_{y\in Y} f(y) := \sup_{Z \subseteq_{fin} Y} \sum_{z\in Z} f(z)\,.$$

I like to point out the following, possibly not very well known, fact.

Proposition 2.1.1. *If $f[Y] < +\infty$ then $|Y| \leq \aleph_0$.*

Proof: [Pro70] For every $n \in \mathbb{N}$ consider the set $Y_n := \{y \in Y \mid f(y) \geq \frac{1}{n}\}$. Clearly $|Y_n| \leq n \cdot f[Y]$, that is Y_n is finite. Notice that $Y = \bigcup_{n\in\mathbb{N}} Y_n$. Since the countable union of finite sets is at most countable, Y is at most countable. □

If X is a set, the set of finite strings over X is denoted by X^*, the set of countable sequences is denoted by X^ω and $X^\infty := X^* \cup X^\omega$.

Other conventions will be introduced along with the introduction of new concepts.

2.2 Category theory

Category theory is recognised to be one of the most important mathematical tools in theoretical computer science. Its abstractness and expressiveness allow us to see connections between different subjects, and provide fruitful insights. It is way beyond the scopes of this introduction to support this argument. Some interesting reading on this can be found in [WN95, AJ94, Mog91, C0́3] and elsewhere.

2.2.1 Basic notions and notation

We refer, for the basic definitions to [Mac71], a good elementary introduction is also [CHW02]. We assume the reader knows the notions of category, functor, natural transformation, adjunction.

We use the following notation: $A \xrightarrow{f} B[\mathbf{C}]$ means that f is an arrow between A, B in the category $\mathbf{C}$. Also we write $A \in \mathbf{C}$ when A is an object of $\mathbf{C}$. Often we write $A \xrightarrow{f} B$ or $f : A \to B$ when the category is clear from the context. The identity on an object A is denoted by Id_A. Commutative diagrams are used in the standard way, as graphical representation of equalities between arrows.

Natural transformations are arrows in the functor category. We usually use Greek letters for natural transformations. Sometimes we will denote them with the dot notation: $\alpha : F \overset{\cdot}{\longrightarrow} G$. We denote horizontal composition by juxtaposition: if $\alpha : F \overset{\cdot}{\longrightarrow} G$ and $\beta : F' \overset{\cdot}{\longrightarrow} G'$, then $\beta\alpha : F'F \overset{\cdot}{\longrightarrow} G'G$. In this context we denote the identity natural transformation on a functor F by the letter F as well. We denote an adjunction $F \dashv G$ by $(F, G, \eta, \epsilon) : \mathbf{C} \to \mathbf{D}$ where $F : \mathbf{C} \to \mathbf{D}$ and η, ϵ are the unit and the counit. A standard theorem that we use is the following, characterising adjunction in term of universality.

Theorem 2.2.1 ([Mac71]). *Let $G : \mathbf{D} \to \mathbf{C}$ be a functor. Suppose for every object $A \in \mathbf{C}$ there exists an object $F(A) \in \mathbf{D}$ and a morphism $A \xrightarrow{\eta_A} G(F(A))[\mathbf{C}]$ satisfying the following* universal property*: for every $A \xrightarrow{f} G(X)[\mathbf{C}]$ there exists a unique g, $F(A) \xrightarrow{g} X[\mathbf{D}]$ such that*

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[dr, "f"'] & G(F(A)) \arrow[d, "G(g)"] \\
 & G(X)\,.
\end{tikzcd}$$

Then F can be uniquely extended to a functor $F : \mathbf{C} \to \mathbf{D}$ such that $F \dashv G$ and η is the unit of such adjunction.

Note that it is not required that F be a functor: it is a consequence of universality. If $X \xrightarrow{f} Y[\mathbf{C}]$ then $F(f)$ is defined as the unique arrow such that

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "f"'] \arrow[dr] & G(F(X)) \arrow[d, "G(F(f))"] \\
Y \arrow[r, "\eta_Y"'] & G(F(Y))\,.
\end{tikzcd}$$

This also shows that η is automatically natural.

2.2.2 Algebras and coalgebras

An *algebra* for an endofunctor $F : \mathbf{C} \to \mathbf{C}$ is simply an object $A \in \mathbf{C}$ together with an arrow $F(A)\xrightarrow{k}A$. The object A is called the *carrier*, while the arrow k is called the *structure*. The dual concept is called *coalgebra*.

Algebras form a category $\mathbf{C}^F$, where a morphism $(X,k)\xrightarrow{\phi}(X',k')[\mathbf{C}^F]$ is given by an arrow $X\xrightarrow{\phi}X'[\mathbf{C}]$ such that the following diagram commutes.

$$\begin{tikzcd}
F(X) \arrow[r, "F(\phi)"] \arrow[d, "k"'] & F(X') \arrow[d, "k'"] & {[\mathbf{C}]} \\
X \arrow[r, "\phi"'] & X' &
\end{tikzcd}$$

The initial object in the category of algebras, when it exists, is called *initial algebra*. The dual notion is that of *final coalgebra*.

Coalgebras are an important tool in theoretical computer science. Transition systems and bisimulations of different kinds can be defined using coalgebras [RT94]. A *bisimulation* between two coalgebras $(X,k),(X',k')$ is an object R together with arrows $R\xrightarrow{p}X, R\xrightarrow{p'}X'[\mathbf{C}]$ such that there exists a coalgebraic structure r on R which makes p, p' coalgebra morphisms. Two coalgebras are *bisimilar* if there exists a bisimulation between them.

The leading example are labelled transition systems [RT94]. Consider the functor $X \mapsto \mathcal{P}(A \times X)$ in the category $\mathbf{SET}$ of sets and functions. A coalgebra for this functor is a *A-labelled transition system.* Two transition systems are coalgebraically bisimilar if and only if they are bisimilar in the sense of Park and Milner [Mil89].

2.2.3 Monads

A *monad* on a category $\mathbf{C}$ is an endofunctor $T : \mathbf{C} \to \mathbf{C}$ together with two natural transformations, $\eta^T : Id_{\mathbf{C}} \to T$, the *unit*, and $\mu^T : T^2 \to T$, the *multiplication*, satisfying the following axioms.

$$\begin{tikzcd}
T \arrow[r, "T\eta^T"] \arrow[dr, "Id_T"'] & T^2 \arrow[d, "\mu^T"] & T \arrow[l, "\eta^T T"'] \arrow[dl, "Id_T"] \\
& T &
\end{tikzcd}$$

$$\begin{tikzcd}
T^3 \arrow[r, "T\mu^T"] \arrow[d, "\mu^T T"'] & T^2 \arrow[d, "\mu^T"] \\
T^2 \arrow[r, "\mu^T"'] & T
\end{tikzcd}$$

If T is a monad and if $f : X \to T(Y)$, the *Kleisli extension* $f^\dagger : T(X) \to T(Y)$ is defined as $T(X)\xrightarrow{T(f)}T(T(Y))\xrightarrow{\mu^T_Y}T(Y)$. The *Kleisli Category* of the monad T, denoted by $\mathbf{C}_T$ has the same objects as $\mathbf{C}$ and $X\xrightarrow{f}Y[\mathbf{C}_T]$ if and only if

$X \xrightarrow{f} T(Y)[\mathbf{C}]$. The identity is the unit of the monad, while composition is defined using the Kleisli extension.

Monads can equivalently be defined using the Kleisli extension as a primitive notion and deriving the multiplication by $\mu^T_X := Id^{\dagger}_{T(X)}$.

2.2.4 Monads as computational notion

Eugenio Moggi [Mog91] introduced the idea of using monads to represent computational effects in denotational semantics. The semantics of a typed programming language can be given in terms of category theory. Types are interpreted by objects, while terms in context are interpreted by morphisms. Composition of morphisms corresponds to substitution of variables for terms. Categorical product is used to model terms with more than one free variable. If the language admits function types, the model is required to be a cartesian closed category. Recursion is usually modelled by using a category of domains. For instance if $x : \tau_1 \vdash t : \tau_2$ is a term in context, its interpretation is a morphism $[\![t]\!] : [\![\tau_1]\!] \to [\![\tau_2]\!]$. Intuitively the program t takes in input values of type τ_1 and returns values of types τ_2. This intuition does not take into account different computational features such as nondeterminism, probabilistic choice, nontermination, exceptions and so on. A program may fail to return any value, or may return many values, etc... The notion of a monad is a way to embrace all these possibilities.

Suppose $T : \mathbf{C} \to \mathbf{C}$ is a monad. The general idea is that if an object $[\![\tau]\!]$ represents values of type τ, then $T([\![\tau]\!])$ represents *computations* of type τ. The unit $\eta^T_X : X \to T(X)$ interprets values as computations, while the multiplication $\mu^T_X : T(T(X)) \to T(X)$ "flattens" computations over computations. Terms of the language are not interpreted in the category $\mathbf{C}$ but in the Kleisli category $\mathbf{C}_T$. Intuitively a program takes values as input but returns computations. The powerset (or powerdomain) monad is used to model nondeterminism, the powerdomain of valuations is used to model probabilities, and so on.

In many cases the monad is freely generated by the operations we want to model. The nondeterministic powerdomains are generated by the nondeterministic choice operator $\uplus$. The normalised probabilistic powerdomain is generated by the probabilistic choice operator $\oplus_p$. A recent account of this point of view can be found in [PP02]. See also Section 2.4.

2.2.5 Algebras for monads

An algebra for a monad (T, η^T, μ^T) is an algebra (A, k) for the functor T, satisfying the following compatibility axioms.

$$\begin{tikzcd}
A \arrow[r, "\eta^T_A"] \arrow[dr, "Id_A"'] & T(A) \arrow[d, "k"] \\
 & A
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
T^2(A) \arrow[r, "\mu^T_A"] \arrow[d, "T(k)"] & T(A) \arrow[d, "k"] \\
T(A) \arrow[r, "k"'] & A
\end{tikzcd}$$

When (T, η^T, μ^T) is a monad on $\mathbf{C}$, we denote the category of algebras for T by $\mathbf{C}^T$. It will be clear from the context whether we talk of the algebras for the *functor* or the algebras for the *monad*.

Every adjunction $(F, G, \eta, \epsilon) : \mathbf{C} \to \mathbf{D}$ generates a monad on $GF : \mathbf{C} \to \mathbf{C}$, with $\eta^{GF} := \eta$ and $\mu^{GF} := G\epsilon F$. Conversely, given a monad T, there is an adjunction $F^T \dashv U^T : \mathbf{C} \to \mathbf{C}^T$, where U^T is the forgetful functor sending an algebra to its carrier

$$U^T\colon \begin{tikzcd} (X,k) \arrow[d, "\phi"'] & & X \arrow[d, "\phi"] \\ (X',k') & \mapsto & X' \end{tikzcd}$$

while F^T sends an object to its multiplication (the *free algebra*)

$$F^T\colon \begin{tikzcd} X \arrow[d, "\phi"'] & & (TX, \mu^T_X) \arrow[d, "T(\phi)"] \\ X' & \mapsto & (TX', \mu^T_{X'}) \,. \end{tikzcd}$$

This adjunction generates precisely the monad (T, η^T, μ^T).

Suppose we have an adjunction $(F, G, \eta, \epsilon) : \mathbf{C} \to \mathbf{D}$ generating a monad (T, η^T, μ^T). Such a monad generates the adjunction $(F^T, U^T, \eta^T, \epsilon^T)$. There is a "comparison" functor $K : \mathbf{D} \to \mathbf{C}^T$, defined as

$$K\colon \begin{tikzcd} D \arrow[d, "f"'] & & (G(D), G(\epsilon_D)) \arrow[d, "G(f)"] \\ D' & \mapsto & (G(D'), G(\epsilon_{D'})) \end{tikzcd}$$

satisfying $U^T K = G$ and $KF = F^T$.

2.2.6 Beck's monadicity theorem

The following fundamental theorem characterises the so called *monadic* adjunctions, that is the adjunctions freely generated by monads. To understand its statements we need to recall the definition of creation and of split coequaliser. A functor $F : \mathbf{C} \to \mathbf{D}$ *creates* the limit for a functor $D : \mathbf{I} \to \mathbf{C}$ when, if $F \circ D$ has a limit $(L, (f_i)_{i \in \mathbf{I}})$ in $\mathbf{D}$ then there is a D-cone $(L', (f'_i)_{i \in \mathbf{I}})$ in $\mathbf{C}$ such that $(F(L'), (F(f'_i))_{i \in \mathbf{I}})$ is a limit in $\mathbf{D}$ and all such D-cones are limiting for D. More simply, in order for F to create the limit for D it is enough that if $(L, (f_i)_{i \in \mathbf{I}})$ is a limit for $F \circ D$ in $\mathbf{D}$ then there exists a D-cone $(L', (f'_i)_{i \in \mathbf{I}})$ in $\mathbf{C}$ such that $(F(L'), (F(f'_i))_{i \in \mathbf{I}}) = (L, (f_i)_{i \in \mathbf{I}})$ and all such cones are limits for D.

A *split coequaliser* for a parallel pair of arrows $C \overset{f}{\underset{g}{\rightrightarrows}} D$ is an arrow $D \xrightarrow{e} E$ such that there exist two arrows $E \xrightarrow{s} D$ and $D \xrightarrow{t} C$ satisfying $ef = eg$, $es = 1_E$, $tf = 1_D$, $tg = se$.

As the name suggests, split coequalisers are indeed coequalisers. More than this, they are *absolute* coequalisers, that is they are preserved by any functor whatever. We can now state the theorem.

Theorem 2.2.2 (Beck). *The comparison functor K defined above is an equivalence if and only if the functor $G : \mathbf{D} \to \mathbf{C}$ creates coequalisers for those parallel arrows f, g for which the pair Gf, Gg has a split coequaliser in $\mathbf{C}$.*

This is the form in which we will use this theorem. Further details can be found in Maclane's book [Mac71].

2.2.7 Distributive laws

A general tool for combining two monads is the notion of *distributive law* [Bec69]. Suppose we have two monads (T, η^T, μ^T), (S, η^S, μ^S) on some category. A *distributive law* of S over T is a natural transformation $d : ST \overset{\cdot}{\longrightarrow} TS$ satisfying the following axioms:

$$\begin{tikzcd} & T \arrow[dl, "\eta^S T"'] \arrow[dr, "T\eta^S"] & \\ ST \arrow[rr, "d"'] & & TS \end{tikzcd} \qquad \begin{tikzcd} & S \arrow[dl, "S\eta^T"'] \arrow[dr, "\eta^T S"] & \\ ST \arrow[rr, "d"'] & & TS \end{tikzcd}$$

$$\begin{tikzcd} SST \arrow[r, "Sd"] \arrow[d, "\mu^S T"'] & STS \arrow[r, "dS"] & TSS \arrow[d, "T\mu^S"] \\ ST \arrow[rr, "d"'] & & TS \end{tikzcd}$$

$$\begin{tikzcd} STT \arrow[r, "dT"] \arrow[d, "S\mu^T"'] & TST \arrow[r, "Td"] & TTS \arrow[d, "\mu^T S"] \\ ST \arrow[rr, "d"'] & & TS\ . \end{tikzcd}$$

With a distributive law we can define a monad on the functor TS. If $d : ST \overset{\cdot}{\longrightarrow} TS$ is a distributive law, then $\big(TS, \eta^T\eta^S, (\mu^T\mu^S) \circ TdS\big)$ is a monad.

$$\begin{tikzcd} TSTS \arrow[r, "\cdot", "TdS"'] & TTSS \arrow[r, "\cdot", "\mu^T\mu^S"'] & TS \end{tikzcd}$$

A *monad morphism* between T and S is a natural transformation $\alpha : T \overset{\cdot}{\longrightarrow} S$ which suitably commutes with units and multiplications. A *lifting* of the monad T to the category of S-algebras is a monad $(\tilde{T}, \eta^{\tilde{T}}, \mu^{\tilde{T}})$ on $\mathbf{C}^S$, such that, if $U^S : \mathbf{C}^S \to \mathbf{C}$ is the forgetful functor then

- $U^S\tilde{T} = TU^S$;
- $U^S\eta^{\tilde{T}} = \eta^T U^S$;
- $U^S\mu^{\tilde{T}} = \mu^T U^S$.

Beck has proved the following theorem [Bec69].

Theorem 2.2.3. *Suppose we have two monads (T, η^T, μ^T), (S, η^S, μ^S) on some category $\mathbf{C}$. Then the following are equivalent*

1. *distributive laws $d : ST \overset{\cdot}{\longrightarrow} TS$;*
2. *multiplications $\mu : TSTS \overset{\cdot}{\longrightarrow} TS$, such that*
    - *$(TS, \eta^T\eta^S, \mu)$ is a monad;*

- *the natural transformations $\eta^T S : S \stackrel{\cdot}{\longrightarrow} TS$ and $T\eta^S : T \stackrel{\cdot}{\longrightarrow} TS$, are monad morphisms;*
- *the following middle unit law holds:*

$$\begin{tikzcd}
TS \arrow[r, "T\eta^S\eta^T S"] \arrow[dr, "Id_{TS}"'] & TSTS \arrow[d, "\mu"] \\
 & TS
\end{tikzcd}$$

3. liftings $\tilde{T}$ of the monad T to $\mathbf{C}^S$.

The way to obtain (2) from (1) has been sketched above. To obtain a lifting from a distributive law we define $\tilde{T}(A,\sigma)$ as the S-algebra

$$\begin{tikzcd}
ST(A) \arrow[r, "d_A"] & TS(A) \arrow[r, "T(\sigma)"] & T(A)\ .
\end{tikzcd}$$

Conversely if we have the multiplication μ we can define d by

$$\begin{tikzcd}
ST \arrow[r, "\eta^T TS\eta^S"] & TSTS \arrow[r, "\mu"] & TS\ .
\end{tikzcd}$$

If we have a lifting $\tilde{T}$, we define d by

$$\begin{tikzcd}
ST \arrow[r, "TS\eta^S"] \arrow[dr, "d"'] & STS \arrow[r, equal] & U^SF^STU^SF^S \arrow[r, equal] & U^SF^SU^S\tilde{T}F^S \arrow[d, "U^S\epsilon\tilde{T}F^S"] \\
 & TS \arrow[r, equal] & TU^SF^S \arrow[r, equal] & U^S\tilde{T}F^S\ ,
\end{tikzcd}$$

where ϵ is the counit of the adjunction $F^S \dashv U^S$

The correctness of the above constructions is shown by several diagram chases [Bec69].

2.3 Domain theory

Domain Theory is the earliest mathematical foundation for denotational semantics. It is based on the idea of modelling recursion via fixed points. An early but still valid overview can be found in [Plo83], while the standard references are [AJ94], and the very recent account [G$^+$03]. We introduce here only the notions we use.

2.3.1 Partial orders

A *preorder* on a set X is a reflexive and transitive relation on X. We will use the term preorder also to denote a structure $(X, \leq)$, where $\leq$ is a preorder on X. An antisymmetric preorder is called a *partial order*. If what follows $(X, \leq)$ is a partial order. We also write $x < x'$ to mean $x \leq x'$ and $x \neq x'$.

An element $x \in X$ is *maximal*, if for all $y \in X$, $x \leq y \Longrightarrow x = y$. An element $x \in X$ is a *maximum* if for all $y \in X$, $y \leq x$. Clearly there is at most one maximum. Dually one defines *minimal* and *minimum* elements. The maximum is sometimes called *top* and it is denoted by $\top$. The minimum is sometimes called *bottom* and it is denoted by $\bot$.

When $Y \subseteq X$, an element $x \in X$ is an *upper bound* for Y, denoted by $Y \leq x$, if $y \in Y \implies y \leq x$. Any element is an upper bound of the empty set. If the set of all upper bounds of Y has a minimum, this is called the *supremum* or the *least upper bound* of Y, sometimes abbreviated as *lub*. The lub of Y, when it exists, is denoted by $\bigsqcup Y$.

A subset Y of X is *downward closed* or *lower* if $y \in Y$ and $x \leq y$ implies $x \in Y$. The dual concept is that of *upward closed* or *upper* set. If Y is any subset of X, the *downward closure* of Y is the set $\downarrow Y := \{z \in Y \mid \exists y \in Y, z \leq y\}$. The notion of *upward closure* $\uparrow Y$ is defined dually.

A subset Y of X is *directed* if for every finite set $F \subseteq Y$, there exists $y \in Y$ such that $F \leq y$. The dual concept is that of *filtered* set. Note that a directed subset is not empty (consider the case $F = \emptyset$). The lub of a directed subset Y is denoted by $\bigsqcup^{\uparrow} Y$.

A lower directed set is called an *ideal*. The dual concept is that of a *filter.*

If $x \in X$ the set $\downarrow x := \downarrow\{x\} = \{y \in X \mid y \leq x\}$ is an ideal. It is called the *principal ideal* generated by x. The dual notion, denoted by $\uparrow x$ is that of *principal filter.*

A subset Y of a partial order $(X, \leq)$, is a *chain* if for every $y, y' \in Y$, either $y \leq y'$ or $y' \leq y$. If X is a chain then $(X, \leq)$ is a *total order.*

A partial order is *well founded* if every nonempty chain has a minimum. Well founded partial orders provide a notion of *induction.* If X is well founded and if $Y \subseteq X$ we have that $Y = X$ if and only if for every $x \in X$, if $z < x \implies z \in Y$ then $x \in Y$

If $(X, \leq), (X', \leq)$ are two preorders, a function $f : X \to X'$ is called *monotonic* [1], or *covariant* if $x \leq y \implies f(x) \leq f(y)$. It is called *contravariant*[2] if $x \leq y \implies f(x) \geq f(y)$. If X is a set, Y is a preorder and $f, g : X \to Y$ are two functions, then we say $f \leq g$ if for every $x \in X$, $f(x) \leq g(x)$.

Finally we present a definition, which is the dual of the more famous notion of embedding projection pair.

Definition 2.3.1. Let X, Y be partial orders, and let $i : X \to Y$, $c : Y \to X$. We say that (i, c) is a *insertion closure pair* if $i \circ c = Id_Y$ and $c \circ i \geq Id_X$.

Note that if (i, c) is an insertion closure pair, then c is injective, while i is surjective.

2.3.2 Domains

A *directed complete partial order* (DCPO) is a partial order $(D, \sqsubseteq)$ such that every directed subset of D has a lub. If D, E are two DCPOs, a function $f : D \to E$ is *continuous* if it is monotonic and for every directed subset $X \subseteq D$, $f \bigsqcup^{\uparrow} X = \bigsqcup^{\uparrow} f(X)$.

DCPOs will be usually denoted with the letters D, E. The order relation on a DCPO will be usually denoted by $\sqsubseteq$.

[1] This is the standard name for that concept. A more precise denomination would be *isotonic.*

[2] Or *antitonic.*

In every DCPO D an *approximation* relation $\ll$ is definable (also known as the *way-below* relation). We say that $x \ll y$ if $y \sqsubseteq \bigsqcup^{\uparrow} Z \implies \exists z \in Z.x \sqsubseteq z$. When $x \in D$ we write $\mathord{\downarrow\!\!\downarrow} x$ to denote the set $\{y \in D \mid y \ll x\}$, and dually for $\mathord{\uparrow\!\!\uparrow} x$. An element x is *compact* if $x \ll x$. Let B be a subset of a DCPO domain D. For every $x \in D$, we define $B_x = \mathord{\downarrow\!\!\downarrow} x \cap B$. A subset B of a DCPO D is a *basis* if for every $x \in D$, $x = \bigsqcup^{\uparrow} B_x$. A DCPO D with a basis is called a *continuous domain.* It is called ω-continuous if it has a countable basis. If B is a basis for D and $B \subseteq B'$ then B' is also a basis for D. A DCPO D is an *algebraic domain* if its compact elements form a basis. It is called ω-algebraic if it has a countable basis.

An example of a continuous domain is $(\overline{\mathbb{R}^+}, \leq)$. Its way-below relation is

$$p \ll q \text{ iff } (p < q \text{ or } p = 0) \,.$$

A basis of $\overline{\mathbb{R}^+}$ is $\mathbb{Q}^+ := \mathbb{Q} \cap \mathbb{R}^+$. Therefore $\overline{\mathbb{R}^+}$ is actually ω-continuous. It is not algebraic, though, as the only compact element is 0. Another example is the set S^∞ of finite and infinite words on a finite alphabet S, with the prefix order. The finite words form a basis of compact elements, so that S^∞ is ω-algebraic.

The category of DCPOs and continuous functions is called **DCPO**. Similarly we have the categories $\mathbf{CONT}, \mathbf{ALG}, \omega\mathbf{CONT}, \omega\mathbf{ALG}$.

One way of interpreting the order on a DCPO is by saying that greater elements provide more *information.* One interpretation of the way-below relation is the following: if $x \ll y$ then the information provided by x is an *essential part* of the information provided by y. More discussion on this subject can be found in [AJ94].

The main theorem that justifies the use of domain theory in computer science is the following.

Theorem 2.3.2. *Let D be a DCPO with bottom, and let $f : D \to D$ be continuous. Then f has a minimum fixed point, that is the set of $d \in D$ for which $f(d) = d$ is nonempty and has a minimum.*

Fixed points are used to give denotations to recursive programs.

2.3.3 Abstract bases

The properties of the way-below relation suggest the following definition.

Definition 2.3.3. A relation $\lhd$ on a set X is an *AB-relation* if it is transitive and satisfies the finite interpolation property: for every $F \subseteq_{fin} X$ and for every $x \in X$, $F \lhd x \implies \exists y \in X.\ F \lhd y \lhd x$. The structure $(X, \lhd)$ is called an *abstract basis*.

Indeed, for every continuous domain D, with basis B, the structure $(B, \ll)$ is an abstract basis. A preorder is also an abstract basis. For AB-relations we shall use the same terminology as for preorders. We therefore speak of monotonic functions, lower sets, directed sets, and so on. In particular we recall that an ideal is a lower directed set. The set of ideals of X is called $\mathit{Idl}(X)$. For any $x \in X$ the set $\iota_X(x) := \Downarrow x := \{y \mid y \lhd x\}$ is an ideal. Notice that directedness implies the following property, that we call *roundness*: if $\mathcal{I}$ is an ideal of $(X, \lhd)$, then for every $x \in \mathcal{I}$ there exists $x' \in \mathcal{I}$ such that $x \lhd x'$.

The structure $(\mathit{Idl}(X), \subseteq)$ is a continuous domain with basis $\iota_X(X)$. It is called the *ideal completion* of X. The function $\iota_X : X \to \mathit{Idl}(X)$ preserves the AB-relation, but it is not injective in general. Conversely if D is a continuous domain with basis B, then $(B, \ll)$ is an abstract basis whose ideal completion is isomorphic to D. The isomorphism $\beta_B : D \to \mathit{Idl}(B)$ is defined by $\beta_B(d) = B_d$. If we consider D as a basis of itself, then β_D is $\iota_D : D \to \mathit{Idl}(D)$.

When the AB-relation is a preorder, $(\mathit{Idl}(X), \subseteq)$ is algebraic. Conversely, if D is an algebraic domain, then the way below relation is a partial order on the set of compact elements.

Let $(X, \lhd)$ be an abstract basis, $(D, \sqsubseteq)$ be a (not necessarily continuous) DCPO, and $f : X \to D$ be a function mapping $\lhd$ to $\sqsubseteq$ ("monotonic"). The function $f^\sharp : \mathit{Idl}(X) \to D$ defined as

$$f^\sharp(I) = \bigsqcup_{x \in I}^{\uparrow} f(x)$$

is continuous. We have that $f^\sharp(\iota(x)) \sqsubseteq f(x)$. The converse inequality does not hold in general.

2.3.4 Weakly monotonic functions

We are going to introduce some new concepts, which cannot be found in [AJ94]. We will need them in Chapter 4.

Suppose $f : D \to D'$ is a continuous function. And suppose that it restricts to a function $f : B \to B'$ between two bases of D, D'. Then f does not necessarily preserve the way below relation. However it satisfies the following property.

Definition 2.3.4. If $(X, \lhd)$, $(Y, \lhd)$ are two abstract bases and if $f : X \to Y$ is a function, we say that f is *weakly monotonic*, if $x \lhd x' \implies \Downarrow f(x) \subseteq \Downarrow f(x')$.

That is, a function $f : X \to Y$ is weakly monotonic if $\iota_Y \circ f : X \to \mathit{Idl}(Y)$ is monotonic. Therefore we can define the *extension* $\mathit{ext}(f)$ to be the continuous function $(\iota \circ f)^\sharp : \mathit{Idl}(X) \to \mathit{Idl}(Y)$.

In particular $f : X \to Y$ is weakly monotonic if it is *strongly monotonic*, that is, if it preserves the AB-relation.

Weakly monotonic functions do not compose in general, we have to add some hypothesis.

Definition 2.3.5. Let $f : (X, \lhd) \to (Y, \lhd)$ be a weakly monotonic function between abstract bases. We say that f is *complete* if whenever $y \lhd f(x)$ there exists $x' \lhd x$ such that $y \lhd f(x')$.

Proposition 2.3.6. *Let $f : X \to Y$ and $g : Y \to Z$ be weakly monotonic functions between abstract bases. Then*

- *if f is strongly monotonic, then $g \circ f$ is weakly monotonic;*
- *if g is complete, then $g \circ f$ is weakly monotonic;*
- *if f, g are both complete, then $g \circ f$ is also complete.*

Moreover, in all cases above $\mathit{ext}(g) \circ \mathit{ext}(f) = \mathit{ext}(g \circ f)$.

Proof: We want to prove that when $x \lhd x'$ then $\Downarrow g(f(x)) \subseteq \Downarrow g(f(x'))$. If f is strongly monotonic then $f(x) \lhd f(x')$ and since g is weakly monotonic we conclude $\Downarrow g(f(x)) \subseteq \Downarrow g(f(x'))$.

If g is complete, take $z \lhd g(f(x))$. By completeness there exists $y' \lhd f(x)$ such that $z \lhd g(y')$. Since f is weakly monotonic, then $y' \lhd f(x')$. Since g is weakly monotonic, then $\Downarrow g(y') \subseteq \Downarrow g(f(x'))$, hence $z \lhd g(f(x'))$.

If f also is complete, take $z \lhd g(f(x))$. By completeness of g there exists $y' \lhd f(x)$ such that $z \lhd g(y')$. By completeness of f there exists $x' \lhd x$ such that $y' \lhd f(x')$. Since f is weakly monotonic, then $z \lhd g(f(x'))$.

To prove the final statement, let I be an ideal in $\mathit{Idl}(X)$. We have to show that $\mathit{ext}(g)(\mathit{ext}(f)(I)) = \mathit{ext}(g \circ f)(I)$. By definition $\mathit{ext}(f)(I) = \bigsqcup^{\uparrow}{}_{x \in I}\, i(f(x)) = \bigcup_{x \in I} \Downarrow f(x)$. Also

$$\mathit{ext}(g)(\mathit{ext}(f)(I)) = \bigsqcup^{\uparrow}_{y \in \mathit{ext}(f)(I)} i(g(y)) = \bigcup_{y \in \mathit{ext}(f)(I)} \Downarrow g(y)\,.$$

Therefore $z \in \mathit{ext}(g)(\mathit{ext}(f)(I))$ if and only if there exist $x' \in I, y \lhd f(x')$ such that $z \lhd g(y)$. Similarly $z \in \mathit{ext}(g \circ f)(I)$ if and only if there exists $x \in I$ such that $z \lhd g(f(x))$. Since g is weakly monotonic we have that if $y \lhd f(x')$, then $z \lhd g(y) \implies z \lhd g(f(x'))$, so that $\mathit{ext}(g)(\mathit{ext}(f)(I)) \subseteq \mathit{ext}(g \circ f)(I)$. To show the other inclusion, assume first that f is strongly monotonic. Let $x \in I$ such that $z \lhd g(f(x))$, by roundness there exists $x' \in I$ such that $x \lhd x'$. Since f is strongly monotonic, then $f(x) \lhd f(x')$, and for $y = f(x)$ we have the result. If g is complete there exists $y \lhd f(x)$ such that $z \lhd g(y)$, and again we are done. □

It is also interesting to note the following:

Proposition 2.3.7. *Let $f : (X, \lhd) \to (Y, \lhd)$ be a weakly monotonic function between abstract bases. Consider the extension $\mathit{ext}(f) : \mathit{Idl}(X) \to \mathit{Idl}(Y)$. We have $\mathit{ext}(f)(\iota_X(x)) = \iota_Y(f(x))$*

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\iota_X"'] & Y \arrow[d, "\iota_Y"] \\
\mathit{Idl}(X) \arrow[r, "\mathit{ext}(f)"'] & \mathit{Idl}(Y)
\end{tikzcd}$$

if and only if f is complete.

2.4 Free constructions

As pointed out in Section 2.2, computational monads often arise as free algebras for suitable equational theories. We present here some results in the theory of universal algebra and their connections with category theory and domain theory. A standard reference is [Coh81]. A good short introduction is in the handbook chapter [AJ94], which we will follow closely.

2.4.1 Signatures and equational theories

A signature Σ is a pair (Ω, α) where Ω is a set of *operation symbols* and $\alpha : \Omega \to \mathbb{N}$ assigns to every symbols its *arity.* An *absolute* Σ*-algebra* for Σ is an algebra k

for the functor $F^{\Sigma} : Y \mapsto \biguplus_{f\in\Omega} Y^{\alpha(f)}$ in the category **SET**. For every operation symbol f, the f-th component of k is a function $f_k : Y^{\alpha(f)} \to Y$. Morphisms of Σ-algebras are functions that *respect* (or commute with) the operations. That is, if h is a Σ-algebra morphism then

$$h(f_k(y_1, \ldots, y_{\alpha(f)})) = f_k(h(y_1), \ldots, h(y_{\alpha(f)}))\,.$$

For any set X consider the functor $Y \mapsto \Sigma(Y) \uplus X$. Its initial algebra exists and its carrier $T(X)$ is the set of *terms* over X. Initiality amounts to saying that if (Y, h) is a Σ-algebra, then every function $j : X \to Y$ can be extended uniquely to a Σ-morphism $j^{\sharp} : T(X) \to Y$.

Terms can be described inductively in the usual way by:

- if $x \in X$ then x is a term;

- if $t_1, \ldots, t_n$ are terms, $f \in \Omega$ and $\alpha(f) = n$ then $f(t_1, \ldots, t_n)$ is a term.

The operator T is indeed a monad in **SET** whose unit interprets elements of X as terms while the multiplication flattens terms of terms into terms (removing parentheses, so to speak).

Fix a countable set V of *variables*. An equational theory is a set of pairs $E \subseteq T(V) \times T(V)$. Every pair of terms $\langle t_1, t_2\rangle$ is interpreted as an equation $t_1 = t_2$. If (Y, k) is a Σ-algebra, and if $j : V \to Y$ is a function, we have the extension $j^{\sharp} : T(V) \to Y$. An algebra (Y, k) satisfies an equational theory E if for every equation $\langle t_1, t_2\rangle \in E$ and every $j : V \to Y$, $j^{\sharp}(t_1) = j^{\sharp}(t_2)$. Such algebras are called (Σ, E)-algebras.

The category of (Σ, E)-algebras is denoted by **SET**(Σ, E), or **SET**(Σ), if E is empty. Using the general adjoint functor theorem and Beck's theorem, it can be proved that the forgetful functor $U : \mathbf{SET}(\Sigma, E) \to \mathbf{SET}$ is monadic. Its left adjoint functor is called the *free algebra* functor for (Σ, E).

2.4.2 Domains and monadicity

All definitions above can be recast in different categories of domains, giving rise to "domain-algebras" where all the operations are continuous. The difference is that pairs in $T(V) \times T(V)$ are interpreted as inequations (equations can be encoded via antisymmetry). We have for example the categories **DCPO**(Σ, E) and **CONT**(Σ, E). Again it can be proved that the forgetful functor has a left adjoint. The proof that the forgetful functor is monadic is also straightforward, but I have not seen it in the literature, so I present it here.

Proposition 2.4.1. *The forgetful functor* $U : \mathbf{DCPO}(\Sigma, E) \to \mathbf{DCPO}$ *is monadic.*

Proof: Let $C \underset{g}{\overset{f}{\rightrightarrows}} D$ be two morphisms in **DCPO**(Σ, E) and suppose $C \underset{g}{\overset{f}{\rightrightarrows}} D \xrightarrow{e} X$ is a split equaliser in **DCPO**. We have to define the Σ-operations on X and show that they satisfy the inequations in E. For every operation symbol *op* of arity n consider the diagram

$$\begin{tikzcd}
C^n \arrow[r, shift left, "f^n"] \arrow[r, shift right, "g^n"'] \arrow[d, "op^C"'] & D^n \arrow[r, "e^n"] \arrow[d, "op^D"'] & X^n \arrow[d, dashed] \\
C \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & D \arrow[r, "e"] & X.
\end{tikzcd}$$

Since e is an absolute coequaliser and the n-folded product is a functor in **DCPO**, e^n is also a coequaliser. Thus there exists a unique arrow $X^n \to X$ which makes the diagram commute. We take this as the definition of op^X. This also shows that e is a morphism in $\mathbf{DCPO}(\Sigma, E)$. To show that the inequations are still satisfied consider an inequation (t_1, t_2) over variables $x_1, \ldots, x_m$. The interpretations of the terms are operations of arity m.

$$\begin{tikzcd}
D^m \arrow[r, "e^m"] \arrow[d, shift right, "t_2^D"'] \arrow[d, shift left, "t_1^D"] & X^m \arrow[d, shift right, "t_2^X"'] \arrow[d, shift left, "t_1^X"] \\
D \arrow[r, "e"] & X
\end{tikzcd}$$

Note that both e and e^m are split epic. In the category **DCPO** split epics are always surjective.

Pick an m-tuple $\bar{x} \in X^m$. We want to prove that $t_1^X(\bar{x}) \sqsubseteq t_2^X(\bar{x})$. Consider a m-tuple $\bar{d} \in D^m$ such that $e^m(\bar{d}) = \bar{x}$. Notice that D satisfies the inequations, therefore $t_1^D(\bar{d})) \sqsubseteq t_2^D(\bar{d})$. Then $t_1^X(\bar{x}) = t_1^X(e^m(\bar{d})) = e(t_1^D(\bar{d})) \sqsubseteq e(t_2^D(\bar{d})) = t_2^X(e^m(\bar{d})) = t_2^X(\bar{x})$. Finally we show that e is a coequaliser in $\mathbf{DCPO}(\Sigma, E)$. Consider the diagram $C \overset{f}{\underset{g}{\rightrightarrows}} D \xrightarrow{k} Y$ where $fk = gk$ in the category $\mathbf{DCPO}(\Sigma, E)$ (that is k is a continuous homomorphism). Since e is a coequaliser in **DCPO** there is a unique continuous function $h : X \to Y$ such that $he = k$. We have to argue that h is a homomorphism too. For every operation *op* of arity n:

$$\begin{tikzcd}
X^n \arrow[r, "h^n"] \arrow[d, "op^X"'] & Y^n \arrow[d, "op^Y"] \\
X \arrow[r, "h"] & Y.
\end{tikzcd}$$

Consider

$$\begin{tikzcd}
D^n \arrow[r, "e^n"'] \arrow[rr, bend left, "k^n"] \arrow[d, "op^D"'] & X^n \arrow[r, "h^n"'] \arrow[d, "op^X"'] & Y^n \arrow[d, "op^Y"] \\
D \arrow[r, "e"] \arrow[rr, bend right, "k"'] & X \arrow[r, "h"] & Y.
\end{tikzcd}$$

Since k is a homomorphism, then $op^Y \circ k^n = k \circ op^D$. Using $he = k$ we can write $op^Y \circ h^n \circ e^n = h \circ e \circ op^D$ Since e is homomorphism, then $op^X \circ e^n = e \circ op^D$. So that $op^Y \circ h^n \circ e^n = h \circ op^X \circ e^n$. Since e^n is epic, $op^Y \circ h^n = h \circ op^X$. □

The composition of a free algebra functor F with the forgetful functor U gives rise to a monad UF. In the sequel we follow the usual convention and we will drop the mention of the forgetful functor when this does not create confusion.

2.4.3 Nondeterministic powerdomains

We present here some well known results that we use later. Consider the following equational theory over the signature $(\{\uplus\}, \alpha)$, where $\alpha(\uplus) = 2$. We call the operation represented by $\uplus$ *formal union.* (A, B, C are variables and the infix notation is used)

- $A \uplus B = B \uplus A$;
- $A \uplus (B \uplus C) = (A \uplus B) \uplus C$;
- $A \uplus A = A$.

Since $\uplus$ will always denote a commutative and associative operation, we introduce the following convention. If X is a set where $\uplus$ is defined, and $(x_i)_{i\in I}$ is a finite family of elements of X, we write

$$\biguplus_{i\in I} x_i$$

to denote the formal union of all x_i's. A similar convention will be used for the operation $\oplus$ (formal sum) which will also always be associative and commutative.

A model for the above theory is a *semilattice*. The category of semilattices is denoted by **SLAT**. It is well known that the free semilattice functor can be concretely represented as (is natural isomorphic to) the finite nonempty powerset functor P : **SET** $\rightarrow$ **SLAT** where the symbol $\uplus$ is interpreted as union. If X is a set, Z is a semilattice and $f : X \to Z$ is a function, the unique extension $\overline{f} : P(X) \to Z$ is defined by

$$\overline{f}(Y) = \biguplus_{y\in Y} f(y)\,.$$

The corresponding monad P : **SET** $\rightarrow$ **SET** has the following unit and multiplication

$$\eta^P_X(x) = \{x\};$$

$$\mu^P_X(\mathcal{S}) = \bigcup \mathcal{S}.$$

The free algebra functor for the same equational theory in the categories **DCPO** and **CONT** is known as the *Plotkin powerdomain* or *convex powerdomain* $\mathcal{P}_P$. No concrete representation of this functor for the above category is known, although it can be given for some restricted categories [AJ94]. The name "convex" comes from the fact that such representations involve order convex sets.

In the categories of domains, we can add an extra inequation to the theory. If we add

- $A \sqsubseteq A \uplus B$;

we obtain the theory of *join-semilattices.* The category of continuous join-semilattice is denoted by **JSL**. Note that a continuous join-semilattice contains the least upper bound of all nonempty set defined as follows.

$$\bigsqcup X := \bigsqcup_{Z \subseteq_{fin} X}^{\uparrow} \biguplus Z\,.$$

A join-semilattice morphism f preserves all such least upper bounds as

$$\begin{aligned} f(\bigsqcup X) = f(\bigsqcup_{Z \subseteq_{fin} X}^{\uparrow} \biguplus Z) = \bigsqcup_{Z \subseteq_{fin} X}^{\uparrow} f(\biguplus Z) \\ = \bigsqcup_{Z \subseteq_{fin} X}^{\uparrow} \biguplus f(Z) = \bigsqcup_{Y \subseteq_{fin} f(X)}^{\uparrow} \biguplus Y = \bigsqcup f(X). \end{aligned}$$

The free join-semilattice functor is known as the *Hoare powerdomain* or *lower powerdomain* $\mathcal{P}_H$. This functor has a nice concrete representation as

$$\mathcal{P}_H(\mathcal{D}) = (\{\emptyset \neq O \subseteq D \mid O \text{ Scott closed}\}, \subseteq)\}\,.$$

(We define Scott closed sets in the next section.) The symbol $\uplus$ is interpreted as binary union.

If D is a continuous domain, E is a continuous join-semilattice and $f : D \to E$ is a continuous function, the unique extension $\overline{f} : \mathcal{P}_H(D) \to E$ is defined by

$$\overline{f}(O) = \overline{\bigsqcup_{d \in O} f(d)}\,.$$

The free join-semilattice monad in **CONT** has the following unit and multiplication

$$\begin{aligned} \eta_X^{\mathcal{P}_P}(d) &= \downarrow\{d\}; \\ \mu_X^{\mathcal{P}_P}(\mathcal{S}) &= \overline{\bigcup \mathcal{S}}. \end{aligned}$$

If instead we add the inequality

- $A \uplus B \sqsubseteq A$;

we obtain the theory of *meet-semilattices.* The corresponding free algebra functor is known as the *Smyth powerdomain* or *upper powerdomain* $\mathcal{P}_S$. This functor has a concrete representation in the category **CONT**, using "compact saturated" sets.

2.5 Topology and measure theory

Formal topology arises as a purely mathematical concept, but recently it has found several applications to theoretical computer science. One of the intuitions is that open sets represent semidecidable properties and continuous functions represent computable functions. Open sets are also thought of as representing *observations.* Good discussions on these ideas can be found in [Smy83, Abr87, Vic96, Esc03] and elsewhere. Various topologies are definable on domains, the most famous one being the *Scott topology* [Sco72].

Probability theory is formally studied via the notion of *measure.* A good reference for measure theory is [Hal50]. Often, measures are defined on topologies under the name of *valuations* [Bir67, SD80, Jon90]. A good overview of the connections between measures and valuations is [AM00].

2.5.1 Topology

A *topological space* is a structure (X, τ), where $\tau \subseteq \mathcal{P}(X)$ is a family of subsets of X satisfying.

- $\emptyset, X \in \tau$;
- if $F \subseteq \tau$ then $\bigcup F \in \tau$;
- if $O_1, O_2 \in \tau$ then $O_1 \cap O_2 \in \tau$.

The family τ is a *topology* on X. The elements of τ are called *open.* If O is open, then $X \setminus O$ is called *closed.* A family $B \subseteq \mathcal{P}(X)$ is a *basis* for the topology τ if for every $O \in \tau$ there exists $C \subseteq B$ such that $O = \bigcup C$. The whole powerset $\mathcal{P}(X)$ is an example of a topology on X. It is called the *discrete* topology.

Let $(D, \sqsubseteq)$ be a DCPO. Let τ be the family of subsets of D defined as follows: $O \in \tau$ if and only if O is upward closed and for every directed set Y, if $\bigsqcup^{\uparrow} Y \in O$ then there exists $y \in Y$ such that $y \in O$.

Theorem 2.5.1 ([AJ94]). *With the above definition, (D, τ) is a topological space.*

The topology τ is called the *Scott* topology, as it was introduced by Dana Scott [Sco72]. If $d \in D$ is compact, then it is the case that $\uparrow d$ is open. In an algebraic domain, the sets of the form $\uparrow d$, with d compact, form a basis of the Scott topology.

The sets of the form $\downarrow d$ are closed for every element $d \in D$.

When X is a subset of a topological space, we define its *closure* $\overline{X}$ as the smallest closed set containing X. In continuous domains, Scott-closure can be characterised as follows.

Lemma 2.5.2. *If X is a subset of a continuous domain, then*

$$\overline{X} = \{\bigsqcup^{\uparrow} Y \mid Y \subseteq \downarrow X \text{ directed}\}\,.$$

2.5.2 Valuations on a lattice

Definition 2.5.3. A *lattice* is an algebra for the following theory in the category of partial orders.

- $A \uplus B = B \uplus A$;
- $A \uplus (B \uplus C) = (A \uplus B) \uplus C$;
- $A \uplus A = A$;
- $A \sqsubseteq A \uplus B$
- $A \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} B = B \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} A$;
- $A \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} (B \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} C) = (A \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} B) \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} C$;
- $A \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} A = A$;
- $A \mathbin{\rotatebox[origin=c]{180}{$\uplus$}} B \sqsubseteq A$.

A lattice is *distributive* if it further satisfies

- $(A \uplus B) \Cap C = (A \Cap C) \uplus (B \Cap C)$.

Definition 2.5.4. A *valuation* on a lattice with bottom X is a function $\nu : X \to \overline{\mathbb{R}^+}$ satisfying

- (Strictness)
  $\nu(\bot) = 0$;
- (Monotonicity)
  $A \sqsubseteq B \Longrightarrow \nu(A) \leq \nu(B)$;
- (Modularity)
  $\nu(A) + \nu(B) = \nu(A \uplus B) + \nu(A \Cap B)$.

The above definition originates from [Bir67], where a valuation is only required to be modular. It is well known that in lattices of sets, modularity implies the inclusion-exclusion principle [BD95]:

Proposition 2.5.5. *If $(X, \uplus, \Cap)$ is a distributive lattice, and $f : X \to \mathbb{R}$ is a modular function, then for every $n \in \mathbb{N}$,*

$$f\left(\biguplus_{i \in I_n} x_i\right) = \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} f\left(\bigcap_{i \in I} x_i\right) .$$

2.5.3 Continuous valuations on a topology

Note that the open sets of a topological space form a distributive lattice with bottom.

Definition 2.5.6. A *continuous valuation* on a topological space (X, τ) is a valuation on τ satisfying:

- (Continuity) whenever $\mathcal{J}$ is a directed subset of $(\tau, \subseteq)$
$$\nu(\bigcup \mathcal{J}) = \sup_{U \in \mathcal{J}} \nu(U) .$$

We will use Greek letters ν, ξ to denote continuous valuations. Two operations of sum and scalar product of valuations are defined pointwise:

$$\nu \oplus \xi(O) = \nu(O) + \xi(O);$$

$$p\nu(O) = p(\nu(O)), \quad p \in [0, +\infty].$$

For each $x \in X$, the function η_x such that

$$\eta_x(U) = \begin{cases} 1 & x \in U; \\ 0 & x \notin U; \end{cases}$$

is a continuous valuation and it is called *point* valuation or Dirac's delta (sometimes denoted as δ_x). A *simple* valuation is a linear combination of point valuations, that is a valuation of the form

$$\bigoplus_{x \in Y} p_x \eta_x$$

for some $Y \subseteq_{fin} X$.

2.5.4 Discrete valuations

Continuous valuations on the discrete topology deserve a special treatment.

Definition 2.5.7. A *discrete valuation* on a set X is a function $\nu : X \to \overline{\mathbb{R}^+}$.

A discrete valuation defines uniquely a continuous valuation on the discrete topology by

$$\nu[Y] = \sum_{y \in Y} \nu(y) \,.$$

The *support* of a discrete valuation ν on X is the set

$$Supp(\nu) := \{x \in X \mid \nu(x) > 0\} \,.$$

The set of discrete valuations on X is denoted by $V_\infty(X)$.

Discrete valuations taking values in $[0, +\infty[$ are called *weightings* [JLY01]. A *finite valuation* is a weighting whose support is finite.

2.5.5 Valuations as a free construction

We can characterise finite valuations as a free algebra for a suitable equational theory.

Definition 2.5.8. A *real cone* is an algebra for the following theory in the category **SET**.

1. $A \oplus B = B \oplus A$;
2. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$;
3. $A \oplus \underline{0} = A$;
4. $0A = \underline{0}$;
5. $1A = A$;
6. $p(A \oplus B) = pA \oplus pB \quad p \in \mathbb{R}^+$;
7. $p(qA) = (pq)A \quad p, q \in \mathbb{R}^+$.

13. $(p + q)A = pA \oplus qA \ p, q \in \mathbb{R}^+$.

(We will see later why we use this strange enumeration.) We call **RCONE** the category of real cones and homomorphisms.

The definition in [Var02, Kir93] is slightly different, in that it allows "scalar" values in $[0, +\infty]$. There is not much difference, though, as every real cone in the extended definition, is a real cone in the restricted sense. Vice versa if R is a real cone in the restricted sense, we can define an "extended" real cone on $R \uplus \{\infty\}$ by putting $+\infty \cdot x = \infty$.

If X is a set, the set of finite valuations over X (denoted by $V(X)$) can be endowed with a real cone structure, the operations being defined pointwise:

$$(p\nu)(x) := p \cdot \nu(x) \,;$$

$$(\nu \oplus \xi)(x) := \nu(x) + \xi(x) \,.$$

We have the following:

Proposition 2.5.9. *The finite valuations are the free real cone.*

Proof: It is easy to see that $V(X)$ is a real cone. To prove the universal property, let $f : X \to R$ be a function, where R is a real cone. We define a real cone homomorphism $\overline{f} : V(X) \to R$ which extends f. The homomorphism condition forces us to define it as follows:

$$\overline{f}(\nu) = \bigoplus_{x \in Supp(\nu)} \nu(x) f(x)\,.$$

The extension respects the sum (and $\underline{0}$) because of laws (1),(2),(3),(4),(13). It respects the scalar multiplication because of laws (6),(7). It extends f because of law (5).

The freely generated monad has the unit defined as: $\eta^V_X(x) = \eta_x$ (the Dirac's delta). The multiplication is defined as:

$$\mu^V_X(\Xi)(x) = \bigoplus_{\nu \in Supp(\Xi)} \Xi(\nu)\nu(x)\,.$$

2.5.6 The powerdomain of continuous valuations

A continuous valuation on a DCPO D is a continuous valuation on its Scott topology. The set $\mathcal{V}(D)$ of continuous valuations on D ordered pointwise is again a DCPO.

The operator $\mathcal{V}$ extends to a functor $\mathcal{V} : \mathbf{DCPO} \to \mathbf{DCPO}$. Jones showed that it is also a functor $\mathbf{CONT} \to \mathbf{CONT}$:

Theorem 2.5.10 ([Jon90]). *If D is a continuous domain, $\mathcal{V}(D)$ is a continuous domain with basis the set of simple valuations.*

The order relation on the simple valuations in $\mathcal{V}(D)$ is characterised by the following theorem.

Theorem 2.5.11 (Splitting Lemma). *Let $\nu := \sum_{b\in B} r_b\eta_b$ and $\xi := \sum_{c\in C} s_c\eta_c$ be two simple valuations. We have that $\nu \sqsubseteq \xi$ if and only if there exist "transport numbers" $t_{b,c}$ such that*

- $\sum_{c\in C} t_{b,c} = r_b$;
- $\sum_{b\in B} t_{b,c} \leq s_c$;
- $t_{b,c} > 0 \Longrightarrow b \sqsubseteq c$.

The way-below relation is characterised by the following propositions

Lemma 2.5.12. *For two simple valuations $\nu := \sum_{b\in B} r_b\eta_b$ and $\xi := \sum_{c\in C} s_c\eta_c$, we have that $\nu \ll \xi$ if and only if there exists "transport numbers" $t_{b,c}$ such that*

- $\sum_{c\in C} t_{b,c} = r_b$;
- $\sum_{b\in B} t_{b,c} \ll s_c$;
- $t_{b,c} > 0 \Longrightarrow b \ll c$.

Also,

Lemma 2.5.13 ([Kir93]). *If ξ is a continuous valuation, then $\sum_{b\in B} r_b\eta_b \ll \xi$ if and only for every nonempty $A \subseteq B$*

$$\sum_{b\in A} r_b < \xi(\uparrow A)$$

Definition 2.5.14 ([Tix99]). A *continuous d-cone* is a structure $(D, \sqsubseteq, \oplus, \odot)$ such that

- $(D, \sqsubseteq)$ is a continuous domain;
- $\oplus : D \times D \to D$ is continuous;
- $\odot : [0, +\infty[\times D \to D$ is continuous[3];
- the equations (1)–(7)(13) hold.

The corresponding category is called **CCONE**.

Alternatively we can say that a continuous d-cone is an algebra for the theory (1)–(7),(13) in the category **CONT** with *the extra requirement* that the scalar multiplication be continuous in the first argument.

Theorem 2.5.15 ([Jon90, Kir93]). *The powerdomain of valuations is the free continuous d-cone.*

Therefore the functor $\mathcal{V}$ is in fact a monad in **CONT**. The unit takes on point valuations, while an explicit definition of the multiplication requires the definition of integration [Jon90].

2.5.7 Normalised valuations

A valuation ν on a topological space (X, τ) is *normalised* if $\nu(X) = 1$. It is called *sub-normalised* if $\nu(X) \leq 1$. We write $V^1_\infty(X)$ and $V^{\leq 1}_\infty(X)$ for the set of normalised and subnormalised discrete valuations on a set X. Elements of $V^1_\infty(X)$ are also called *probability distributions* over X, while elements of $V^{\leq 1}_\infty(X)$ are also called *subprobability distributions.* They all form monads in **SET** with the multiplication defined as for V. The set of normalised valuations on a domain D is denoted by $\mathcal{V}^1(D)$. When D has a bottom element $\perp$, then $\mathcal{V}^1(D)$ has a bottom element $\eta_\perp$. The way-below relation has a different characterisation than the one in $\mathcal{V}(D)$ [Eda95a].

Lemma 2.5.16. *For two simple valuations $\nu := \sum_{b\in B} r_b\eta_b$ and $\xi := \sum_{c\in C} s_c\eta_c$ in $\mathcal{V}^1(D)$ we have that $\nu \ll \xi$ if and only if $\perp \in B$ with $r_\perp \neq 0$ and there exists "transport numbers" $t_{b,c}$ such that*

- $t_{\perp,c} \neq 0$*;*
- $\sum_{c\in C} t_{b,c} = r_b$*;*
- $\sum_{b\in B} t_{b,c} \ll s_c$*;*
- $t_{b,c} > 0 \Longrightarrow b \ll c$*.*

[3] The symbol $\odot$ is used here for clarity. Everywhere else the scalar multiplication is denoted simply by juxtaposition of its arguments.

Sub-normalised valuations are sufficient for semantics of probabilistic processes. They also are freely generated by an equational theory. Nevertheless, we choose to deal with a more general notion of valuation, because the corresponding equational theory is nicer while the other fundamental properties are the same [Kir93].

2.5.8 Measure spaces

Traditionally, probability theory has been studied via measure theory, rather than topology. We sketch here the relevant notions. A *σ-algebra* on a set Ω is a family of subsets of X which is closed under countable union and complementation and which contains $\emptyset$. The intersection of an arbitrary family of σ-algebras is again a σ-algebra. In particular if $\mathcal{S} \subseteq \mathcal{P}(\Omega)$, and $\Xi := \{\mathcal{F} \mid \mathcal{F} \text{ is a } \sigma\text{-algebra } \& \ \mathcal{S} \subseteq \mathcal{F}\}$, then $\bigcap \Xi$ is again a σ-algebra and it belongs to Ξ. We call $\bigcap \Xi$ the *smallest* σ-algebra containing $\mathcal{S}$.

If $\mathcal{S}$ is a topology, the smallest σ-algebra containing $\mathcal{S}$ is called the *Borel σ-algebra* of the topology. Note that although a topology is closed under arbitrary union, its Borel σ-algebra need not be.

A *measure space* is a triple $(\Omega, \mathcal{F}, \nu)$ where $\mathcal{F}$ is a σ-algebra on Ω and ν is a *measure* on $\mathcal{F}$ that is a function $\nu : \mathcal{F} \to \overline{\mathbb{R}^+}$ satisfying:

- (Strictness)
  $\nu(\emptyset) = 0$;
- (Countable additivity) if $(A_n)_{n\in\mathbb{N}}$ is a countable family of pairwise disjoint sets of $\mathcal{F}$, then
  $\nu(\bigcup_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} \nu(A_n)$.

Finite additivity follows by putting $A_n = \emptyset$ for all but finitely many n.

Among the various results of measure theory we state two that we will need in Chapter 7.

Theorem 2.5.17 ([Hal50] Theorem 9.E). *Let ν be a measure on a σ-algebra $\mathcal{F}$, and let A_n be a decreasing sequence of sets in $\mathcal{F}$, that is $A_{n+1} \subseteq A_n$, such that $\nu(A_0) < \infty$. Then*

$$\nu\left(\bigcap_{n\in\mathbb{N}} A_n\right) = \lim_{n\to\infty} \nu(A_n)\,.$$

One may ask when it is possible to extend a valuation on a topology to a measure on the Borel σ-algebra. This problem is discussed in Mauricio Alvarez-Manilla's thesis [AM00]. The result we need is the following. It can also be found in [AMESD00], as Corollary 4.3.

Theorem 2.5.18. *Any normalised continuous valuation on a continuous DCPO extends uniquely to a measure on the Borel σ-algebra.*

2.6 Probabilistic Automata

Probabilistic transition systems come in different flavours. The literature goes back to the probabilistic finite automata of Rabin [Rab63]. More recently, Vardi introduced the notion of labelled concurrent Markov chain [Var85] which combines probability and nondeterminism. Larsen and Skou [LS91] studied a model which is now known as the reactive model. This name was first introduced in a paper by vanGlabbek et al. [vG$^+$90] where the notions of generative models and reactive models are also studied. Desharnais et. al [BDEP97, DEP98] generalise the reactive model allowing continuous state spaces. Hansson and Jonsson [HJ89] study a model which combines probability and nondeterminism, similar to the one of Vardi. Segala and Lynch introduce the notion of probabilistic automaton, and a corresponding notion of bisimulation [SL95, Seg95]. All these notions were introduced, oblivious to the fact that in the mathematical community the notion of Markov Decision Process [Put94] had long existed. Luca de Alfaro [dA97] brought this notion to the attention of the computer science community. Other models were introduced in [PZ93]. A full overview can be found in [BSdV03], where a coalgebraic point of view allows to compare formally all different models. This is the approach we follow in the rest of this section. For the details of the theory of probabilistic automata à la Segala, we follow the overview by Mariëlle Stoelinga [Sto02].

2.6.1 Coalgebraic Definitions

A deterministic automaton on a set A of *labels* is a coalgebra $\mathcal{A}$ for the functor $-^A : \mathbf{SET} \to \mathbf{SET}$. A transition system on a set A is a coalgebra $\mathcal{A}$ for the functor $P_\perp(A \times -)$. Every deterministic automaton can be seen as a transition system via the inclusion $X^A \to P_\perp(A \times X)$. With the usual notation for transition systems we write $x \xrightarrow{a}_{\mathcal{A}} x'$ when $(a, x') \in \mathcal{A}(x)$.

A generative probabilistic transition system on A is a coalgebra for the functor $V^1_\infty(A \times -)_\perp$. A reactive probabilistic transition system on A is a coalgebra for the functor $(V^1_\infty(-)_\perp)^A$. Reactive systems are also known as Markov decision processes, in which case the elements of A are called *actions*. Reactive systems owe this name to the following intuition. Labels represent offers from the environment. At every state, *given* an offer, the process probabilistically chooses the next state. It therefore *reacts* to the offer of the environment. In generative systems, the probabilistic choice is over the next state *and the label*. Intuitively the process *generates* that label.

A simple probabilistic automaton on A is a coalgebra for the functor $P_\perp(A \times V^1_\infty(-))$. A general probabilistic automaton on A is a coalgebra for the functor $P_\perp(V^1_\infty(A \times -))$. Markov decision processes can be seen as simple probabilistic automata via the inclusion $X^A_\perp \to P_\perp(A \times X)$. Simple probabilistic automata can be seen as general probabilistic automata via the natural transformation α defined as $\alpha_X(\{\langle a, \nu \rangle\}) = \lambda \langle a, x \rangle . \nu_a(x)$. For a simple probabilistic automaton, we write $x \xrightarrow{a}_{\mathcal{A}} \nu$ when $(a, \nu) \in \mathcal{A}(x)$.

Probabilistic automata are more general than reactive systems in that, even given the offer from the environment, they can, nondeterministically, choose different probability distributions over the next states.

Deterministic automata can be seen as reactive systems via the natural transformation $\eta^{V^1_\infty} : Id_{\mathbf{SET}} \to V^1_\infty$. Transition systems can be seen as simple probabilistic automata via the same transformation.

All the above definitions can be generalised to subprobabilities by using $V^{\leq 1}_\infty$.

2.6.2 Paths and Schedulers

Often it is useful to endow the coalgebra with an *initial state* $x_0 \in X$. The behaviour of a system is usually defined using the notion of initial state.

A *path* of a nondeterministic automaton $\mathcal{A} : X \to P_\perp(A \times X)$ with initial state x_0 is a finite or infinite sequence in $(X \times A)^\infty X$, denoted as $x_0 a_1 x_1 a_2 \ldots$ such that $x_i \xrightarrow{a_{i+1}}_{\mathcal{A}} x_{i+1}$.

Traditionally the notion of path for a probabilistic automaton is obtained via the notion of a *scheduler*, also known as a *policy*.

A *path* of a simple probabilistic automaton $\mathcal{A}$ is a sequence in $(X \times A \times V^1_\infty(X))^\omega X$, denoted as $x_0, a_1, \nu_1, \ldots, x_n, a_{n+1}, \nu_{n+1}, \ldots$, such that $x_i \xrightarrow{a_i}_{\mathcal{A}} \nu_i$, and $\nu_i(x_i) > 0$. The *weight* of a path h is defined as $\Pi(h) := \prod_i \nu_i(x_{i+1})$. The paths of $\mathcal{A}$ are denoted by $\mathcal{B}(\mathcal{A})$. If $h := x_0, a_0, \nu_0, \ldots, x_n, a_n, \nu_n, x_{n+1}$, is a finite path then $l(h) := x_{n+1}$. The set of finite paths is denoted by $\mathcal{B}_{fin}(\mathcal{A})$. A path is *maximal* if it is infinite or if $\mathcal{A}(l) = \emptyset$.

A *probabilistic scheduler* is a function $\mathcal{S} : \mathcal{B}_{fin}(\mathcal{A}) \to V^1_\infty(A \times V^1_\infty(X)) \cup \{\partial\}$ such that $Supp(\mathcal{S}(h)) \subseteq \mathcal{A}(l(h))$. Here ∂ denotes the possibility that the scheduler stops the computation. A *deterministic* scheduler is a function $\mathcal{S} : \mathcal{B}_{fin}(\mathcal{A}) \to (A \times V^1_\infty(X)) \cup \{\partial\}$ such that $\mathcal{S}(h) \in \mathcal{A}(l(h))$. A deterministic scheduler can be thought of as a probabilistic scheduler choosing Dirac's deltas only. The scheduler is a device to resolve the nondeterminism present in the model. It decides taking into account the whole history of the process and it resolves both internal and external nondeterminism (when this distinction is present). What we are left with is a purely probabilistic process.

The probability of a finite path h under a probabilistic scheduler $\mathcal{S}$ is defined recursively by

$$\Pi_{\mathcal{S}}(x_0) = 1;$$

$$\Pi_{\mathcal{S}}(h a_n \nu_n x_{n+1}) = \Pi_{\mathcal{S}}(h) \cdot \mathcal{S}(h)(a_n, \nu_n) \cdot \nu_n(x_{n+1}).$$

For a deterministic scheduler $\mathcal{S}$, by $\mathcal{B}(\mathcal{A}, \mathcal{S})$ we denote the subset of $\mathcal{B}(\mathcal{A})$ whose elements $x_0 a_1 \nu_1 \ldots$ satisfy

$$(a_n, \nu_n) = \mathcal{S}(x_0 a_1 \nu_1 \ldots x_{n-1}).$$

For a deterministic scheduler $\mathcal{S}$,

$$\Pi_{\mathcal{S}}(h) = \begin{cases} \Pi(h) & \text{if } h \in \mathcal{B}(\mathcal{A}, \mathcal{S}); \\ 0 & \text{otherwise.} \end{cases}$$

We define a measure space over the set of maximal paths $maxm(\mathcal{A})$. Let $\mathcal{X}$ be the sets of *cones* generated by finite paths, that is the sets of the form $C_h := \{h' \in maxm(\mathcal{A}) \mid h \subseteq h'\}$. Let $\mathcal{F}$ be the smallest σ-algebra containing $\mathcal{X}$. Let $v_s : \mathcal{X} \to [0, 1]$ be defined as $v_{\mathcal{S}}(C_h) := \Pi_{\mathcal{S}}(h)$. It can be proved that $v_{\mathcal{S}}$ extends to a unique measure $\nu_{\mathcal{S}}$ on $\mathcal{F}$ [Seg95].

If we are interested in the labels only, we can remove states and valuations from the paths and get a probability space over the set of sequences A^ω.

A slightly different approach will be taken in Chapter 5 and in Chapter 7. There we will not use the above notions formally, but only as inspiration.

Part II

Probability and Nondeterminism

Chapter 3

Indexed Valuations

In this part we analyse combinations of the nondeterministic monad and the probabilistic monad.

There are various ways of combining two monads. If the monads arise from equational theories, we can first combine the equational theories in some way and then generate a new monad. In [HPP02] three main ways of combining theories are identified: sum, commutative combination, distributive combination. In the first case the two equational theories are combined by joining the operations and the equations, without adding new equations. In the second case, one adds equations expressing that every operation of one theory commutes with every operation in the other theory. In the third case, one adds equations expressing distributivity of every operation of one theory over every operation in the other theory. This last way can sometimes be followed more categorically using the notion of distributive law (see 2.2.7). The leading example is given by the theory of abelian groups and the theory of monoids. Their distributive combination (distributing the monoid over the group) is the theory of rings. The free ring monad can also be obtained by giving a categorical distributive law between the free abelian group monad and the free monoid monad [Bec69].

The study of the operational semantics of systems combining probability and nondeterminism suggests that, in some cases, probabilistic choice should distribute over nondeterministic choice [M^{+}94, M^{+}95]. It turns out, though, that there is no categorical distributive law between the nondeterministic monad and the probabilistic monad. Two solutions are possible at this point.

We can still form the distributive combination of the equational theories and generate a new monad. This is the path followed by Tix [Tix99, Tix00] and Mislove [Mis00], who, independently, define the notion of geometrically convex powerdomain $\mathcal{P}_{TM}$. When X is a subset of a real cone, $\mathcal{P}_{TM}(X)$ is, roughly speaking, the set of all *convex* subsets of X. The nondeterministic choice is interpreted as union followed by convex closure. We will briefly recall this construction at the end of this and of the following chapter.

The other possibility is to modify the definition of one of the monads, so as to allow the existence of a categorical distributive law. Analysing the reasons behind the failure of the distributive law, we are led to modify the probabilistic monad, defining the notion of *indexed valuation.* Mathematically, indexed valuations arise as a free algebra for an equational theory obtained from the theory of real cones by removing one equation. Besides their categorical justification, indexed valuations have a computational meaning, which we will present by giving semantics to an imperative language containing both random assignment and nondeterministic choice.

In this chapter we first show the failure of the distributive law. We then define the indexed valuations monad in the category **SET**. We show the existence of the distributive law between the indexed valuation monad and the finite nonempty powerset monad. We characterise indexed valuations as a free construction and we show that the categorical distributive law corresponds to an equational distributive law. Finally we discuss the notion of *finitely generated convex powerset*, inspired by the work of Tix and Mislove.

The notions introduced in this chapter are also necessary in the next chapter where we carry out similar constructions in the category of continuous domains.

3.1 Failure of the distributive law

Assume that $\oplus_p$ is a probabilistic choice operator: $A \oplus_p B$ is choosing A with probability p and B with probability $(1-p)$. This operator usually satisfies $A \oplus_p A = A$, because the choice between two equivalent possibilities is considered to be the same as not making any choice at all. Note that this assumes that the act of making the choice is invisible: the coin is always flipped behind one's back. Assume also that $\uplus$ represents some kind of nondeterministic choice operator: $A \uplus B$ offers to the environment the choice between A and B. Distributing one operator over the other amounts to the following law:

$$A \oplus_p (B \uplus C) = (A \oplus_p B) \uplus (A \oplus_p C) \; .$$

Intuitively this means that it is indifferent whether the environment chooses before or after the probabilistic choice is made. Clearly this is not true in all situations, but if we assume that the environment cannot see the probabilistic choice, it is at least plausible.

Once we accept the distributive law, then the extra *convexity* law [BS01]

$$A \uplus B = A \uplus B \uplus (A \oplus_p B) \uplus (B \oplus_p A)$$

must be also accepted, because

$$A \uplus B = (A \uplus B) \oplus_p (A \uplus B) = (A \oplus_p A) \uplus (B \oplus_p B) \uplus (A \oplus_p B) \uplus (B \oplus_p A).$$

If the equational distributive law corresponded to a categorical distributive law, by Theorem 2.2.3 the nondeterministic monad would lift to the category of algebras for the probabilistic monads. In the category **SET** this means that the powerset monad would lift to the category of real cones. The convexity law suggests that this is not possible as sets, in general, do not satisfy it. In fact the following theorem says that the obvious definition of the operations for the powerset cannot satisfy $A \oplus_p A = A$. Suppose we have an equational theory. Take a model X for it. We can extend every operation f of arity n to the subsets of X by

$$f(X_1, \ldots, X_n) = \{f(x_1, \ldots, x_n) \mid x_i \in X_i, i \in I_n\}.$$

Theorem 3.1.1 ([Gau57]). *A necessary and sufficient condition for the operations defined in $\mathcal{P}(X)$ to satisfy an equation of the theory is that each individual variable occurs at most once on both sides of the equation.*

The equation $A \oplus_p A = A$ does not satisfy the above requirement. This would not exclude the possibility of lifting the operations in a different way, thus obtaining another distributive law. However, it turns out that there is no distributive law at all between the two monads. If (P, η^P, μ^P) is the finite nonempty powerset monad, and (V, η^V, μ^V) is the finite valuation monad in the category **SET**, we have

Proposition 3.1.2. *There is no distributive law of V over P.*

Proof: The idea for this proof is due to Gordon Plotkin. Assume that $d : VP \overset{\cdot}{\longrightarrow} PV$ is a distributive law. Consider the set $X := \{a, b, c, d\}$. Take $\Xi := \frac{1}{2}\eta_{\{a,b\}} + \frac{1}{2}\eta_{\{c,d\}} \in VP(X)$. We try to find out what $R := d_X(\Xi)$ is.

Let $Y := \{a, b\}$. Consider:

$$f : X \to Y \quad f : \left\{ \begin{array}{ccc} a & \mapsto & a \\ b & \mapsto & b \\ c & \mapsto & a \\ d & \mapsto & b \end{array} \right.$$

$$f' : X \to Y \quad f' : \left\{ \begin{array}{ccc} a & \mapsto & a \\ b & \mapsto & b \\ c & \mapsto & b \\ d & \mapsto & a\,. \end{array} \right.$$

Consider the naturality diagram for f:

$$\begin{tikzcd}
\Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f)"'] & R \arrow[d, mapsto, "PV(f)"] \\
\eta_Y \arrow[r, mapsto, "d_Y"'] & S.
\end{tikzcd}$$

One of the unit laws for d tells us that $S := d_Y(\eta_Y) = \{\eta_a, \eta_b\}$. Therefore, considering the functorial action of PV, we must have that

$$\emptyset \neq R \subseteq \big\{p\eta_a + (1-p)\eta_c \mid p \in [0,1]\big\} \cup \big\{q\eta_b + (1-q)\eta_d \mid q \in [0,1]\big\}$$

Consider the same diagram for f':

$$\begin{tikzcd}
\Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f')"'] & R \arrow[d, mapsto, "PV(f')"] \\
\eta_Y \arrow[r, mapsto, "d_Y"'] & S.
\end{tikzcd}$$

This tells us that

$$\emptyset \neq R \subseteq \big\{p'\eta_a + (1-p')\eta_d \mid p' \in [0,1]\big\} \cup \big\{q'\eta_b + (1-q')\eta_c \mid q \in [0,1]\big\}\,.$$

Combining these pieces of information we conclude that R must be a nonempty subset of $\{\eta_a, \eta_b, \eta_c, \eta_d\}$.

Now let $Z := \{a, c\}$. Consider

$$f'' : X \to Z \quad f'' : \begin{cases} a & \mapsto & a \\ b & \mapsto & a \\ c & \mapsto & c \\ d & \mapsto & c\,. \end{cases}$$

Let us look at the naturality diagram for f'':

$$\begin{tikzcd}
\Xi \arrow[r, mapsto, "d_X"] \arrow[d, mapsto, "VP(f'')"'] & R \arrow[d, mapsto, "PV(f'')"] \\
\frac{1}{2}\eta_{\{a\}} + \frac{1}{2}\eta_{\{c\}} \arrow[r, mapsto, "d_Z"'] & T.
\end{tikzcd}$$

Since $T = PV(f'')(R)$, then T must be a nonempty subset of $\{\eta_a, \eta_c\}$. But the other unit law for d tells us that $T = d(\frac{1}{2}\eta_{\{a\}} + \frac{1}{2}\eta_{\{c\}}) = \{\frac{1}{2}\eta_a + \frac{1}{2}\eta_c\}$. Contradiction. □

A very similar argument can be applied to prove the dual.

Proposition 3.1.3. *There is no distributive law of P over V.*

Similar statements are true for the corresponding monads in the category **CONT** of continuous domains and continuous functions. If $\mathcal{P}$ is some powerdomain monad and $\mathcal{V}$ is the powerdomain of valuations monad, then there is no distributive law between them.

Our solution consists in changing the definition of probabilistic monad by removing the law $A \oplus_p A = A$. In our presentation, the probabilistic monad is generated by the theory of real cones. The probabilistic choice is defined there by $A \oplus_p B = pA \oplus (1-p)B$. We remove the equation $pA \oplus qA = (p+q)A$ from the theory of real cones. In the category **SET**, the monad freely generated by the new equational theory is called the *finite indexed valuation* monad IV. We give a concrete characterisation of this monad. By Theorem 3.1.1, we can lift the operations to the powerset, thus obtaining a distributive law. We give explicitly the definition of the distributive law between the finite nonempty powerset monad and the finite indexed valuation monad. The computational intuition of this construction will be discussed in Chapter 5.

3.2 Indexed valuations in the category of sets

In this section we present the definition of the indexed valuation monad in the category **SET**, and we show the existence of the categorical distributive law between indexed valuations and the finite nonempty powerset.

3.2.1 Definition

We first introduce the concrete characterisation of our construction and show its functoriality.

Definition 3.2.1. Let X be a set. A *discrete indexed valuation* (DIV) on X is a pair $(\mathit{Ind}, \mathit{Weight})$ where $\mathit{Ind} : I \to X$ is a function and Weight is a discrete valuation on I, for some set I.

Note that we do not require that *Ind* be injective. This is indeed the main point of this construction: we want to divide the probability of an element among its indices. One possible interpretation is that indices in I represent computations, while elements of X represent observations. The semantics we present in Chapter 5 will confirm this intuition.

We shall also write x_i for $\mathit{Ind}(i)$ and p_i for $\mathit{Weight}(i)$. A discrete indexed valuation $\xi := (\mathit{Ind}, \mathit{Weight})$ will also be denoted as $(x_i, p_i)_{i\in I}$.

We are now going to define an equivalence relation on the class of DIVs. It is the transitive closure of two simpler equivalence relations.

Definition 3.2.2. We set

$$(x_i, p_i)_{i\in I} \sim_1 (y_j, q_j)_{j\in J}$$

if and only if there exists a bijection $h : I \to J$ such that

$$\forall i \in I.\ y_{h(i)} = x_i\ ,$$

$$\forall i \in I.\ q_{h(i)} = p_i\ .$$

This says that two DIVs are equivalent up to renaming of the indices. If we interpret indices as computations, we may say that we do not care about the identity of a single computation. We only care how many different computations there are, and how they relate to observations.

Given a DIV $(x_i, p_i)_{i\in I}$, let $I_0 := \{i \in I \mid p_i = 0\}$.

Definition 3.2.3. We set

$$(x_i, p_i)_{i\in I} \sim_2 (y_j, q_j)_{j\in J}$$

if and only if

$$I \setminus I_0 = J \setminus J_0\ ,$$

$$\forall i \in I \setminus I_0.\ x_i = y_i\ \&\ p_i = q_i\ .$$

This says that only indices in the support matter. Intuitively, computations with probability 0 do not happen, so we may as well ignore them.

Definition 3.2.4. The equivalence relation $\sim$ is the transitive closure of $\sim_1 \cup \sim_2$.

¿From now on we will use the term "discrete indexed valuations" to denote equivalence classes under $\sim$.

Given a set X and an infinite cardinal number α we define the set $IV_\alpha(X)$ as follows:

$$IV_\alpha(X) := \{(x_i, p_i)_{i\in I} \mid |I| < \alpha\}/\sim\ .$$

It is easy to realise that $IV_\alpha(X)$ is indeed a set. For every cardinal number $\beta < \alpha$ choose a set I_β such that $|I_\beta| = \beta$. The class $\{I_\beta \mid \beta < \alpha\}$ is a set. And clearly $IV_\alpha(X)$ is a quotient of $\bigcup_{\beta<\alpha} X^{I_\beta} \times \overline{\mathbb{R}^+}^{I_\beta}$. In particular $IV_{\aleph_0}(X)$ is the set of discrete indexed valuations whose indexing set is finite.

Definition 3.2.5. A *finite indexed valuation* on X is an element of $IV_{\aleph_0}(X)$ for which $\mathit{Weight}(i) < +\infty$ for all indices $i \in I$. The set of finite indexed valuations on X is denoted by $IV(X)$.

The construction above can be extended to a functor $IV : \mathbf{SET} \to \mathbf{SET}$ as follows. If $f : X \to Y$ then

$$IV(f)([(x_i, p_i)_{i\in I}]_\sim) := [(f(x_i), p_i)_{i\in I}]_\sim \,.$$

It is easy to check that this construction is well defined (i.e. does not depend on the representative).

3.2.2 The monad of indexed valuations

The functor IV extends to a monad, with the following unit and multiplication (we drop the mention of equivalence classes to simplify the reading):

$$\begin{gathered}
\eta^{IV}_X : X \to IV(X)\,, \\
\eta^{IV}_X(x) := (x, 1)_{*\in\{*\}}\,; \\
\mu^{IV}_X : IV(IV(X)) \to IV(X)\,, \\
\mu^{IV}_X\,(((x_{i_\lambda}, p_{i_\lambda})_{i_\lambda\in I_\lambda}, \pi_\lambda)_{\lambda\in\Lambda}) := (x_j, q_j)_{j\in J}
\end{gathered}$$

where

$$J = \biguplus_{\lambda\in\Lambda} I_\lambda\,, \quad q_j = p_j\pi_\lambda \text{ if } j \in I_\lambda\,.$$

To simplify the definition of μ, recall that a DIV is in fact an equivalence class. We can therefore assume that $I_\lambda = I$ for every $\lambda \in \Lambda$ because we can always reindex and add indices with probability 0. Therefore

$$((x_{i_\lambda}, p_{i_\lambda})_{i_\lambda\in I_\lambda}, \pi_\lambda)_{\lambda\in\Lambda} \sim ((x^\lambda_i, p^\lambda_i)_{i\in I}, \pi_\lambda)_{\lambda\in\Lambda}\,.$$

And

$$\mu^{IV}_X\left(((x^\lambda_i, p^\lambda_i)_{i\in I}, \pi_\lambda)_{\lambda\in\Lambda}\right) := (x^\lambda_i, \pi_\lambda p^\lambda_i)_{(i,\lambda)\in I\times\Lambda}\,.$$

Proposition 3.2.6. *The triple $(IV, \eta^{IV}, \mu^{IV})$ defined above is a monad.*

Proof: It is easy to check that η, μ are well defined and are natural transformations. Let us now check the diagrams for the monad laws:

1.

$$\begin{tikzcd}
IV(X) \arrow[d, "IV(\eta_X)"'] \arrow[dr, "Id_{IV(X)}"] & \\
IV^2(X) \arrow[r, "\mu_X"'] & IV(X)
\end{tikzcd}$$

$$\begin{gathered}
IV(\eta_X)\big((x_i, p_i)_{i\in I}\big) = ((x_i, 1)_{*\in\{*\}}, p_i)_{i\in I}; \\
\mu_X\big(((x_i, 1)_{*\in\{*\}}, p_i)_{i\in I}\big) = (x_i, 1p_i)_{(i,*)\in I\times\{*\}} \sim (x_i, p_i)_{i\in I}\,.
\end{gathered}$$

2.

$$\begin{tikzcd}
 & IV(X) \arrow[dl, "Id_{IV(X)}"'] \arrow[d, "\eta_{IV(X)}"] \\
IV(X) & IV^2(X) \arrow[l, "\mu_X"]
\end{tikzcd}$$

$$\eta_{IV(X)}\big((x_i,p_i)_{i\in I}\big) = ((x_i,p_i)_{i\in I},1)_{*\in\{*\}};$$

$$\mu_X\big(((x_i,p_i)_{i\in I},1)_{*\in\{*\}}\big) = (x_i,1p_i)_{(*,i)\in\{*\}\times I} \sim (x_i,p_i)_{i\in I}\ .$$

3.

$$\begin{tikzcd}
IV^3(X) \arrow[r, "\mu_{IV(X)}"] \arrow[d, "IV(\mu_X)"'] & IV^2(X) \arrow[d, "\mu_X"] \\
IV^2(X) \arrow[r, "\mu_X"'] & IV(X)
\end{tikzcd}$$

$$\begin{aligned}
&\mu_X\circ IV(\mu_X)\left((((x_i^{j,l},p_i^{j,l})_{i\in I},r_j^l)_{j\in J},q_l)_{l\in L}\right) = \\
&\quad = \ \mu_X\left(((x_i^{j,l},p_i^{j,l}r_j^l)_{(i,j)\in I\times J},q_l)_{l\in L}\right)\\
&\quad = \ (x_i^{j,l},p_i^{j,l}r_j^l q_l)_{(i,j,l)\in I\times J\times L}\\
&\quad = \ \mu_X\left(((x_i^{j,l},p_i^{j,l})_{i\in I},r_j^l q_l)_{(j,l)\in J\times L}\right)\\
&\quad = \ \mu_X\circ\mu_{IV(X)}\left((((x_i^{j,l},p_i^{j,l})_{i\in I},r_j^l)_{j\in J},q_l)_{l\in L}\right)\ .
\end{aligned}$$

Note that we make essential use of the fact that the exact identity of the indices does not matter. □

3.2.3 The distributive law

We now define the categorical distributive law between the indexed valuation monad IV and the nonempty finite powerset monad P. Recall that the monad on P is defined as follows:

$$\eta^P_X : X \to P(X)\,,$$

$$\eta^P(x) = \{x\}\,;$$

$$\mu^P_X : P(P(X)) \to P(X)\,,$$

$$\mu^P_X(\mathcal{S}) = \bigcup \mathcal{S}\,.$$

For every set X define the function $d_X : IV(P(X)) \to P(IV(X))$ as follows:

$$d_X\left((S_i,p_i)_{i\in I}\right) = \{(h(i),p_i)_{i\in I} \mid h : I \to X,\ h(i)\in S_i\}$$

We first note (omitting the easy proof) that the definition of d_X does not depend on the representative.

Theorem 3.2.7. *Let IV : **SET** → **SET** be as above, and P : **SET** → **SET** be the covariant nonempty finite powerset monad. Then the family of functions $(d_X)_{X\in\mathbf{SET}}$ defines a distributive law*

$$d : IV\circ P \to P\circ IV\,.$$

Proof: First we have to show that d is a natural transformation.

$$\begin{tikzcd}
X \arrow[d, "f"'] & IV(P(X)) \arrow[r, "d_X"] \arrow[d, "IV(P(f))"'] & P(IV(X)) \arrow[d, "P(IV(f))"] \\
Y & IV(P(Y)) \arrow[r, "d_Y"] & P(IV(Y))
\end{tikzcd}$$

Take a function $f : X \to Y$. Take $\Xi \in IV(P(X))$, $\Xi = (S_i, p_i)_{i\in I}$. We have $IV(P(f))(\Xi) = (f(S_i), p_i)_{i\in I}$. Then

$$d_Y(f(S_i), p_i)_{i\in I} = \big\{(h'(i), p_i)_{i\in I} \mid h' : I \to Y,\ h'(i) \in f(S_i)\big\} =: A\,.$$

On the other hand consider

$$P(IV(f))\Big(\big\{(h(i), p_i)_{i\in I} \mid h : I \to X,\ h(i) \in S_i\big\}\Big)\,.$$

This is equal to

$$\big\{(f(h(i)), p_i)_{i\in I} \mid h : I \to X,\ h(i) \in S_i\big\} =: B\ .$$

We have to show that $A = B$. Clearly $B \subseteq A$, by setting $h' = f \circ h$. Take now an element $(h'(i), p_i)_{i\in I}$ of A. This means that $h'(i) = f(x_i)$ for some $x_i \in S_i$. For every S_i, we select one such x_i and then we define $h(i) = x_i$. Thus we have

$$(f(h(i)), p_i)_{i\in I} = (h'(i), p_i)_{i\in I}$$

so that $(h'(i), p_i)_{i\in I}$ belongs to B.

Now we have to check the four diagrams characterising the distributive law:

1.

$$\begin{tikzcd}
& P \arrow[dl, "\eta^{IV}P"'] \arrow[dr, "P\eta^{IV}"] & \\
IVP \arrow[rr, "d"'] & & PIV
\end{tikzcd}$$

$$\begin{aligned}
d_X\left((S,1)_{*\in\{*\}I}\right) &= \big\{(h(*), 1)_{*\in\{*\}} \mid h : \{*\} \to X,\ h(*) \in S\big\} \\
&= \big\{(x, 1)_{*\in\{*\}} \mid x \in S\big\}\,.
\end{aligned}$$

2.

$$\begin{tikzcd}
& IV \arrow[dl, "IV\eta^P"'] \arrow[dr, "\eta^P IV"] & \\
IVP \arrow[rr, "d"'] & & PIV
\end{tikzcd}$$

$$\begin{aligned}
d_X\big((\{x_i\}, p_i)_{i\in I}\big) &= \big\{(h(i), p_i)_{i\in I} \mid h : I \to X,\ h(i) \in \{x_i\}\big\} \\
&= \big\{(x_i, p_i)_{i\in I}\big\}\,.
\end{aligned}$$

3.

$$\begin{tikzcd}
IVIVP \arrow[r, "IVd"] \arrow[d, "\mu^{IV}P"'] & IVPIV \arrow[r, "dIV"] & PIVIV \arrow[d, "P\mu^{IV}"] \\
IVP \arrow[rr, "d"'] & & PIV
\end{tikzcd}$$

Let $\mathcal{X} := (\Xi_\lambda, \pi_\lambda)_{\lambda\in\Lambda} \in IV(IV(P(X)))$, where $\Xi_\lambda := (S_{i_\lambda}, p_{i_\lambda})_{i_\lambda \in I_\lambda}$. As before we can assume that $I_\lambda = I$ for every $\lambda \in \Lambda$. Therefore $\Xi_\lambda = (S_i^\lambda, p_i^\lambda)_{i\in I}$. We have that

$$\mu^{IV}_{P(X)}(\mathcal{X}) = (S_i^\lambda, p_i^\lambda \pi_\lambda)_{(i,\lambda)\in I\times\Lambda}\,.$$

If we apply d_X to this term we get

$$\big\{(h(i,\lambda), p_i^\lambda \pi_\lambda)_{(i,\lambda)\in I\times\Lambda} \mid h : I\times\Lambda \to X,\ h(i,\lambda) \in S_i^\lambda\big\} := A\,.$$

Consider now $IV(d_X)(\mathcal{X})$. It is

$$(d_X(\Xi_\lambda), \pi_\lambda)_{\lambda\in\Lambda}$$

where

$$d_X(\Xi_\lambda) = \big\{(h^\lambda(i), p_i^\lambda)_{i\in I} \mid h^\lambda : I \to X,\ h^\lambda(i) \in S_i^\lambda\big\}\,.$$

Now apply $d_{IV(X)}$. We get

$$\big\{(H(\lambda), \pi_\lambda)_{\lambda\in\Lambda}\ | H : \Lambda \to IV(X),\ H(\lambda) \in d_X(\Xi_\lambda)\big\} := B\,.$$

The function H is choosing an element in $d_X(\Xi_\lambda)$. We can think of H as choosing a function $h^\lambda : I \to X, h^\lambda(i) \in S_i^\lambda$. Therefore we can equivalently define B as follows:

$$B = \big\{((H(\lambda)(i), p_i^\lambda)_{i\in I}, \pi_\lambda)_{\lambda\in\Lambda}\ | H : \Lambda \to (I \to X),\ H(\lambda)(i) \in S_i^\lambda\big\}\,.$$

Now we have to show that the flattening (through μ^{IV}) of every valuation in B gives a valuation in A, and that every valuation in A can be obtained by flattening a valuation in B. We have

$$\mu^{IV}_X\big(((H(\lambda)(i), p_i^\lambda)_{i\in I}, \pi_\lambda)_{\lambda\in\Lambda}\big) \quad = \quad (H(\lambda)(i), p_i^\lambda \pi_\lambda)_{i\in I}\,.$$

Now it is enough to observe that "uncurrying" H we get an $h : I\times\Lambda \to X$, satisfying $h(i,\lambda) \in S_i^\lambda$. So $P(\mu^{IV}_X)(B) \subseteq A$. The other inclusion is obtained by "currying" h to get H.

4.

$$\begin{tikzcd}
IVPP \arrow[r, "dP"] \arrow[d, "IV\mu^P"'] & PIVP \arrow[r, "Pd"] & PPIV \arrow[d, "\mu^P IV"] \\
IVP \arrow[rr, "d"'] & & PIV
\end{tikzcd}$$

Remember that $\mu^P_X(\mathcal{S}) = \bigcup \mathcal{S}$. Let $\mathcal{X} := (\mathcal{S}_i, p_i)_{i\in I} \in IV(P(P(X)))$. We have that

$$IV(\mu^P_X)(\mathcal{X}) = (\bigcup \mathcal{S}_i, p_i)_{i\in I}\,.$$

If we apply d_X to this term we get

$$\big\{(h(i), p_i)_{i\in I} \mid h : I \to X,\ h(i) \in \bigcup \mathcal{S}_i\big\} := A\,.$$

Consider now $d_{P(X)}(\mathcal{X})$. It is

$$\big\{(h'(i), p_i)_{i\in I} \mid h' : I \to P(X),\ h'(i) \in \mathcal{S}_i\big\} := D\,.$$

The function h' is choosing a set in $\mathcal{S}_i$ for every i. Now two steps in one. First step: we apply $P(d_X)$ to D and we obtain a set C of sets of valuations. Second step: we flatten C to a set B of valuations defined as:

$$\big\{(h''(i), p_i)_{i\in I} \mid h'' : I \to X,\ h''(i) \in h'(i),\ h' : I \to P(X),\ h'(i) \in \mathcal{S}_i\big\}\,.$$

We claim that $A = B$. Clearly $B \subseteq A$ because $h''(i) \in \bigcup \mathcal{S}_i$. But also $A \subseteq B$. We build h' as follows: for every i we choose $S_i \in \mathcal{S}_i$ such that $h(i) \in S_i$. Then $h'' = h$ does the job.

□

3.3 Equational characterisation

In this section we characterise the monad IV as a free construction and we show the correspondence between categorical and equational distributive laws.

3.3.1 Real quasi-cones

We define two operations on discrete indexed valuations.

Definition 3.3.1. Let $\nu := (\mathit{Ind}, \mathit{Weight}) = (x_i, p_i)_{i\in I}$, $\xi := (\mathit{Ind}', \mathit{Weight}') = (y_j, q_j)_{j\in J}$ be DIVs on X. Assume that $I \cap J = \emptyset$ (this is not restrictive, because we can always reindex).

We define $\nu \oplus \xi$ to be $(\mathit{Ind} \cup \mathit{Ind}', \mathit{Weight} \cup \mathit{Weight}')$. For $p \in \mathbb{R}^+$ we define $p\nu$ to be $(x_i, pp_i)_{i\in I}$. With $\underline{0}$ we denote the DIV whose indexing set is empty.

Note, in particular, that when $p \neq 0, 1$, $p\nu \oplus (1-p)\nu \not\sim \nu$, because the indexing sets do not have the same cardinality.

Consider the following equational theory:

1. $A \oplus B = B \oplus A$;
2. $A \oplus (B \oplus C) = (A \oplus B) \oplus C$;
3. $A \oplus \underline{0} = A$;
4. $0A = \underline{0}$;
5. $1A = A$;
6. $p(A \oplus B) = pA \oplus pB \quad p \in \mathbb{R}^+$;
7. $p(qA) = (pq)A \quad p, q \in \mathbb{R}^+$.

These axioms are almost the ones defining a *real cone* (see Section 2.5). The only difference is that we drop the axiom $(p+q)A = (pA \oplus qA)$.

Definition 3.3.2. A *real quasi-cone* is an algebra for the equational theory (1)–(7) in the category **SET**.

Proposition 3.3.3. *The finite indexed valuations are the free real quasi-cone.*

Proof: For any set X, it is clear that $IV(X)$ with the operations defined above is a quasi-cone. Let Q be a quasi-cone and let $f : X \to Q$ a function. We have to show that there is a unique quasi-cone homomorphism $\overline{f} : IV(X) \to Q$ such that $\overline{f}(x,1) = f(x)$. The homomorphism condition forces us to define

$$\overline{f}(x_i, p_i)_{i\in I} = \bigoplus_{i\in I} p_i f(x_i)\,.$$

Associativity, commutativity, and the two $\underline{0}$-laws guarantee that the definition does not depend on the representative for $(x_i, p_i)_{i\in I}$. The unit law guarantees that $\overline{f}(x,1) = f(x)$. The homomorphism condition for the sum (and $\underline{0}$) is obvious, while for the scalar product we have to use the laws (6) and (7). □

3.3.2 The distributive law, equationally

Recall that a semilattice is a model of the following theory.

8. $A \uplus B = B \uplus A$;
9. $A \uplus (B \uplus C) = (A \uplus B) \uplus C$;
10. $A \uplus A = A$.

We have seen in chapter 2 that the finite nonempty powerset is the free semilattice.

Consider now the combined equational theory (1)–(10) augmented with the following axioms.

11. $p(A \uplus B) = pA \uplus pB$;
12. $A \oplus (B \uplus C) = (A \oplus B) \uplus (A \oplus C)$.

Equations (11)–(12) express that the probabilistic operators distribute over the nondeterministic one.

Theorem 3.3.4. *The monad on $P \circ IV$ obtained via the categorical distributive law defined above is the free algebra for the equational theory (1)–(12).*

Proof: First we show that $P \circ IV$ is left adjoint to the forgetful functor. Let's start by observing that $P(IV(X))$ is indeed a model of (1)–(12), where $\uplus$ is interpreted as union, addition and scalar multiplication are the standard extensions to subsets of the corresponding operations in $IV(X)$, and $\underline{0}$ is the singleton of the empty indexed valuation. Now let Q be a model of (1)–(12), and let $f : X \to Q$ be a function. We have to show that there is a unique function

$\overline{f} : P(IV(X)) \to Q$ which respects the operations and such that $\overline{f}\{(x,1)\} = f(x)$. The homomorphism condition forces us to define

$$\overline{f}\{(x_i, p_i)_{i\in I}\} = \bigoplus_{i\in I} p_i f(x_i)\,;$$

$$\overline{f}(A) = \bigcup_{\nu\in A} \overline{f}(\{\nu\})\,.$$

Laws (1)–(4) again guarantee that the definition in the first line does not depend on the representative for $(x_i, p_i)_{i\in I}$. Laws (8),(9) guarantee that the second line is well defined. Law (5) guarantees that $\overline{f}\{(x,1)\} = f(x)$. The function respects the sum (and $\underline{0}$) because of law (12). It respects the product because of laws (6),(7),(11). It respects the union because of law (10).

Note that the unit of the adjunction (which is also the unit of the corresponding monad) is just $\eta_P \eta_{IV}$

We have to show that the monad generated by this adjunction is the same as the monad generated by the distributive law. The functor and the unit are the same. Instead of showing that the multiplication is the same, we equivalently show that the Kleisli extension operators are the same.

Let $f : X \to P(IV(Y))$ be a function. Consider a finite set of finite indexed valuations $A \in P(IV(X))$. Since A is finite it is not restrictive to assume that all its elements are indexed by the same set I. So we can write

$$A = \left\{(x_i^\rho, p_i^\rho)_{i\in I} \mid \rho \in R\right\},$$

with the convention that for two different ρ, ρ' the corresponding indexed valuations are different. Analogously, we write

$$f(x_i^\rho) = \left\{(y_j^{\sigma^{i,\rho}}, q_j^{\sigma^{i,\rho}})_{j\in J} \mid \sigma^{i,\rho} \in S^{i,\rho}\right\},$$

with a similar convention as above for any fixed (i,ρ), and also assuming that the $S^{i,\rho}$ are all disjoint. Again it is not restrictive to assume that all the valuations are indexed by the same set J.

We want to evaluate $\overline{f}(A)$, the Kleisli extension of the monad generated by the universal property:

$$\overline{f}\Big(\left\{(x_i^\rho, p_i^\rho)_{i\in I} \mid \rho \in R\right\}\Big) = \bigcup_{\rho\in R} \bigoplus_{i\in I} p_i^\rho f(x_i^\rho)\,.$$

Now it can be proved by induction on the size of I that

$$\begin{aligned}
&\bigoplus_{i\in I} p_i^\rho \left\{(y_j^{\sigma^{i,\rho}}, q_j^{\sigma^{i,\rho}})_{j\in J} \mid \sigma^{i,\rho} \in S^{i,\rho}\right\} \\
&\quad = \quad \Big\{(y_j^{k^\rho(i)}, p_i^\rho q_j^{k^\rho(i)})_{(j,i)\in J\times I} \mid k^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ k^\rho(i) \in S^{i,\rho}\Big\}\,.
\end{aligned}$$

Therefore:

$$\begin{aligned}
&\overline{f}\Big(\left\{(x_i^\rho, p_i^\rho)_{i\in I} \mid \rho \in R\right\}\Big) \\
&\quad = \quad \Big\{(y_j^{k^\rho(i)}, p_i^\rho q_j^{k^\rho(i)})_{(j,i)\in J\times I} \mid k^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ k^\rho(i) \in S^{i,\rho}, \rho \in R\Big\}\,.
\end{aligned}$$

Let's now look at $f^\dagger(A)$, the Kleisli extension of the monad obtained via the distributive law.

$$\begin{aligned}
&f^\dagger\Big(\big\{(x_i^\rho, p_i^\rho)_{i\in I} \,|\, \rho \in R\big\}\Big)\\
&\quad= \quad \mu^P \mu^{IV} \circ PdIV\Big(\big\{(f(x_i^\rho), p_i^\rho)_{i\in I} \,|\, \rho \in R\big\}\Big)\\
&\quad= \quad \mu^P \mu^{IV}\Big(\Big\{\big\{(h^\rho(i), p_i^\rho)_{i\in I} \,|\, h^\rho : I \to IV(Y), h^\rho(i) \in f(x_i^\rho)\big\} | \rho \in R\Big\}\Big)
\end{aligned}$$

By the conventions we have assumed, choosing an element in $f(x_i^\rho)$ is the same as choosing a $\sigma^{i,\rho} \in S^{i,\rho}$, therefore it is equivalent to think of h^ρ as a function $h^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ h^\rho(i) \in S^{i,\rho}$. Then we can continue the chain of equalities

$$= \mu^P \mu^{IV}\Big(\Big\{\big\{((y_j^{h^\rho(i)}, q_j^{h^\rho(i)})_{j\in J}, p_i^\rho)_{i\in I} \,|\, h^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ k^\rho(i) \in S^{i,\rho}\big\} | \rho \in R\Big\}\Big)$$

$$= \mu^P\Big(\Big\{\big\{(y_j^{h^\rho(i)}, p_i^\rho q_j^{h^\rho(i)})_{(j,i)\in J\times I} \,|\, h^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ h^\rho(i) \in S^{i,\rho}\big\} \,|\, \rho \in R\Big\}\Big)$$

$$= \Big\{(y_j^{h^\rho(i)}, p_i^\rho q_j^{h^\rho(i)})_{(j,i)\in J\times I} \,|\, h^\rho : I \to \bigcup_{i\in I} S^{i,\rho},\ h^\rho(i) \in S^{i,\rho}, \rho \in R\Big\}.$$

□

We can see finite indexed valuations and finite sets as (equivalence classes of) terms. In this way we can give a syntactic interpretation of the categorical distributive law: it takes a term where there are no probabilistic operators inside a nondeterministic one and transforms it into a term where all the nondeterministic operators have been pushed outside. In other words we can interpret the equations (11)–(12) as rewriting rules, from left to right.

3.4 The convex powerset

Another solution for combining the nondeterministic and probabilistic monad consists in forming the distributive combinations of the theories thus freely generating a new monad. The convexity law suggests a way of representing this construction concretely. This section is inspired by the work of Tix and Mislove, although they are only concerned with DCPOs, while we work here in the category **SET**.

3.4.1 Finitely generated convex sets

Recall that a *real cone* is a real quasi-cone satisfying the extra axiom

13. $(p + q)A = pA \oplus qA$.

Definition 3.4.1. A subset X of a real cone is *convex* if for every $x, y \in X, p \in [0, 1]$, we have $px \oplus (1 - p)y \in X$. Given a set X, its *convex closure* $\overline{X}$ is the smallest convex set containing X. A convex set X is *finitely generated* if there exists a finite set X_0 such that $X = \overline{X_0}$. Given a finite set I, elements $x_i, i \in I$ of a real cone and nonnegative real numbers $p_i, i \in I$ such that $\sum_{i\in I} p_i = 1$, the element $\bigoplus_{i\in I} p_i x_i$ is said to be a *convex combination* of the x_i.

The following result is standard.

Proposition 3.4.2. *For a set X, we have that $\overline{X}$ is the set of convex combinations of elements of X.*

Definition 3.4.3. For a real cone Z we define

$$P_{TM}(Z) = \{Y \subseteq Z \,|\, Y \text{ convex, finitely generated}\} .$$

We define

- $pY = \{py \,|\, y \in Y\}$;
- $Y \oplus Y' = \{y \oplus y' \,|\, y \in Y, y' \in Y'\}$;
- $\underline{0} = \{\underline{0}\}$;
- $Y \uplus Y' = \overline{Y \cup Y'} = \{py \oplus (1-p)y' \,|\, p \in [0,1],\, y \in Y, y' \in Y'\}$.

3.4.2 Equational characterisation

We characterise the functor P_{TM} as a free construction.

Definition 3.4.4. A *real cone-semilattice* is a model for the theory (1)–(13). The corresponding category is called **RCS**.

Proposition 3.4.5. *The operator P_{TM} with the operations as above defines a functor* **RCONE** $\rightarrow$ **RCS** *which is left adjoint of the forgetful functor.*

Proof: First we have to show that the operations are well defined and satisfy the axioms. If Y, Y' are convex, it is easy to show that $pY, Y \oplus Y', Y \uplus Y$ are convex. If Y_0, Y_0' are finite generators for Y, Y' then pY_0 is a finite generator for pY, $Y_0 \oplus Y_0'$ is a finite generator for $Y \oplus Y'$ and $Y_0 \cup Y_0'$ is a finite generator for $Y \uplus Y'$. As for the axioms the only nontrivial ones are (12)-(13): here is where convexity is needed.

Then we have to show the universal property characterising freeness. For every real cone Z and real cone-semilattice H and real cone homomorphism $f : Z \rightarrow H$, there exists a unique **RCS**-morphism $\overline{f} : P_{TM}(Z) \rightarrow H$ such that $\overline{f}(\{z\}) = f(z)$. Now for every $Y \in P_{TM}(Z)$ let Y_0 be one of its finite generators, then

$$\overline{f}(Y) = \biguplus_{y \in Y_0} f(y) .$$

The homomorphism condition implies uniqueness. We have to show that this function is well defined and that it is indeed a homomorphism. First we need to show that the definition does not depend on the chosen finite generator.

Lemma 3.4.6. *Let H be a real cone-semilattice, let Y_0, Z_0 be finite subsets of H. If $\overline{Y_0} = \overline{Z_0}$, then $\biguplus Y_0 = \biguplus Z_0$*

Proof: We prove this for the simple case where $Y_0 = \{y, y'\}, Z_0 = \{z, z'\}$. The general case can be proved in a similar way. We want to prove that $y \uplus y' = z \uplus z'$. We will prove that $y \uplus y' = y \uplus y' \uplus z \uplus z'$ (which, by symmetry, implies our result). Note that, from the assumption, z, z' must be convex combinations of y, y'. The statement is thus a consequence of the following proposition.

Proposition 3.4.7. *In a real cone-semilattice, if w is a convex combination of y, y' then*

$$y \uplus y' = y \uplus y' \uplus w\,.$$

Proof: Let $w = py \oplus (1-p)y'$. Then

$$\begin{aligned} y \uplus y' &= p(y \uplus y') \oplus (1-p)(y \uplus y') \\ &= y \uplus y' \uplus (py \oplus (1-p)y') \uplus (py' \oplus (1-p)y)\,. \end{aligned}$$

The statement of the proposition follows from

Lemma 3.4.8. *In a semilattice, if $x = x \uplus x' \uplus x''$, then $x = x \uplus x'$.*

Proof:

$$x \uplus x' = x \uplus x' \uplus x'' \uplus x' = x \uplus x' \uplus x'' = x\,.$$

□

Finally it is easy to verify that $\overline{f}$ respects the operation, using the distributive law (11-12), and the fact that f is already a homomorphism of real cones. □

The combination of the two adjunctions

$$\begin{tikzcd} \mathbf{SET} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{RCONE} \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathbf{RCS} \arrow[l, shift left] \end{tikzcd}$$

gives rise to a monad in **SET**.

Note that the the monad P_{TM} on **RCONE** is not a lifting of the monad P, because, in general, convex sets are not finite. Therefore the monad $P_{TM} \circ V$ on **SET** is not obtained by any distributive law $V \circ P \to P \circ V$.

3.4.3 The Kleisli extension

Let's look concretely at the Kleisli extension of the monad $P_{TM} \circ V$.

Take $f : X \to P_{TM}(V(Y))$, say $f(x) = B_x$. We have that $f^\dagger : P_{TM}(V(X)) \to P_{TM}(V(Y))$ is defined as

$$f^\dagger(A) = \biguplus_{\xi \in A_0} \sum_{x \in X} \xi(x) B_x\,.$$

In chapter 5 we will need the following proposition.

Proposition 3.4.9.

$$f^\dagger(A) = \bigcup_{\xi \in A} \bigoplus_{x \in X} \xi(x) B_x = \left\{ \bigoplus_{x \in X} \xi(x) h(x) \,|\, h : X \to V(Y), h(x) \in B_x,\, \xi \in A \right\}.$$

Proof: Let's call

- $V := \bigcup_{\xi \in A_0} \bigoplus_{x \in X} \xi(x) B_x$;
- $U := \biguplus_{\xi \in A_0} \bigoplus_{x \in X} \xi(x) B_x$;
- $W := \bigcup_{\xi \in A} \bigoplus_{x \in X} \xi(x) B_x$.

Remember that $U = \overline{V}$.

Clearly $V \subseteq W$. Moreover W is convex:

$$p \bigoplus_{x \in X} \xi(x)h(x) \oplus (1-p) \bigoplus_{x \in X} \xi'(x)h'(x)$$

$$= \bigoplus_{x \in X} p\xi(x)h(x) \oplus (1-p)\xi'(x)h'(x)\,.$$

Define $\xi'' = p\xi \oplus (1-p)\xi' \in A$, and $h''(x) = \frac{p\xi(x)}{\xi''(x)}h(x) + \frac{(1-p)\xi'(x)}{\xi''(x)}h'(x)$. Since B_x is convex , then $h''(x) \in B_x$. (If $\xi''(x) = 0$ then $h''(x)$ can be set equal to any element of B_x.) We have

$$\xi''(x)h''(x) = p\xi(x)h(x) \oplus (1-p)\xi'(x)h'(x)\,.$$

Therefore $U \subseteq W$.

For the other direction take $\bigoplus_{x \in X} \xi(x)h(x)$. We know that $\xi = \bigoplus_{i \in I} p_i\xi_i$ with $\xi_i \in A_0$. So

$$\bigoplus_{x \in X} \xi(x)h(x) = \bigoplus_{x \in X} \bigoplus_{i \in I} p_i\xi_i(x)h(x) = \bigoplus_{i \in I} p_i \bigoplus_{x \in X} \xi_i(x)h(x)$$

which is a convex combination of elements of V. □

Chapter 4

Indexed Valuations and Domains

In this chapter we define various notions of indexed valuations on continuous domains. We make use of the theory of abstract bases seen in 2.3.3. All constructions we perform start by defining an AB-relation on the set of finite indexed valuations, and then consider its ideal completion.

This allows us to characterise equationally all our constructions. In the category **CONT**, we have three choices for modifying the theory of real cones: we can remove the equation $pA \oplus qA = (p+q)A$ completely, or we can substitute an inequation for it. Our first choice, the reason for which we discuss later, is to substitute the inequation $pA \oplus qA \sqsubseteq (p+q)A$. The monad freely generated by such inequational theory is called *Hoare indexed valuation* monad $\mathcal{IV}$.

We show the relations between Hoare indexed valuations and continuous valuations. Specifically we show the existence of a insertion-closure pair between them.

We define a categorical distributive law between Hoare indexed valuations and the Hoare powerdomain and show the correspondence with the equational distributive law. We discuss why we were not able to define analogous distributive laws involving the Plotkin powerdomain or the Smyth powerdomain.

We discuss the other two choices of an equational theory. Removing the equation $pA \oplus qA = (p+q)A$ completely gives rise to the *Plotkin indexed valuations*, while substituting for it the inequation $pA \oplus qA \sqsupseteq (p+q)A$ gives rise to the *Smyth indexed valuations.* We briefly study them and their relation with the powerdomains. In particular we show that there is no insertion-closure pair between either of them and continuous valuations.

Unfortunately we are not able to provide concrete characterisations of these constructions yet, and we will discuss the difficulties we have encountered. This discussion leads us to propose another definition of indexed valuations that may overcome some of the problems.

Finally we briefly present the convex powerdomain of Tix and Mislove.

4.1 Hoare indexed valuations

In this section we define the Hoare indexed valuations as ideal completion of an abstract basis (see 2.3.3).

The definition of the abstract basis is inspired by the Splitting Lemma 2.5.11.

4.1.1 The AB-relation

Let $(X, \lhd)$ be an abstract basis. We define a relation $\prec$ on $IV(X)$ in such a way that $(IV(X), \prec)$ is an abstract basis:

Definition 4.1.1. For $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \in IV(X)$

$$(x_i, p_i)_{i\in I} \prec (y_j, q_j)_{j\in J}$$

if and only if there exists a partial surjective function $f : J \to I$, such that

$$x_{f(j)} \lhd y_j\,,$$

$$p_i \ll \sum_{f(j)=i} q_j\,.$$

We call such an f a *witness* for the relation.

The above definition uses representatives of equivalence classes. It should be read as: "$\nu \prec \xi$ if there is a representative $(x_i, p_i)_{i\in I}$ of ν, a representative $(y_j, q_j)_{j\in J}$ of ξ and a witness f". We could therefore restrict to *total* witnesses: suppose $\nu \prec \xi$ and f is a witness on the representatives $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J}$. If f is partial, consider the set $J_0 \subseteq J$ where f is not defined. For every $j \in J_0$ let $z_j \in X$ be such that $z_j \lhd y_j$. Such elements exist because of the interpolation property instantiated with $|F| = 0$. Let $K := I \uplus J_0$ and for every $k \in K$ let

$$p'_k := \begin{cases} p_k & \text{if } k \in I \\ 0 & \text{if } k \in J_0, \end{cases}$$

$$x'_k := \begin{cases} x_k & \text{if } k \in I \\ z_k & \text{if } k \in J_0. \end{cases}$$

Clearly $(x_i, p_i)_{i\in I} \sim (x'_k, p'_k)_{k\in K}$, because we added only indices with weight 0. Now define $f' : J \to K$ by

$$f'(j) := \begin{cases} f(j) & \text{if } j \notin J_0 \\ j & \text{if } j \in J_0. \end{cases}$$

It is easy to see that f' satisfies the conditions for being a witness, and moreover it is total.

We choose to deal with partial functions as witnesses, because this gives us more versatility.

There are three reasons for Definition 4.1.1. Firstly, this definition is an "indexed" version of the splitting lemma (2.5.11). Secondly, it has an interesting computational interpretation. We will discuss it in Section 4.7, and in Section 5.4, where we will make use of the notion of scheduler for a probabilistic and nondeterministic operational model (see Section2.6). Finally, this definition corresponds to an inequational theory that allows us to match equational and categorical distributive laws, as we did in the category of sets.

Proposition 4.1.2. $(IV(X), \prec)$ *is an abstract basis.*

Proof: To show transitivity we show that if f is a witness for $(x_i, p_i)_{i\in I} \prec (y_j, q_j)_{j\in J}$ and g is a witness for $(y_j, q_j)_{j\in J} \prec (z_l, r_l)_{l\in L}$ then $f \circ g$ is a witness for $(x_i, p_i)_{i\in I} \prec (z_l, r_l)_{l\in L}$. Clearly it is surjective.

- $x_{f(g(l))} \lhd y_{g(l)} \lhd z_l$. And $\lhd$ is transitive.
- $p_i \ll \sum_{f(j)=i} q_j \ll \sum_{f(j)=i} \sum_{g(l)=j} r_l = \sum_{(f(g(l))=i} r_l$.

We have now to show the finite interpolation property. It is enough to consider the cases for which $|F| = 0, 2$. The case $|F| = 0$ is straightforward.

Now, let $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \prec (z_l, r_l)_{l\in L}$ with witnesses f and g. For every $l \in L$ consider the set $Z_l := \{x_{f(l)} \mid l \in L\} \cup \{y_{g(l)} \mid l \in L\}$. Since f, g are witnessing functions, we have $Z_l \lhd z_l$. Since $(X, \lhd)$ is an abstract basis, and Z_l is finite, by the interpolation property there exists a z'_l such that $Z_l \lhd z'_l \lhd z_l$.

Let $s := \sum_{l\in L} r_l$. Let $2s\epsilon$ be the minimum among all the numbers of the form

$$\left(\sum_{f(l)=i} r_l\right) - p_i, \quad \left(\sum_{g(l)=j} r_l\right) - q_j\,.$$

Consider $(z'_l, (1-\epsilon)r_l)_{l\in L}$. The identity function on L is a witness for $(z'_l, (1-\epsilon)r_l)_{l\in L} \prec (z_l, r_l)_{l\in L}$ while f, g are witnesses for $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \prec (z'_l, (1-\epsilon)r_l)_{l\in L}$:

$$\begin{aligned}
\left(\sum_{f(l)=i} (1-\epsilon) r_l\right) - p_i &= \left(\sum_{f(l)=i} r_l\right) - \epsilon\left(\sum_{f(l)=i} r_l\right) - p_i \\
&\geq \left(\sum_{f(l)=i} r_l\right) - \epsilon\left(\sum_{l\in L} r_l\right) - p_i \\
&= \left(\sum_{f(l)=i} r_l\right) - s\epsilon - p_i \\
&\gg \left(\sum_{f(l)=i} r_l\right) - 2s\epsilon - p_i \geq 0\,.
\end{aligned}$$

□

4.1.2 Indexed valuation as ideal completion

Definition 4.1.3. If D is a continuous domain generated by the abstract basis B, let $\mathcal{IV}(D)$ be the ideal completion of $(IV(B), \prec)$. Its elements are called *Hoare indexed valuations* or simply indexed valuations.

We will see later why the name of Hoare appears here.

Apparently the definition depends on the choice of a basis for D. We will show in the next Section that different choices of the basis give rise to isomorphic constructions.

As a corollary of the proof of Proposition 4.1.2 we have the following proposition.

Proposition 4.1.4. *If*

$$(b_i, p_i)_{i\in I} \prec (c_j, q_j)_{j\in J}\ ,$$

then for every j there exist $c'_j \ll c_j$ and $q'_j \ll q_j$ such that

$$(b_i, p_i)_{i\in I} \prec (c'_j, q'_j)_{j\in J} \prec (c_j, q_j)_{j\in J}\ .$$

Recall the definition of the function $\iota_B : B \to Idl(B)$ defined as $\iota_B(b) = \Downarrow b$. It is interesting to make the following observation.

Proposition 4.1.5. *If $\iota_B : B \to D$ is injective, then $\iota_{IV(B)} : IV(B) \to \mathcal{IV}(D)$ is injective.*

This is in contrast to what happens, for instance, with powerdomains.

In order to prove Proposition 4.1.5 we need some lemmas. The first two are easily proved by contraposition.

Lemma 4.1.6. *If $\iota_B : B \to D$ is injective, and $\iota\big((b, p)_{*\in\{*\}}\big) = \iota\big((b', p')_{*\in\{*\}}\big)$, then $b = b'$ and $p = p'$.*

Lemma 4.1.7. *If $\iota_B : B \to D$ is injective, and $\iota\big((b_i, p_i)_{i\in I}\big) = \iota\big((b'_j, p'_j)_{j\in J}\big)$, then $|I| = |J|$.*

Proposition 4.1.5 is consequence of the following lemma.

Lemma 4.1.8. *If $\iota\big((b_i, p_i)_{i\in I}\big) = \iota\big((b'_i, p'_i)_{i\in I}\big)$, then there exists a bijection $f : I \to I$, such that for all $i \in I$, $b_i = b'_{f(i)}$ and $p_i = p'_{f(i)}$.*

Proof: For every $i \in I$, pick $(a_i, q_i)_{*\in\{*\}}$ such that $(a_i, q_i)_{*\in\{*\}} \prec (b_i, p_i)_{*\in\{*\}}$ and such that whenever $(a_i, q_i)_{*\in\{*\}} \prec (b'_j, p'_j)_{*\in\{*\}}$ then $\iota\big((b_i, p_i)_{*\in\{*\}}\big) \subseteq \iota\big((b'_j, p'_j)_{*\in\{*\}}\big)$.

We have to show that such element exists. Let I_i be the set of $j \in I$ such that $\iota\big((b_i, p_i)_{*\in\{*\}}\big) \not\subseteq \iota\big((b'_j, p'_j)_{*\in\{*\}}\big)$. For every $j \in I_i$ we can thus find $(a^j_i, q^j_i)_{*\in\{*\}} \prec (b_i, p_i)_{*\in\{*\}}$, such that $(a^j_i, q^j_i)_{*\in\{*\}} \not\prec (b'_j, p'_j)_{*\in\{*\}}$. By the interpolation property we find a_i such that $a^j_i \ll a_i \ll b_i$ and q_i such that $q^j_i \ll q_i \ll p_i$. Clearly $(a_i, q_i)_{*\in\{*\}} \prec (b_i, p_i)_{*\in\{*\}}$, and if $j \in I_i$, $(a_i, q_i)_{*\in\{*\}} \not\prec (b'_j, p'_j)_{*\in\{*\}}$.

Since by hypothesis we have $\iota\big((b_i, p_i)_{i\in I}\big) = \iota\big((b'_i, p'_i)_{i\in I}\big)$, then it must be $(a_i, q_i)_{i\in I} \prec (b'_i, p'_i)_{i\in I}$. Let $f : I \to I$ be the witness, which is necessarily a bijection. Therefore $(a_{f(i)}, q_{f(i)})_{*\in\{*\}} \prec (b'_i, p'_i)_{*\in\{*\}}$ for all $i \in I$, and hence $\iota\big((b_{f(i)}, p_{f(i)})_{*\in\{*\}}\big) \subseteq \iota\big((b'_i, p'_i)_{*\in\{*\}}\big)$.

Symmetrically, there is $g : I \to I$, such that $\iota\big((b'_{g(f(i))}, p'_{g(f(i))})_{*\in\{*\}}\big) \subseteq \iota\big((b_{f(i)}, p_{f(i)})_{*\in\{*\}}\big)$.

Since gf is a permutation, then if $n = |I|$, $(gf)^n(i) = i$. So that

$$\begin{gathered}\iota\big((b'_i, p'_i)_{*\in\{*\}}\big) = \iota\big((b'_{(gf)^n(i))}, p'_{(gf)^n(i))})_{*\in\{*\}}\big)\\ \subseteq \iota\big((b_{f((gf)^{n-1}(i))}, p_{f((gf)^{n-1}(i))})_{*\in\{*\}}\big) \subseteq \ldots \subseteq \iota\big((b_{f(i)}, p_{f(i)})_{*\in\{*\}}\big)\\ \subseteq \iota\big((b'_i, p'_i)_{*\in\{*\}}\big)\end{gathered}$$

Therefore, for every $i \in I$ we have $\iota\big((b_{f(i)}, p_{f(i)})_{*\in\{*\}}\big) = \iota\big((b'_i, p'_i)_{*\in\{*\}}\big)$, which by lemma 4.1.6 implies the thesis. □

4.2 Equational characterisation

In this section we characterise Hoare indexed valuations as a free construction.

4.2.1 Continuous quasi-cones

Recall the equational theory (1)–(7) of section 3.3. We add only one more axiom, which corresponds to definition 4.1.1 of the AB-relation on finite indexed valuations.

- HV: $(p + q)A \sqsubseteq (pA \oplus qA)$

Definition 4.2.1. A *continuous Hoare quasi-cone*, or simply continuous quasi-cone, is a structure $(D, \sqsubseteq, \oplus, \odot)$ such that

- $(D, \sqsubseteq)$ is a continuous domain;
- $\oplus : D \times D \to D$ is continuous;
- $\odot : [0, +\infty[\times D \to D$ is continuous;
- axioms (1)–(7) + (HV) are satisfied.

We can extend the definition of the scalar multiplication to $+\infty$ by continuity. The defining axioms (1)–(7) + (HV) are still valid for this extended set of scalars.

Let **CONT** be the category of continuous domains, and **QCONT** be the category of continuous quasi-cones and continuous homomorphisms. (In fact, in what follows, we will always mention bases. Therefore **CONT** will be the category of abstract bases and continuous functions between their completions. This is clearly equivalent to the category of continuous domains and continuous functions. Similar considerations apply to all the other categories we will define.)

Proposition 4.2.2. *If D is a continuous domain then $\mathcal{IV}(D)$ is a continuous quasi-cone.*

Proof: By construction $\mathcal{IV}(D)$ is a continuous domain. We have to define the operations. We put

- $\mathcal{I} \oplus \mathcal{J} = \downarrow\{\nu \oplus \xi \,|\, \nu \in \mathcal{I}, \xi \in \mathcal{J}\}$;
- $p\mathcal{I} = \{p\nu \,|\, \nu \in \mathcal{I}\}$;
- $\underline{0} = \{(\, , \,)_{i \in \emptyset}\}$.

The operations are well defined: the sum of two ideals is downward closed by construction, while downward closedness of the other two and directedness follow from the fact that the operations on $IV(X)$ respect the AB-relation:

Lemma 4.2.3. *If ν, ξ, ν', ξ' are finite indexed valuation on X,*

a) If $\nu \prec \underline{0}$ then $\nu = \underline{0}$;

b) if $\nu \prec \nu'$ then $p\nu \prec p\nu'$;

c) if $\nu \prec \nu'$ & $\xi \prec \xi'$ then $\nu \oplus \xi \prec \nu' \oplus \xi'$.

Proof:

a) The empty function is a witness for $\underline{0} \prec \nu$. If $(x_i, p_i)_{i\in I}$ is a representative for $\underline{0}$, then $p_i = 0$ for all $i \in I$. If $\nu = (y_j, q_j)_{j\in J}$ and if $f : I \to J$ is a witness for $\nu \prec \underline{0}$ then $q_j \ll \sum_{f(i)=j} p_i = 0$. Therefore for all $j \in J$, $q_j = 0$.

b) A witness for $\nu \prec \nu'$ is also a witness for $p\nu \prec p\nu'$.

c) If f is a witness for $\nu \prec \nu'$ and g is a witness for $\xi \prec \xi'$, then, assuming that $dom(f) \cap dom(g) = \emptyset$, we have that $f \cup g$ is a witness for $\nu \oplus \xi \prec \nu' \oplus \xi'$.

□ (4.2.3)

It is easy to see that the operations satisfy axioms (1)–(7). The fact that they satisfy (HV) follows from the roundness of the ideals and the following lemma.

Lemma 4.2.4. *If $\nu \prec \xi$ and $p \ll q_1 + q_2$ then $p\nu \prec q_1\xi \oplus q_2\xi$.*

Proof: Let $\nu := (a_i, p_i)_{i\in I}, \xi := (b_j, r_j)_{j\in J}$. Since $\nu \prec \xi$, by definition there exists $f : J \twoheadrightarrow I$, s.t.:

$$a_{f(j)} \lhd b_j \;;$$
$$p_i \ll \sum_{f(j)=i} r_j \,.$$

We want to prove that $p\nu \prec q_1\xi \oplus q_2\xi$, i.e. that.

$$(a_i, pp_i)_{i\in I} \prec (b_j, q_k r_j)_{(j,k)\in J\times\{1,2\}} \,.$$

We need a function $f' : J \times \{1, 2\} \twoheadrightarrow I$, s.t.:

1. $a_{f'(j,k)} \lhd b_j$;
2. $pp_i \ll \sum_{f'(j,k)=i} q_k r_j$.

Define f' as follows: $f'(j, k) := f(j)$. First f' is clearly surjective. Secondly, the expression (1) is obviously satisfied. As for (2) notice first that $pp_i \ll (q_1+q_2)p_i$; since $p_i \ll \sum_{f(j)=i} r_j$, then for $k = 1, 2$, $q_k p_i \ll \sum_{f(j)=i} q_k r_j$. So $(q_1 + q_2)p_i \ll \sum_{f(j)=i} q_1 r_j + \sum_{f(j)=i} q_2 r_j = \sum_{f'(j,1)=i} q_1 r_j + \sum_{f'(j,2)=i} q_2 r_j = \sum_{f'(j,k)=i} q_k r_j$.

□ (4.2.4)

It is easy to show that the operations are continuous. The scalar multiplication is also continuous in the first argument. It is monotonic: if $\nu \in p\mathcal{I}$ by roundness there exists $p\xi \in p\mathcal{I}$ such that $\nu \prec p\xi$ with witness f. If $p \leq q$ then the same witness shows that $\nu \prec q\xi$. It preserves lubs: take an ideal $\mathcal{I} \in \mathcal{IV}(D)$. We want to prove that

$$\bigcup_{p \ll q} p\mathcal{I} = q\mathcal{I} \,,$$

that is for every ν, we have $\nu \in q\mathcal{I}$ if and only if there exist $p \ll q$ s.t. $\nu \in p\mathcal{I}$. The "if" direction follows from monotonicity. It remains to prove the other inclusion.

Take $\nu \in q\mathcal{I}$. By roundness there is $\nu' \in q\mathcal{I}$ s.t. $\nu \prec \nu'$. There exists ϵ such that the witness for $\nu \prec \nu'$ is also a witness for $\nu \prec (1 - \epsilon)\nu'$. But $\nu' \in q\mathcal{I}$, therefore $\nu \in (1 - \epsilon)q\mathcal{I}$. □

4.2.2 Indexed valuations as a free construction

Proposition 4.2.5. *The operator $\mathcal{IV}$ extends to a functor* **CONT** $\rightarrow$ **QCONT** *which is left adjoint to the forgetful functor.*

Proof: We show the universal property, which proves both that $\mathcal{IV}$ is a functor and that it is left adjoint. The fact that $\mathcal{IV}$ is a functor implies also that if B, B' are two abstract bases generating D then the ideal completions of $IV(B), IV(B')$ are isomorphic. For every continuous function $g : D \to \mathcal{E}$ where $\mathcal{E} \in$ **QCONT** there is a unique $g^\dagger : \mathcal{IV}(D) \to \mathcal{E}$ (in the category **QCONT**) s.t.

$$\begin{tikzcd}
D \arrow[d, "\eta"'] \arrow[dr, "g"] & \\
\mathcal{IV}(D) \arrow[r, dashed, "g^\dagger"'] & \mathcal{E}.
\end{tikzcd}$$

where $\eta(d) = \{(b,p)_{*\in\{*\}} \,|\, b \in d,\, p < 1\}$. We also claim that the assignment $g \mapsto g^\dagger$ is continuous.

Note first that η is continuous. Then take the "restriction" of g to B_D, defined as $g(b) = g(\iota_B(b))$. It has a unique homomorphic extension $\overline{g} : IV(B_D) \to \mathcal{E}$, defined by

$$\overline{g}\left((b_i, p_i)_{i\in I}\right) \quad := \quad \bigoplus_{i\in I} p_i g(b_i)\,.$$

We claim that $\overline{g}$ is monotonic, in the following sense.

Lemma 4.2.6. *If $\nu \prec \xi$, then $\overline{g}(\nu) \sqsubseteq \overline{g}(\xi)$.*

Proof: First suppose that $(b,p)_{*\in\{*\}} \prec (c_j, q_j)_{j\in J}$, and that the witness f for this is total. Then $p \ll \sum_{j\in J} q_j =: q$ and for every j, $b \ll c_j$. Notice also that $p = \sum_{j\in J} \frac{q_j}{q} p$. Applying iteratively the inequation (HV), we can show that:

$$\overline{g}\left((b,p)_{*\in\{*\}}\right) = p g(b) \sqsubseteq \bigoplus_{j\in J} \frac{q_j}{q} p g(b)\,.$$

Then, by monotonicity of the operations, of g and of ι_B,

$$\bigoplus_{j\in J} \frac{q_j}{q} p g(b) \sqsubseteq \bigoplus_{j\in J} q_j g(c_j) = \overline{g}\left((c_j, q_j)_{j\in J}\right)\,.$$

Now suppose $(b_i, p_i)_{i\in I} \prec (c_j, q_j)_{j\in J}$ with again a total witness f. Let $J_i = f^{-1}(i)$. Clearly $(b_i, p_i)_{*\in\{*\}} \prec (c_j, q_j)_{j\in J_i}$ for every $i \in I$. Therefore

$$\overline{g}\left((b_i, p_i)_{*\in\{*\}}\right) \sqsubseteq \overline{g}\left((c_j, q_j)_{j\in J_i}\right)\,.$$

Notice that $(b_i, p_i)_{i\in I} = \bigoplus_{i\in I} (b_i, p_i)_{*\in\{*\}}$ and $(c_j, q_j)_{j\in J} = \bigoplus_{i\in I} (c_j, q_j)_{j\in J_i}$. Monotonicity of the sum, and the homomorphism condition on $\overline{g}$ imply that

$$\overline{g}\left((b_i, p_i)_{i\in I}\right) \sqsubseteq \overline{g}\left((c_j, q_j)_{j\in J}\right)\,.$$

Finally, for the case where f is not total, let J_0 be the domain of f and J_1 be its complement. Clearly $(c_j, q_j)_{j\in J} = (c_j, q_j)_{j\in J_0} \oplus (c_j, q_j)_{j\in J_1}$. Moreover

$$\overline{g}\left((b_i, p_i)_{i\in I}\right) \sqsubseteq \overline{g}\left((c_j, q_j)_{j\in J_0}\right)\,.$$

Equations (4)-(5) together with the monotonicity of the scalar multiplication imply $\underline{0} \sqsubseteq A$. Therefore

$$\underline{0} = \overline{g}\left((\ ,\ )\right) \sqsubseteq \overline{g}\left((c_j, q_j)_{j\in J_1}\right) \ .$$

And finally

$$\overline{g}\left((b_i, p_i)_{i\in I}\right) = \overline{g}\left((b_i, p_i)_{i\in I} \oplus (\ ,\ )\right) \sqsubseteq \overline{g}\left((c_j, q_j)_{j\in J}\right) \ .$$

□ (4.2.6)

Let us call $g^\dagger$ the extension of $\overline{g}$ to $\mathcal{IV}(D)$, the ideal completion of $(IV(B_D)$. We recall that $g^\dagger(\mathcal{I}) := \bigsqcup^{\uparrow}{}_{\nu\in\mathcal{I}}\, \overline{g}(\nu)$. We know that the function $g^\dagger$ is continuous. The continuity of the operations implies that $g^\dagger$ is also an homomorphism. Thus it is a morphism of the category.

It remains to show that $g^\dagger\big(\{(b,p)_{*\in\{*\}} \,|\, b \in d,\, p < 1\}\big) = g(d)$

Now

$$g^\dagger\big(\{(b,p)_{*\in\{*\}} \,|\, b \in d,\, p < 1\}\big) = \bigsqcup_{b\in d,\, p<1}^{\uparrow} \overline{g}((b,p)_{*\in\{*\}})$$

$$= \bigsqcup_{b\in d,\, p<1}^{\uparrow} pg(b) = \bigsqcup_{p<1}^{\uparrow} pg(d) = g(d)\,.$$

The last two equalities follow from the continuity of g and of the scalar multiplication. We also use that $\bigsqcup^{\uparrow}{}_{b\in d}\, \iota_B(b) = d$.

To prove uniqueness we need the following lemma.

Lemma 4.2.7. *If ν, ξ are finite indexed valuations on B,*

a) $\iota_{IV(B)}(\underline{0}) = \underline{0}$;

b) $p(\iota_{IV(B)}(\nu)) = \iota_{IV(B)}(p\nu)$;

c) $\iota_{IV(B)}(\nu) \oplus \iota_{IV(B)}(\xi) = \iota_{IV(B)}(\nu \oplus \xi)$.

Proof: Equation a) holds by definition. Equation b) is easily proved using the definitions. Equation c) is a consequence of Proposition 4.1.4. □ (4.2.7)

As a consequence, for every $\nu \in IV(B_D)$, $\overline{g}(\nu) = g^\dagger(\iota(\nu))$.

$$\begin{tikzcd} IV(B_D) \arrow[d, "\iota"'] \arrow[dr, "\overline{g}"] & \\ \mathcal{IV}(D) \arrow[r, dashed, "g^\dagger"'] & \mathcal{E}. \end{tikzcd}$$

Let $h : \mathcal{IV}(D) \to \mathcal{E}$ be a continuous homomorphism such that for every $d \in D$, $h(\eta(d)) = g(d)$. Since h is an homomorphism, we have that for every $(b_i, p_i)_{i\in I} \in IV(B_D)$

$$h(\iota\big((b_i, p_i)_{i\in I}\big)) = \bigoplus_{i\in I} p_i h(\iota\big((b_i, 1)_{*\in\{*\}}\big)) = \bigoplus_{i\in I} p_i h(\eta(b_i)) = \bigoplus_{i\in I} p_i g(b_i)$$

$$= \overline{g}\big((b_i, p_i)_{i\in I}\big) = g^\dagger\big(\downarrow(b_i, p_i)_{i\in I}\big)$$

Since h and $g^\dagger$ coincide on the basis, they are equal.

Continuity of the operations and an exchange of suprema proves continuity of the assignment $g \mapsto g^\dagger$. □

We want to give a meaning to the expression $(d_i, p_i)_{i\in I}$ even when $d_i \in D$ but not necessarily $d_i \in B$. We stipulate that

$$(d_i, p_i)_{i\in I} := \bigoplus_{i\in I} p_i \eta(d_i)\,.$$

Proposition 4.2.8.

$$(d_i, p_i)_{i\in I} = \downarrow \left\{ (b_i, q_i)_{i\in I} \mid b_i \ll d_i \;\&\; q_i \ll p_i \right\}.$$

Omitting the mention of the forgetful functor, we can say that $\mathcal{IV}$ is a monad in **CONT**. The unit $\eta^{\mathcal{IV}}_D : D \to \mathcal{IV}(D)$ is the extension of $\eta^{IV}_B : B \to IV(B)$; the multiplication $\mu^{\mathcal{IV}}_D : \mathcal{IV}^2(D) \to \mathcal{IV}(D)$ is the extension of $\mu^{IV}_B : IV^2(B) \to IV(B)$ defined as in the category **SET**.

4.3 Relationship with continuous valuations

The fact that the definition of the abstract basis is similar to the Splitting Lemma is reflected by the following theorem which shows the tight relationship between Hoare indexed valuations and continuous valuations.

Let's call $\mathit{Flat}_D : \mathcal{IV}(D) \to \mathcal{V}(D)$ the extension of $\eta^{\mathcal{V}}_D : D \to \mathcal{V}(D)$. This makes sense since $\mathcal{V}$ is a continuous d-cone, and then, a fortiori, is continuous quasi-cone. The function Flat_D "forgets" the indices. It "flattens" an indexed valuation down to a valuation. For a finite indexed valuation $\nu := (b_i, p_i)_{i\in I}$, we have that

$$\mathit{Flat}(\iota(\nu)) = \bigoplus_{i\in I} p_i \eta_{b_i}\,.$$

This operation has an adjoint defined as

$$\mathit{Sat}_D \nu = \{\xi \in IV(B_D) \mid \mathit{Flat}(\iota(\xi)) \ll \nu\}\,.$$

The function Sat_D, takes a continuous valuation and returns a corresponding indexed valuation "with the most possible indices".

Theorem 4.3.1. *Let $\mathcal{V}$ be Jones' powerdomain of valuations functor. The functions $\mathit{Flat}_D, \mathit{Sat}_D$ are continuous homomorphisms of real quasi-cones, natural in D and form a continuous insertion-closure pair $\mathcal{IV}(D) \to \mathcal{V}(D)$.*

In particular the function Flat_D is surjective: every continuous valuation has an indexed representative. The fact that Sat is an adjoint implies that its definition does not depend on the choice of the basis.

Proof: To simplify the notation, we assume that $\iota : B_D \to D$ is injective, so as to drop the mention of ι. The proof does not rely on this assumption. We first prove that Sat_D is well defined. It is enough to prove that for every $\nu \in \mathcal{V}(D)$ the set $W = \{\xi \in IV(B_D) \mid \mathit{Flat}(\xi) \ll \nu\}$ is an ideal in $\mathcal{IV}(D)$. It is clearly downward closed, because Flat is monotonic. Take $\xi_1, \xi_2 \in W$. By directedness of the set of valuations way below ν, there exists a simple valuation ζ such that

$\mathit{Flat}(\xi_1), \mathit{Flat}(\xi_2) \ll \zeta \ll \nu$. We want to build a finite indexed valuation ξ such that $\xi_1, \xi_2 \prec \xi$ and $\mathit{Flat}(\xi) = \zeta \ll \nu$. This proves the directedness of W. Let $\xi_1 := (a_i, p_i)_{i \in I}$, $\xi_2 := (b_j, q_j)_{j \in J}$. Assume that $i \neq i' \Longrightarrow a_i \neq a_{i'}$ and similarly for ξ_2. We will argue that this is not a restriction.

Therefore we can write $\xi_1 = (a, p_a)_{a \in A}$ and $\xi_2 = (b, q_b)_{b \in B}$. Since *Flat* is a homomorphism, $\mathit{Flat}(\xi_1) = \sum_{a \in A} p_b \eta_b$, $\mathit{Flat}(\xi_2) = \sum_{b \in B} q_b \eta_b$. Let $\zeta := \sum_{c \in C} s_c \eta_c$. By the splitting lemma we know that there are $t^1_{a,c}$, and $t^2_{b,c}$, such that

$$\sum_{c \in C} t^1_{a,c} = p_a, \quad \sum_{c \in C} t^2_{b,c} = q_b,$$

$$\sum_{a \in A} t^1_{a,c} < s_c, \quad \sum_{b \in B} t^2_{b,c} < s_c,$$

with $t^1_{a,c} > 0 \Longrightarrow a \ll c$ and $t^2_{b,c} > 0 \Longrightarrow b \ll c$.

Define $\xi := (x_{a,b,c}, t_{a,b,c})_{(a,b,c) \in A \times B \times C}$ where $x_{a,b,c} := c$ and

$$t_{a,b,c} := \frac{s_c t^1_{a,c} t^2_{b,c}}{\sum_{a,b \in A \times B} t^1_{a,c} t^2_{b,c}} \, .$$

We have $\mathit{Flat}(\xi) = \zeta$.

Then consider $f(a,b,c) := a$, defined for all (a,b,c) for which $t^1_{a,c} > 0$. Similarly $g(a,b,c) := b$ defined for all (a,b,c) for which $t^2_{b,c} > 0$. We show that f is a witness for $\xi_1 \prec \xi$ (and similarly for g and ξ_2). For every a we have $p_a > 0$ and $p_a = \sum_{c \in C} t^1_{a,c}$. Therefore some of the $t^1_{a,c} > 0$ and f is surjective. For the (a,b,c) where f is defined we have $a \ll c = x_{a,b,c}$ because $t^1_{a,c} > 0$. Finally we have to show that

$$p_a < \sum_{f(a,b,c)=a} t_{a,b,c} =: r_a \, .$$

Let's go:

$$r_a = \sum_{f(a,b,c)=a} t_{a,b,c} = \sum_{b \in B, t^1_{a,c} > 0} \frac{s_c t^1_{a,c} t^2_{b,c}}{\sum_{a,b \in A \times B} t^1_{a,c} t^2_{b,c}} \, .$$

Adding the $t^1_{a,c} = 0$ we have that

$$\begin{aligned} r_a &= \sum_{b \in B, c \in C} \frac{s_c t^1_{a,c} t^2_{b,c}}{\sum_{a,b \in A \times B} t^1_{a,c} t^2_{b,c}} = \sum_{c \in C} \frac{s_c t^1_{a,c} \sum_{b \in B} t^2_{b,c}}{\sum_{a \in A} t^1_{a,c} \sum_{b \in B} t^2_{b,c}} \\ &= \sum_{c \in C} \frac{s_c t^1_{a,c} \sum_{b \in B} t^2_{b,c}}{(\sum_{a \in A} t^1_{a,c})(\sum_{b \in B} t^2_{b,c})} = \sum_{c \in C} \frac{s_c t^1_{a,c}}{\sum_{a \in A} t^1_{a,c}} \, . \end{aligned}$$

Since $\sum_{a \in A} t^1_{a,c} < s_c$ we have that

$$r_a > \sum_{c \in C} \frac{s_c t^1_{a,c}}{s_c} = \sum_{c \in C} t^1_{a,c} = p_a \, .$$

The assumption that $i \neq i' \Longrightarrow a_i \neq a_{i'}$ is not restrictive: let's try to suggest why. Suppose $\nu = (a, p)_{* \in \{*\}} \prec \xi = (y_j, q_j)_{j \in J}$. Now we split the index in ν: take $\nu' := (a_k, r_k p)_{k \in \{1,2\}}$ where $a_k = a$ for $k = 1, 2$ and $r_1 + r_2 = 1$. Consider then $\xi' := (y_{j,k}, r_k q_j)_{(j,k) \in J \times \{1,2\}}$. It should be clear that $\nu' \prec \xi'$. Moreover, although $\xi \not\prec \xi'$, for every $\zeta \prec \xi$ we have $\zeta \prec \xi'$. Thus if we find an upper bound ξ for an indexed valuation satisfying the restriction, we find a suitable upper bound ξ' for the more general ones.

Monotonicity and continuity of Sat_D are obvious. We have to check that Sat_D is a homomorphism. Clearly $Sat(\underline{0}) = \underline{0}$. Also $Sat(p\nu) = pSat(\nu)$. To show that $Sat(\nu_1 \oplus \nu_2) = Sat(\nu_1) \oplus Sat(\nu_2)$, we first notice that, since $Flat$ is a homomorphism and since $\ll$ respects $\oplus$ in $\mathcal{V}(D)$, then $Sat(\nu_1) \oplus Sat(\nu_2) \subseteq Sat(\nu_1 \oplus \nu_2)$. Take now $\zeta \in Sat(\nu_1 \oplus \nu_2)$. To prove that $\zeta \in Sat(\nu_1) \oplus Sat(\nu_2)$ it is enough to show that there are ζ_1, ζ_2 such that $Flat(\zeta_1) \ll \nu$, $Flat(\zeta_2) \ll \xi$ and $\zeta \prec \zeta_1 \oplus \zeta_2$.

Consider the sets $\twoheaddownarrow \nu_1$ and $\twoheaddownarrow \nu_2$. Since the addition preserves the way-below relation (proposition 2.22 in [Tix99]) we have that $\twoheaddownarrow \nu_1 + \twoheaddownarrow \nu_2 \subseteq \twoheaddownarrow \nu_1 + \nu_2$. Notice that $\twoheaddownarrow \nu_1 + \twoheaddownarrow \nu_2$ is directed with lub $\nu_1 + \nu_2$. Thus for every $\chi \ll \nu_1 + \nu_2$ there exists $\chi_1 + \chi_2 \in \twoheaddownarrow \nu_1 + \twoheaddownarrow \nu_2$ such that $\chi \ll \chi_1 + \chi_2$. With enough indices we can find ζ_1, ζ_2, such that $Flat(\zeta_1) = \chi_1$, $Flat(\zeta_2) = \chi_2$ and $\zeta \prec \zeta_1 \oplus \zeta_2$.

We now prove that $Flat_D \circ Sat_D$ is the identity on $\mathcal{V}(D)$. Take a valuation $\nu \in \mathcal{V}(D)$. We know (Theorem 2.5.10) that it is the directed supremum of the set $\twoheaddownarrow \nu$ of all way-below simple valuations. Now $Sat_D(\nu) = \{\xi \in IV(B_D) \mid Flat(\xi) \ll \nu\}$. By definition of $Flat_D$, we have that $Flat_D(Sat_D(\nu)) = \bigsqcup^{\uparrow}{}_{\xi \in Sat_D(\nu)} Flat(\xi)$. And since $Flat$ is surjective onto the set of simple valuations, we have that $\bigsqcup^{\uparrow}{}_{\xi \in Sat_D(\nu)} Flat(\xi) = \bigsqcup^{\uparrow}{}_{\zeta \ll \nu} \zeta = \nu$.

It is easy to see that $Sat_D \circ Flat_D$ is above the identity on $\mathcal{IV}(D)$.

We now prove naturality of $Flat_D$. We have to show that for every continuous $f : D \to E$ the following diagram commutes.

$$\begin{tikzcd}
\mathcal{IV}(D) \arrow[r, "Flat_D"] \arrow[d, "\mathcal{IV}(f)"'] & \mathcal{V}(D) \arrow[d, "\mathcal{V}(f)"] \\
\mathcal{IV}(E) \arrow[r, "Flat_E"] & \mathcal{V}(E)
\end{tikzcd}$$

We prove this by showing that both sides of the squares are equal to the extension of $\eta^{\mathcal{V}}_E \circ f$.

$$\begin{tikzcd}
D \arrow[rr, "\eta^{\mathcal{IV}}_D"] \arrow[dr, "f"'] & & \mathcal{IV}(D) \arrow[dd, dashed] \\
& E \arrow[dr, "\eta^{\mathcal{V}}_E"'] & \\
& & \mathcal{V}(E)
\end{tikzcd}$$

In the following diagram

$$\begin{tikzcd}
D \arrow[rr, "\eta^{\mathcal{IV}}_D"] \arrow[dd, "f"'] & & \mathcal{IV}(D) \arrow[dl, "\mathcal{IV}(f)"'] \arrow[dd, dashed] \\
& \mathcal{IV}(E) \arrow[dr, "Flat_E"] & \\
E \arrow[ur, "\eta^{\mathcal{IV}}_E"'] \arrow[rr, "\eta^{\mathcal{V}}_E"'] & & \mathcal{V}(E)
\end{tikzcd}$$

the upper left triangle commutes because of the naturality of $\eta^{\mathcal{IV}}$. The lower triangle commutes by definition of $Flat_E$. The right triangle commutes by uniqueness.

In the following diagram

$$\begin{tikzcd}
D \arrow[rr, "\eta^{\mathcal{IV}}_D"] \arrow[dd, "f"'] \arrow[dr, "\eta^{\mathcal{V}}_D"'] & & \mathcal{IV}(D) \arrow[dl, "Flat_D"'] \arrow[dd, dashed] \\
& \mathcal{V}(D) \arrow[dr, "\mathcal{V}(f)"'] & \\
E \arrow[rr, "\eta^{\mathcal{V}}_E"'] & & \mathcal{V}(E)
\end{tikzcd}$$

the upper triangle commutes by definition of $Flat_D$. The lower left triangle commutes because of the naturality of $\eta^{\mathcal{V}}$. The right triangle commutes by uniqueness.

The naturality of Sat_D,

$$\begin{tikzcd}
\mathcal{V}(D) \arrow[r, "Sat_D"] \arrow[d, "\mathcal{V}(f)"'] & \mathcal{IV}(D) \arrow[d, "\mathcal{IV}(f)"] \\
\mathcal{V}(E) \arrow[r, "Sat_E"] & \mathcal{IV}(E)
\end{tikzcd}$$

can be proved with some pain using the following observations (which we have already used implicitly).

Proposition 4.3.2. *If $\xi, \zeta \in IV(B_D)$ and $\xi \prec \zeta$, then $Flat_D(\xi) \ll Flat_D(\zeta)$. If $\xi, \zeta \in IV(B_D)$ and $Flat_D(\xi) \ll Flat_D(\zeta)$, then there exists $\zeta' \in IV(B_D)$ such that $\xi \prec \zeta'$ and $Flat_D(\zeta') = Flat_D(\zeta)$.*

The idea being that ζ' can contain more indices than ζ. □

4.4 The distributive law

We now show that there is a categorical distributive law between the indexed valuations monad, and the Hoare powerdomain monad. We will first show this by showing that the Hoare powerdomain monad lifts to the category of algebras for $\mathcal{IV}$ in the category **CONT**. We then propose an alternative proof using the abstract bases. This second proof is less insightful and more technical than the previous one, but has the advantage that it is possible to apply the same technique for the Smyth and the Plotkin powerdomains, for which a concrete characterisation is less forthcoming.

4.4.1 The distributive law via Beck's theorem

First, using Beck's monadicity theorem, we prove that the category of $\mathcal{IV}$-algebras is equivalent to **QCONT**. This proof is basically the one in Section 2.2, although it has to be recast in the context where scalar multiplication is continuous in the first argument, which is straightforward.

We have then to define the lifting of the Hoare powerdomain to the category **QCONT**. That is, when $\mathcal{E}$ is a Hoare continuous quasi-cone, we have to define a Hoare continuous quasi-cone structure on $\mathcal{P}_H(\mathcal{E})$. Then we show a universal property for this operator, showing that it lifts the monad in **CONT**. Recall that when D is a continuous domain, the Hoare powerdomain $\mathcal{P}_H(D)$ is the free join-semilattice on D and it is concretely characterised as the set of nonempty, Scott-closed subsets of D, ordered by inclusion (see Section 2.4).

Let's define the operations on $\mathcal{P}_H(\mathcal{E})$. We put

- $\underline{0} := \{\underline{0}\}$;
- $pC := \{p\nu \mid \nu \in C\}$;
- $C \oplus C' := \overline{\{\nu \oplus \nu' \mid \nu \in C, \nu' \in C'\}}$.

We have to show that these operations are well defined and satisfy the axioms. $\underline{0}$ is closed, and if C is closed then pC is closed. Finally $C \oplus C'$ is closed by definition.

The operations are obviously monotonic.

We show that the operations are continuous in the stronger sense that they preserve all suprema, even non-directed ones. The scalar multiplication is continuous in the second argument: take a family of closed sets $(C_i)_{i\in I}$. We have to check that $p\overline{\bigcup C_i} = \overline{\bigcup pC_i}$. This follows from the fact that for every set X, $p\overline{X} = \overline{pX}$, which in turn follows from the fact that multiplying by $p > 0$ is a homeomorphism. The scalar multiplication is continuous in the first argument. Let $(p_i)_{i\in I}$ be a chain in $\mathbb{R}^+$, and let $p := \sup_{i\in I} p_i$. If $p\nu \in pC$ then clearly $p\nu = \sup_{i\in I} p_i\nu \in \overline{\bigcup p_iC}$. The other direction follows from monotonicity. To prove continuity of the addition let $(C_i)_{i\in I}$ be a family of closed sets. We have to prove that

$$\overline{\bigcup_{i\in I} C_i} \oplus C \subseteq \overline{\bigcup_{i\in I}(C_i \oplus C)}$$

To do that it is enough to show that

$$\bigcup_{i\in I} C_i \oplus C \subseteq \overline{\bigcup_{i\in I}(C_i \oplus C)}$$

Take $\nu \in \bigcup_{i\in I} C_i \oplus C$. Let's say $\nu = \xi_i \oplus \xi$ with $\xi_i \in C_i$ and $\xi \in C$. But then $\nu \in C_i \oplus C$.

We omit the simple proofs of most of the axioms. To prove axiom (HV), pick $(p+q)\nu \in (p+q)C$. We have to show that $(p+q)\nu \in pC \oplus qC$. But $p\nu \in pC$ and $q\nu \in qC$, so that $p\nu \oplus q\nu \in pC \oplus qC$. By definition $pC \oplus qC$ is downward closed. Since $\mathcal{E}$ is satisfies (HV) then $(p+q)\nu \sqsubseteq p\nu \oplus q\nu$ so that $(p+q)\nu \in pC \oplus qC$.

I want to draw the reader's attention to this last paragraph. We use the fact that the sets in the Hoare powerdomain are downward closed. If we tried to lift one of the other powerdomains, the proof would break down here. In the next section we will we give an alternative proof of the distributive law and we will again draw the reader's attention when we reach the crucial point.

We now show the universal property that at once shows that $\mathcal{P}_H$ is a monad in **QCONT**, and that it lifts the monad in **CONT**. First we notice that $\eta^{\mathcal{P}_H}_{\mathcal{E}} : \mathcal{E} \to \mathcal{P}_H(\mathcal{E})$ lifts to a morphism in **QCONT**. We have to check that $\eta(\underline{0}) = \{\underline{0}\}$ which is true. Secondly $\eta(p\nu) = \overline{\{p\nu\}} = p\overline{\nu} = p\eta(\nu)$. Finally $\eta(\nu \oplus \xi) = \overline{\{\nu \oplus \xi\}} = \overline{\{\nu' \oplus \xi' \mid \nu' \sqsubseteq \nu, \xi' \sqsubseteq \xi\}} = \overline{\{\nu\}} \oplus \overline{\{\xi\}} = \eta(\nu) \oplus \eta(\xi)$. The second equality holds because of monotonicity of $\oplus$ in $\mathcal{E}$.

We need to define the category QCJ-algebras **QCJ**.(QCJ stands for quasicone join-semilattice).

Definition 4.4.1. A *continuous QCJ-algebra* is a continuous domain algebra over the theory (1)–(12) + (HV) + (HP), with the extra requirement that the scalar multiplication be continuous in the first argument.

The category **QCJ** has QCJ-algebras as objects and continuous homomorphisms as arrows.

The universal property is the following: for every morphism $g : \mathcal{E} \to \mathcal{J}$ in **QCONT**, with $\mathcal{J} \in$ **QCJ** there is a unique $g^\dagger : \mathcal{P}_H(\mathcal{E}) \to \mathcal{J}$ (in the category **QCJ**) s.t.

$$\begin{tikzcd}
\mathcal{E} \arrow[d, "\eta"'] \arrow[dr, "g"] & \\
\mathcal{P}_H(\mathcal{E}) \arrow[r, dashed, "g^\dagger"'] & \mathcal{J}.
\end{tikzcd}$$

We first have to observe that $\mathcal{P}_H(\mathcal{E})$ is an object of the category. We have seen it satisfies axioms (1)–(7),(HV) and we know it satisfies axioms (8)–(10),(HP), because the definition of $\uplus$ is the same as in the category **CONT**. We have only to show the distributive axioms (11)–(12), which is straightforward.

Finally we have to observe that the extension obtained by the universal property of the Hoare powerdomain preserves sum, scalar product and $\underline{0}$.

$$g^\dagger(\{\underline{0}\}) = g^\dagger(\eta(\underline{0})) = g(\underline{0}) = \underline{0}$$

$$g^\dagger(pC) = \bigsqcup_{\nu \in pC} g(\nu) = \bigsqcup_{\nu \in C} g(p\nu) = p \bigsqcup_{\nu \in C} g(\nu) = pg^\dagger(C)$$

$$g^\dagger(C \oplus C') = \bigsqcup_{\nu \in C \oplus C'} g(\nu)$$

Now $C \oplus C'$ is the Scott-closure of the set $W := \{\nu \oplus \nu' \mid \nu \in C, \nu' \in C'\}$. We argue that $\bigsqcup_{\xi \in \overline{W}} g(\xi) = \bigsqcup_{\xi \in W} g(\xi)$.

The hard direction is showing that $\bigsqcup_{\xi \in \overline{W}} g(\xi) \sqsubseteq \bigsqcup_{\nu \oplus \nu' \in W} g(\nu \oplus \nu')$. Take $\xi \in \overline{W}$, by Lemma 2.5.2, $\xi = \bigsqcup^{\uparrow}{}_{zeta \in Z} \zeta$ for some directed $Z \subseteq {\downarrow} W$. Since g is continuous, $g(\xi) = \bigsqcup^{\uparrow}_{\zeta \in Z} g(\zeta) \sqsubseteq \bigsqcup_{\nu \oplus \nu' \in W} g(\nu \oplus \nu')$. Since this is true for any $\xi \in \overline{W}$, our claim is proved. Therefore

$$g^{\dagger}(C \oplus C') = \bigsqcup_{\nu \in C, \nu' \in C'} g(\nu \oplus \nu')$$

Since g is a homomorphism of quasi-cones, we have $\bigsqcup_{\nu \oplus \nu' \in W} g(\nu \oplus \nu') = \bigsqcup_{\nu \in C, \nu' \in C'} g(\nu) \oplus (\nu')$. Equation (12) tells us that $\oplus$ is a homomorphism of join-semilattices and therefore preserves all least upper bounds (not only the directed ones). Thus

$$\bigsqcup_{\nu \in C, \nu' \in C'} g(\nu) \oplus g(\nu') = \bigsqcup_{\nu \in C} g(\nu) \oplus \bigsqcup_{\nu' \in C'} g(\nu') = g^{\dagger}(C) \oplus g^{\dagger}(C')$$

The way the extension is defined is exactly the same as the way as in the monad in **CONT**. This automatically implies that the multiplication lifts. We have thus proved the following theorem.

Theorem 4.4.2. *The Hoare powerdomain monad lifts to a monad in the category of continuous Hoare quasi-cones.*

By Beck's theorem on distributive laws (Theorem 2.2.3), we obtain the existence of the distributive law. Note also that we have obtained the lifted monad via an adjunction involving the category **QCJ**. This shows the coincidence of the equational and the categorical distributive laws.

4.4.2 The distributive law via the bases

We provide an alternative proof of the existence of the distributive law. We will use some notions defined in Section 2.3.

Given a continuous domain D with basis B, consider the set $P(B)$ of nonempty finite subsets of B, endowed with the Hoare AB-relation:

$$X \prec Y \iff \forall x \in X. \exists y \in Y. x \ll y\,.$$

It is known (see [AJ94]) that $(P(B), \prec)$ is a basis for the Hoare powerdomain $\mathcal{P}_H(D)$ (in the sequel we write $\mathcal{P}$ for $\mathcal{P}_H$).

To define a distributive law we need to give a family α_D of continuous functions $\mathcal{IV} \circ \mathcal{P}(D) \to \mathcal{P} \circ \mathcal{IV}(D)$. Our approach is to define a function between the bases and take the extension as our candidate. Consider the function $a_B : IV(P(B)) \to P(IV(B))$:

$$a_B\big((S_i, p_i)_{i \in I}\big) = \big\{(h(i), p_i)_{i \in I} \,|\, h : I \to B,\ h(i) \in S_i\big\}\,.$$

Lemma 4.4.3. *The function a_B is strongly monotonic and complete.*

Proof: Take $(S_i, p_i)_{i \in I} \prec (T_j, q_j)_{j \in J}$. Let $f : J \twoheadrightarrow I$ be a witness for that. Therefore:

- $p_i \ll \sum_{f(j)=i} q(j)$;
- $S_{f(j)} \prec T_j$.

The second formula is by definition equivalent to saying that for every $b \in S_{f(j)}$ there exists $c \in T_j$ such that $b \ll c$.

To prove strong monotonicity we have to prove that

$$\left\{(h(i), p_i)_{i\in I} | \, h : I \to B, \; h(i) \in S_i\right\} \prec \left\{(k(j), q_j)_{j\in J} | \, k : J \to B, \; k(j) \in T_j\right\}.$$

We have to show that for every $h : I \to B, h(i) \in S_i$ there exist a $k : J \to B, k(j) \in T_j$ such that $(h(i), p_i)_{i\in I} \prec (k(j), q_j)_{j\in J}$. How is k defined? For every j, consider $h(f(j))$. It is an element of $S_{f(j)}$. Therefore there exists some $c \in T_j$ with $h(f(j)) \ll c$. Let $k(j)$ be one such c. Now we claim that f is a witness of $(h(i), p_i)_{i\in I} \prec (k(j), q_j)_{j\in J}$. We have already $p_i \ll \sum_{f(j)=i} q(j)$. And by construction $h(f(j)) \ll k(j)$, so we are done.

To prove completeness take $\mathcal{S} \prec \left\{(h(i), p_i)_{i\in I} | \, h : I \to B, \; h(i) \in S_i\right\}$. We are going to find sets T_i and numbers q_i such that $(T_i, q_i)_{i\in I} \prec (S_i, p_i)_{i\in I}$ and $\mathcal{S} \prec \left\{(k(i), q_i)_{i\in I} | \, k : I \to B, \; k(i) \in T_i\right\}$. For every $\nu \in \mathcal{S}$ there exists $h : I \to B$ such that $\nu \prec (h(i), p_i)_{i\in I}$. By Proposition 4.1.4 there are $b_i \ll h(i)$ and $r_i \ll p_i$ such that $\nu \prec (b_i, r_i)_{i\in I} \prec (h(i), p_i)_{i\in I}$. Let T_i be the collection of all such b_i's and q_i be the maximum of all the r_i's. Clearly $T_i \prec S_i$ and $q_i \ll p_I$. Moreover by defining $k(i) = b_i$ (choosing one such) we get $\nu \prec (k(i), q_i)_{i\in I}$, so that $\mathcal{S} \prec \left\{(k(i), q_i)_{i\in I} | \, k : I \to B, \; k(i) \in T_i\right\}$. □

Note that it is essential the way the AB-relation is defined. Had we used the Egli-Milner or the Smyth AB-relation on finite sets, a_B would not be strongly monotonic. As we observed in the previous section, we are not able to find a distributive law between the Hoare indexed valuations, and either the Plotkin or the Smyth powerdomain and here is where the candidate proof breaks down.

Define α_D to be the extension of a_B.

Theorem 4.4.4. *The family $\alpha_D : \mathcal{IV}(\mathcal{P}(D)) \to \mathcal{P}(\mathcal{IV}(D))$ defined above is a distributive law.*

Proof: We prove naturality relying on the naturality of a_B in the category of sets. The proof is conceptually simple, but rather technical. Working with bases has its price.

We first state some lemmas, of which we omit the proofs. In the following B, B' will be abstract basis, D, D' their ideal extension, $P(B)$ will be endowed with the Hoare AB-relation, and $IV(B)$ will be endowed with the AB-relation of definition 4.1.1.

Lemma 4.4.5. *The following hold:*

- $\eta^P_B : B \to P(B)$ *is strongly monotonic and complete;*
- $\mu^P_B : P(P(B)) \to P(B)$ *is strongly monotonic and complete;*
- $\eta^{IV}_B : B \to IV(B)$ *is weakly monotonic and complete;*
- $\mu^{IV}_B : IV(IV(B)) \to IV(B)$ *is strongly monotonic and complete;*

and moreover

- $\eta^{\mathcal{P}}_D = ext(\eta^P_B)$;
- $\mu^{\mathcal{P}}_D = ext(\mu^P_B)$;

- $\eta^{\mathcal{IV}}_D = ext(\eta^{IV}_B)$;
- $\mu^{\mathcal{IV}}_D = ext(\mu^{IV}_B)$.

Lemma 4.4.6. *Let $f : B \to B'$ be weakly monotonic and complete. Then*

- $ext(P(f)) = \mathcal{P}(ext(f))$;
- $ext(IV(f)) = \mathcal{IV}(ext(f))$.

Finally we recall one of the statements of Proposition 2.3.6. Let B, B', B'' be abstract bases. Let $f : B \to B'$ and $g : B' \to B''$ be weakly monotonic and complete. Then $g \circ f$ is also weakly monotonic and complete and $ext(g) \circ ext(f) = ext(g \circ f)$.

Now, let $f : D \to D'$ be a continuous function. First suppose there exists a function $f^r : B \to B'$ (the "restriction" of f to the bases) such that

$$\begin{tikzcd}
B \arrow[r, "\iota_B"] \arrow[d, "f^r"'] & D \arrow[d, "f"] \\
B' \arrow[r, "\iota_{B'}"'] & D'.
\end{tikzcd}$$

Then f^r is weakly monotonic and complete. We have

Lemma 4.4.7. *Let B, B' be two abstract bases, and let $f : B \to B'$ be weakly monotonic and complete. Then*

- $P(f) : P(B) \to P(B')$ *is weakly monotonic and complete;*
- $IV(f) : IV(B) \to IV(B')$ *is weakly monotonic and complete.*

Therefore the two functions $IV(P(f^r)) : IV(P(B)) \to IV(P(B'))$ and $P(IV(f^r) : P(IV(B)) \to P(IV(B'))$ are also weakly monotonic and complete. The following diagram commutes, because it is the naturality diagram for the distributive law in **SET**.

$$\begin{tikzcd}
IV(P(B)) \arrow[r, "a_B"] \arrow[d, "IV(P(f^r))"'] & P(IV(B)) \arrow[d, "P(IV(f^r))"] \\
IV(P(B')) \arrow[r, "a_{B'}"'] & P(IV(B')).
\end{tikzcd}$$

Because of Proposition 2.3.6 the commutativity of the diagram carries over to the diagram

$$\begin{tikzcd}
\mathcal{IV}(\mathcal{P}(D)) \arrow[r, "\alpha_D"] \arrow[d, "\mathcal{IV}(\mathcal{P}(f))"'] & \mathcal{P}(\mathcal{IV}(D)) \arrow[d, "\mathcal{P}(\mathcal{IV}(f))"] \\
\mathcal{IV}(\mathcal{P}(D')) \arrow[r, "\alpha_{D'}"'] & \mathcal{P}(\mathcal{IV}(D')).
\end{tikzcd}$$

For the general case, when the restriction to the bases cannot be found, we notice that a continuous domain is a basis for itself. We make explicit the difference between a domain D and its ideal completion as abstract basis $\bar{D}$. Notice that the isomorphism $\iota_D : D \to \bar{D}$ is the extension of $\iota_B : B \to D$ as

$$\begin{tikzcd}
B \arrow[r, "\iota_B"] \arrow[d, "\iota_B"'] & D \arrow[d, "\iota_D"] \\
D \arrow[r, "\iota_D"'] & \bar{D}
\end{tikzcd}$$

and ι_B is weakly monotonic and complete. Therefore the following diagram commutes

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, leftrightarrow, "\iota_D"'] & D' \arrow[d, leftrightarrow, "\iota_{D'}"] \\
\bar{D} \arrow[r, "\bar{f}"'] & \bar{D}'
\end{tikzcd}$$

The commutativity of the diagram

$$\begin{tikzcd}
IV(P(B)) \arrow[r, "a_B"] \arrow[d, "IV(P(\iota_B))"'] & P(IV(B)) \arrow[d, "P(IV(\iota_B))"] \\
IV(P(D)) \arrow[r, "a_D"] \arrow[d, "IV(P(f))"'] & P(IV(D)) \arrow[d, "P(IV(f))"] \\
IV(P(D')) \arrow[r, "a_{D'}"'] & P(IV(D')) \\
IV(P(B')) \arrow[r, "a_{B'}"] \arrow[u, "IV(P(\iota_{B'}))"] & P(IV(B')) \arrow[u, "P(IV(\iota_{B'}))"']
\end{tikzcd}$$

carries over to the diagram

$$\begin{tikzcd}
\mathcal{IV}(\mathcal{P}(B)) \arrow[r, "\alpha_D"] \arrow[d, leftrightarrow, "\mathcal{IV}(\mathcal{P}(\iota_D))"'] \arrow[dddd, bend right=60, "\mathcal{IV}(\mathcal{P}(f))"'] & \mathcal{P}(\mathcal{IV}(B)) \arrow[d, leftrightarrow, "\mathcal{P}(\mathcal{IV}(\iota_D))"] \arrow[dddd, bend left=60, "\mathcal{IV}(\mathcal{P}(f))"] \\
\mathcal{IV}(\mathcal{P}(\bar{D})) \arrow[r, "\alpha_{\bar{D}}"] \arrow[d, "\mathcal{IV}(\mathcal{P}(\bar{f}))"'] & \mathcal{P}(\mathcal{IV}(\bar{D})) \arrow[d, "\mathcal{P}(\mathcal{IV}(\bar{f}))"] \\
\mathcal{IV}(\mathcal{P}(\bar{D}')) \arrow[r, "\alpha_{\bar{D}'}"'] \arrow[d, leftrightarrow, "\mathcal{IV}(\mathcal{P}(\iota_{D'}))"'] & \mathcal{P}(\mathcal{IV}(\bar{D}')) \arrow[d, leftrightarrow, "\mathcal{P}(\mathcal{IV}(\iota_{D'}))"] \\
\mathcal{IV}(\mathcal{P}(D')) \arrow[r, "\alpha_{D'}"] & \mathcal{P}(\mathcal{IV}(D'))
\end{tikzcd}$$

The four conditions defining a distributive law are proved using the same idea: we use the commutativity of a diagram in the category of sets and functions, which carries over to the diagram constituted by their extensions.

For instance to prove that the diagram

$$\begin{tikzcd}
\mathcal{IV}(\mathcal{IV}(\mathcal{P}(D))) \arrow[r, "\mu^{\mathcal{IV}}_{\mathcal{P}(D)}"] \arrow[d, "\mathcal{IV}(\alpha_D)"'] & \mathcal{IV}(\mathcal{P}(D)) \arrow[dd, "\alpha_D"] \\
\mathcal{IV}(\mathcal{P}(\mathcal{IV}(D))) \arrow[d, "\alpha_{\mathcal{IV}(D)}"'] & \\
\mathcal{P}(\mathcal{IV}(\mathcal{IV}(D))) \arrow[r, "\mathcal{P}(\mu^{\mathcal{IV}}_D)"'] & \mathcal{P}(\mathcal{IV}(D))
\end{tikzcd}$$

commutes, it is enough to prove that the following diagram commutes

$$\begin{tikzcd}
IV(IV(P(B))) \arrow[r, "\mu^{IV}_{P(B)}"] \arrow[d, "IV(a_B)"'] & IV(P(B)) \arrow[dd, "a_B"] \\
IV(P(IV(B))) \arrow[d, "a_{IV(B)}"'] & \\
P(IV(IV(B))) \arrow[r, "P(\mu^{IV}_B)"'] & P(IV(B))
\end{tikzcd}$$

which we have already proved in the category **SET**.

All the other diagrams are proved in the same way. □

4.4.3 Relations with other powerdomains

The above proof could be applicable to other AB-relations on both $P(B)$ and $IV(B)$. The key point is that the function $a_B : IV(P(B)) \to P(IV(B))$ be weakly monotonic and complete. All other conditions come almost for free. If, for example we put the Smyth AB-relation on $P(B)$:

$$X \prec Y \text{ if } \forall y \in Y \exists x \in X.\ x \lhd y$$

then the function a_B is not weakly monotonic. To show this, let $X = \{x_0, x_1\}$ and $Y = \{y_0, y_1\}$ with $x_i \lhd y_i$, so that $X \prec Y$. Suppose moreover that $x_i \not\lhd y_{1-i}$ and that for $i = 0, 1$ there is x'_i such that $x'_i \lhd x_i$ and $x'_i \not\lhd y_{1-i}$ (this is true when, for example, the AB-relation $\lhd$ is a preorder). Then consider $\nu := (X, \frac{1}{2})_{*\in\{*\}}$ and $\xi := (Y, \frac{1}{2})_{j\in\{0,1\}}$, so that $\nu \prec \xi$. Consider $a_B(\nu) = \big\{(x_0, \frac{1}{2})_{*\in\{*\}},\ (x_1, \frac{1}{2})_{*\in\{*\}}\big\}$. And $a_B(\xi) = \big\{(\theta(j), \frac{1}{2})_{j\in\{0,1\}} \mid \theta\colon \{0,1\} \to Y\big\}$ Therefore we have $\big\{(x'_0, \frac{1}{3})_{*\in\{*\}},\ (x'_1, \frac{1}{3})_{*\in\{*\}}\big\} \prec a_B(\nu)$. Also we have that $(y_i, \frac{1}{2})_{i\in\{0,1\}} \in a_B(\xi)$. But $(x'_i, \frac{1}{3})_{*\in\{*\}} \not\prec (y_j, \frac{1}{2})_{j\in\{0,1\}}$ for any i.

The same counterexample shows that a_B is not weakly monotonic also when we put the Egli-Milner AB-relation on $P(B)$:

$$X \prec Y \text{ if } \forall y \in Y \exists x \in X.\ x \lhd y \ \&\ \forall x \in X \exists y \in Y.\ x \lhd y$$

Recall the the Smyth AB-relation generates the Smyth powerdomain, while the Egli-Milner AB-relation generates the Plotkin powerdomain. Consequently, we are not able with our techniques to show there are distributive laws between (Hoare) indexed valuations and either the Smyth or the Plotkin powerdomains.

4.5 Plotkin and Smyth indexed valuations

What happens if we remove the axiom (HV) from our equational theory? Or if we replace it with its dual (SV)? We can still perform analogous constructions, with the interesting exception of Theorem 4.3.1.

4.5.1 Plotkin indexed valuations

We are going to define a different AB-relation on $IV(X)$. This relation corresponds to an equational theory without the axiom (HV).

Definition 4.5.1. Let $(X, \lhd)$ be an abstract basis. For $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \in IV(X)$ define

$$(x_i, p_i)_{i\in I} \prec_P (y_j, q_j)_{j\in J}$$

if and only if there exists a bijection $f : J \to I$, s.t.:

$$x_{f(j)} \lhd y_j \,,$$

$$p_{f(j)} \ll q_j \,.$$

This is essentially the same as definition 4.1.1, except that we require a witness to be injective as well. An alternative definition makes use of the possibility of changing the names of the indices. For $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \in IV(X)$ we have that $(x_i, p_i)_{i\in I} \prec_P (y_j, q_j)_{j\in J}$ if and only if $I = J$ and $x_i \lhd y_i$ and $p_i \ll q_i$.

Proposition 4.5.2. $(IV(X), \prec_P)$ *is an abstract basis.*

Proof: To show transitivity, suppose f is a witness for $(x_i, p_i)_{i\in I} \prec_P (y_j, q_j)_{j\in J}$ and g is a witness for $(y_j, q_j)_{j\in J} \prec_P (z_l, r_l)_{l\in L}$. Then we know, by Proposition 4.1.2, that $g \circ f$ is a witness for $(x_i, p_i)_{i\in I} \prec_P (z_l, r_l)_{l\in L}$. Moreover if both f, g are injective, then $g \circ f$ is also injective.

We have now to show the finite interpolation property. Again we omit the case $|F| = 0$.

Now, let $(x_i, p_i)_{i\in I}, (y_i, q_i)_{i\in I} \prec_P (z_i, r_i)_{i\in I}$ (without loss of generality). Then $x_i, y_i \lhd z_i$ and $p_i, q_i \ll r_i$. Since $(X, \lhd)$ is an abstract basis, for every $i \in I$ there exist z'_i such that $x_i, y_i \lhd z'_i \lhd z_i$. Let $s := \sum_{i\in I} r_i$. Let $2s\epsilon$ be the minimum among all the numbers $r_i - p_i, r_i - q_i$. Consider $(z'_i, (1-\epsilon)r_i)_{i\in I}$. Clearly $(z'_i, (1-\epsilon)r_i)_{i\in I} \prec_P (z_i, r_i)_{i\in I}$. But also $(x_i, p_i)_{i\in I}, (y_i, q_i)_{i\in I} \prec_P (z'_i, (1-\epsilon)r_i)_{i\in I}$
□

Definition 4.5.3. If D is a continuous domain with basis B, let $\mathcal{IV}_P(D)$ be the ideal completion of $(IV(B), \prec_P)$. Its elements are called *Plotkin indexed valuations.*

Proposition 4.5.4. *If*

$$(b_i, p_i)_{i\in I} \prec_P (c_j, q_j)_{j\in J} \,,$$

then for every j there exist $c'_j \ll c_j$ and $q'_j \ll q_j$ such that

$$(b_i, p_i)_{i\in I} \prec_P (c'_j, q'_j)_{j\in J} \prec_P (c_j, q_j)_{j\in J} \ .$$

The operations on $IV(X)$ respect the AB-relation

Lemma 4.5.5. *If ν, ξ, ν', ξ are finite indexed valuation on B,*

a) if $\nu \prec_P \underline{0}$ then $\nu = \underline{0}$;

b) if $\nu \prec_P \nu'$ then $p\nu \prec_P p\nu'$;

c) if $\nu \prec_P \nu'$ & $\xi \prec_P \xi'$ then $\nu \oplus \xi \prec_P \nu' \oplus \xi'$.

4.5.2 Plotkin indexed valuations as a free construction

We present the equational theory characterising Plotkin indexed valuations.

Definition 4.5.6. A *continuous Plotkin quasi-cone* is a structure $(D, \sqsubseteq, \oplus, \odot)$ such that

- $(D, \sqsubseteq)$ is a continuous domain;
- $\oplus : D \times D \to D$ is continuous;
- $\odot : [0, +\infty[\times D \to D$ is continuous
- axioms (1)–(7) are satisfied.

Let **CONT** be the category of continuous domains, and **QPCONT** be the category of continuous Plotkin quasi-cones and continuous homomorphisms.

Proposition 4.5.7. *If D is a continuous domain then $\mathcal{IV}_P(D)$ is a continuous Plotkin quasi-cone.*

Proposition 4.5.8. *The operator $\mathcal{IV}_P$ extends to a functor* **CONT** $\to$ **QPCONT** *which is left adjoint to the forgetful functor.*

Proof: For every continuous function $g : D \to \mathcal{E}$ where $\mathcal{E} \in$ **QPCONT** there is a unique $g^\dagger : \mathcal{IV}_P(D) \to \mathcal{E}$ (in the category **QPCONT**) s.t.

$$\begin{tikzcd}
D \arrow[d, "\eta"'] \arrow[dr, "g"] & \\
\mathcal{IV}(D) \arrow[r, dashed, "g^\dagger"'] & \mathcal{E}.
\end{tikzcd}$$

where $\eta(d) = \{(b, p)_{* \in \{*\}} \,|\, b \ll d,\, p < 1\}$.

Take the restriction of g to B_D. It has a unique homomorphic extension $\overline{g} : IV(B_D) \to \mathcal{E}$, defined by

$$\begin{aligned}
\overline{g}\left((b, 1)_{* \in \{*\}}\right) &:= g(b)\,; \\
\overline{g}\left((b_i, p_i)_{i \in I}\right) &:= \bigoplus_{i \in I} p_i g(b_i)\,.
\end{aligned}$$

We claim that $\overline{g}$ is monotonic, in the sense that if $\nu \prec_P \xi$, then $\overline{g}(\nu) \sqsubseteq \overline{g}(\xi)$. First suppose that $(b, p)_{* \in \{*\}} \prec_P (c, q)_{* \in \{*\}}$. Then $p \ll q$ and $b \ll c$. By monotonicity of scalar multiplication and of g,

$$\overline{g}\left((b, p)_{* \in \{*\}}\right) = pg(b) \sqsubseteq qg(c) = \overline{g}\left((c, q)_{* \in \{*\}}\right)\ .$$

Now suppose $(b_i, p_i)_{i\in I} \prec_P (c_i, q_i)_{i\in I}$. Clearly $(b_i, p_i)_{*\in\{*\}} \prec_P (c_i, q_i)_{*\in\{*\}}$ for every $i \in I$. Therefore

$$\overline{g}\left((b_i, p_i)_{*\in\{*\}}\right) \sqsubseteq \overline{g}\left((c_i, q_i)_{*\in\{*\}}\right) \ .$$

Notice that $(b_i, p_i)_{i\in I} = \bigoplus_{i\in I}(b_i, p_i)_{*\in\{*\}}$ and $(c_i, q_i)_{i\in I} = \bigoplus_{i\in I}(c_i, q_i)_{*\in\{*\}}$. Monotonicity of the sum, and the homomorphism condition on $\overline{g}$ imply that

$$\overline{g}\left((b_i, p_i)_{i\in I}\right) \sqsubseteq \overline{g}\left((c_i, q_i)_{i\in I}\right) \ .$$

Let us call $g^\dagger$ the extension of $\overline{g}$ to $\mathcal{IV}(D)$ (the ideal completion of $(IV(B_D))$. We recall that $g^\dagger(\mathcal{I}) := \bigsqcup^{\uparrow}{}_{\nu\in\mathcal{I}}\, \overline{g}(\nu)$. We know that the function $g^\dagger$ is continuous. The continuity of the operations implies that $g^\dagger$ is also an homomorphism. Thus it is a morphism of the category.

It remains to show that $g^\dagger\big(\{(b, p)_{*\in\{*\}} \,|\, b \ll d,\, p < 1\}\big) = g(d)$

Now

$$g^\dagger\big(\{(b, p)_{*\in\{*\}} \,|\, b \ll d,\, p < 1\}\big) = \bigsqcup_{b\ll d,\, p<1}^{\uparrow} \overline{g}((b, p)_{*\in\{*\}})$$

$$= \bigsqcup_{b\ll d,\, p<1}^{\uparrow} pg(b) = \bigsqcup_{p<1}^{\uparrow} pg(d) = g(d)\,.$$

The last two equalities being a consequence of the continuity of g and of the scalar multiplication.

To prove uniqueness we need the following lemma

Lemma 4.5.9. *If ν, ξ are finite indexed valuation on B,*

a) $\iota(\underline{0}) = \underline{0}$;

b) $p(\iota(\nu)) = \iota(p\nu)$;

c) $\iota(\nu) \oplus \iota(\xi) = \iota(\nu \oplus \xi)$.

As a consequence, for every $\nu \in IV(B_D)$, $\overline{g}(\nu) = g^\dagger(\iota(\nu))$.

$$\begin{tikzcd}
IV(B_D) \arrow[d, "\iota"'] \arrow[dr, "\overline{g}"] & \\
\mathcal{IV}(D) \arrow[r, dashed, "g^\dagger"'] & \mathcal{E}.
\end{tikzcd}$$

Let $h : \mathcal{IV}_P(D) \to \mathcal{E}$ be a continuous homomorphism such that for every $d \in D$, $h(\eta(d)) = g(d)$. Since h is an homomorphism, we have that when for every $(b_i, p_i)_{i\in I} \in IV(B_D)$

$$h(\iota\big((b_i, p_i)_{i\in I}\big)) = \bigoplus_{i\in I} p_i h(\iota\big((b_i, 1)_{*\in\{*\}}\big)) = \bigoplus_{i\in I} p_i h(\eta(b_i)) = \bigoplus_{i\in I} p_i g(b_i)$$

$$= \overline{g}\big((b_i, p_i)_{i\in I}\big) = g^\dagger\big(\downarrow(b_i, p_i)_{i\in I}\big)$$

Since h and $g^\dagger$ coincide on the basis, they are equal. □

4.5.3 Failure of surjectivity

The universal property proved above gives us the function $\mathit{Flat}_D : \mathcal{IV}_P(D) \to \mathcal{V}(D)$ as the extension of $\eta : D \to \mathcal{V}_P(D)$.

Interestingly, in this case Flat_D is not surjective in general, and therefore there is no insertion-closure pair. To show this, let D be the domain of finite and infinite sequences over a two letter alphabet $D := \{a, b\}^\infty$. It is an algebraic domain, whose compact elements are the finite sequences. Consider the following chain of continuous valuations on D.

$$\nu_0 = \eta_\epsilon$$

$$\nu_1 = \frac{1}{2}\eta_a + \frac{1}{2}\eta_b$$

$$\nu_n = \sum_{|\sigma|=n} \frac{1}{2^n}\eta_\sigma$$

The limit of this chain is a valuation ν_∞ such that for every open set of the form $\uparrow\sigma$, $\nu_\infty(\uparrow\sigma) = \frac{1}{2^{|\sigma|}}$. Suppose, by contradiction, that $\mathcal{I}$ is an ideal such that $\mathit{Flat}(\mathcal{I}) = \nu_\infty$. Then $\nu_\infty = \bigsqcup^\uparrow_{\zeta\in\mathcal{I}} \mathit{Flat}(\zeta)$. This means that for every $\xi \ll \nu_\infty$ there exists $\zeta \in \mathcal{I}$ such that $\xi \sqsubseteq \mathit{Flat}(\zeta) \sqsubseteq \nu_\infty$. Let $\xi_1 := (1-\varepsilon)\eta_\epsilon$. We have that $\xi_1 \ll \nu_\infty$. Let $\zeta_1 \in \mathcal{I}$ such that $\xi_1 \sqsubseteq \mathit{Flat}(\zeta_1) \sqsubseteq \nu_\infty$. Say $\zeta_1 = (\tau_i, p_i)_{i\in I}$, with $p_i > 0$ for all i. Notice that $\mathit{Flat}(\zeta)(D) = \sum_{i\in I} p_i$. Therefore $1 \geq \sum_{i\in I} p_i \geq (1-\varepsilon)$. Let n be such that $\min_{i\in I} p_i > \frac{1}{2^n}$. This implies that $|I| < 2^n$.

Consider now

$$\xi_2 := \sum_{|\sigma|=n+1} \frac{(1-\varepsilon)}{2^{n+1}}\eta_\sigma$$

We have that $\xi_2 \ll \nu_\infty$. Let $\zeta_2 \in \mathcal{I}$ such that $\xi_2 \sqsubseteq \mathit{Flat}(\zeta_2) \sqsubseteq \nu_\infty$. Say $\zeta_2 = (\tau'_j, p'_j)_{j\in J}$, with $p'_j > 0$ for all $j \in J$. We must have that $|J| \geq 2^{n+1}$. This is because for every $|\sigma| = n$ there must exist some $j \in J$ with $\tau'_j \sqsupseteq \sigma$. (And for different σ the corresponding τ'_j must be different). If there were a $\bar{\sigma}$ contradicting this, then $\mathit{Flat}(\zeta_2)(\uparrow\bar{\sigma}) = 0$, while $\xi_2(\uparrow\bar{\sigma}) = \frac{(1-\varepsilon)}{2^{n+1}}$, which contradicts $\xi_2 \sqsubseteq \mathit{Flat}(\zeta_2)$. For the same reason, if $J_\sigma := \{j \in J \mid \tau'_j \sqsupseteq \sigma\}$, then

$$\sum_{j\in J_\sigma} p'_j \geq \frac{(1-\varepsilon)}{2^{n+1}}$$

Notice also that $J = \bigcup_\sigma J_\sigma$. Since $\mathcal{I}$ is an ideal, there is $\zeta \in \mathcal{I}$ such that $\zeta_1, \zeta_2 \prec_P \zeta$. Notice that since $\zeta \in \mathcal{I}$, we must have $\mathit{Flat}(\zeta) \sqsubseteq \nu_\infty$. Pick one such ζ, say $\zeta = (\tau''_k, p''_k)_{k\in K}$ with $p''_k > 0$ for all $k \in K$.

Rename the indices of ζ_1, ζ_2 and add indices with weight 0 so that we have

$$\zeta_1 = (\tau_k, p_k)_{k\in K}$$

$$\zeta_2 = (\tau'_k, p'_k)_{k\in K}$$

$$\zeta = (\tau''_k, p''_k)_{k\in K}$$

with $p''_k > p'_k, p_k$ for all $k \in K$.

Define $K_1 := \{k \in K \mid p_k > 0\}$ and $K_2 := \{k \in K \mid p'_k > 0\}$. Finally define $K_\sigma := \{k \in K_2 \mid \tau'_k \sqsupseteq \sigma\}$. Modulo the renaming we have $K_1 = I, K_2 = J$ and $K_\sigma = J_\sigma$. In particular $|K_1| < 2^n, |K_2| \geq 2n + 1$,

$$\sum_{k \in K_\sigma} p''_k > \sum_{k \in K_\sigma} p'_k \geq \frac{(1-\varepsilon)}{2^{n+1}}$$

and

$$\sum_{k \in K_1} p''_k > \sum_{k \in K_1} p_k \geq (1-\varepsilon)\,.$$

Now we have

$$\begin{aligned} 1 &\geq \sum_{k \in K} p''_k \\ &\geq \sum_{k \in K_1} p''_k + \sum_{k \in K_2 \setminus K_1} p''_k \\ &= \sum_{k \in K_1} p''_k + \sum_{|\sigma|=n} \sum_{k \in K_\sigma \setminus K_1} p''_k \end{aligned}$$

Since $|K_1| < 2^n$, and all K_σ are pairwise disjoint, for more than 2^n many σ's, we have that $K_\sigma \setminus K_1 = K_\sigma$. Therefore

$$\begin{aligned} &\sum_{|\sigma|=n} \sum_{k \in K_\sigma \setminus K_1} p''_k \\ &\geq \sum_{K_\sigma \setminus K_1 = K_\sigma} \sum_{k \in K_\sigma \setminus K_1} p''_k \\ \geq &\sum_{K_\sigma \setminus K_1 = K_\sigma} \frac{(1-\varepsilon)}{2^{n+1}} \geq 2^n \frac{(1-\varepsilon)}{2^{n+1}} = \frac{(1-\varepsilon)}{2} \end{aligned}$$

Continuing the main chain of inequalities we have

$$\begin{aligned} 1 &\geq \sum_{k \in K_1} p''_k + \frac{(1-\varepsilon)}{2} \\ &\geq (1-\varepsilon) + \frac{(1-\varepsilon)}{2} = \frac{3}{2} - 2\varepsilon \end{aligned}$$

And for ε small enough this is a contradiction.

4.5.4 Smyth indexed valuations

We can define a third notion of AB relation. Let $(X, \lhd)$ be an abstract basis.

Definition 4.5.10. Let $(X, \lhd)$ be an abstract basis. For $(x_i, p_i)_{i \in I}, (y_j, q_j)_{j \in J} \in IV(X)$

$$(x_i, p_i)_{i \in I} \prec_S (y_j, q_j)_{j \in J}$$

if and only if there exists a function $f : I \to J$, s.t.:

$$x_i \lhd y_{f(i)}\,,$$

$$\sum_{f(i)=j} p_i \ll q_j\,.$$

Proposition 4.5.11. *$(IV(X), \prec_S)$ is an abstract basis.*

Definition 4.5.12. If D is a continuous domain with basis B, let $\mathcal{IV}_S(D)$ be the ideal completion of $(IV(B), \prec_S)$. Its elements are called *Smyth indexed valuations*.

Definition 4.5.13. A *continuous Smyth quasi-cone* is a structure $(D, \sqsubseteq, \oplus, \odot)$ such that

- $(D, \sqsubseteq)$ is a continuous domain;
- $\oplus : D \times D \to D$ is continuous;
- $\odot : [0, +\infty[\times D \to D$ is continuous;
- axioms (1)–(7) are satisfied;
- the following axiom (SV) is satisfied: $pA \oplus qA \sqsubseteq (p+q)A$.

Let **CONT** be the category of continuous domains, and **QSCONT** be the category of continuous Smyth quasi-cones and continuous homomorphisms.

Proposition 4.5.14. *If D is a continuous domain then $\mathcal{IV}_S(D)$ is a continuous Smyth quasi-cone.*

Proposition 4.5.15. *The operator $\mathcal{IV}_S$ extends to a functor* **CONT** $\to$ **QSCONT** *which is left adjoint to the forgetful functor.*

Again, the transformation *Flat* : $\mathcal{IV} \to \mathcal{V}$ is not surjective, the counterexample being the one shown above.

4.5.5 Comparing different indexed valuations

Plotkin and Smyth indexed valuations are less interesting than Hoare indexed valuations, because not all continuous valuations have a Plotkin or Smyth "representative".

How about the distributive law? We just observe in which cases the function $a_B : IV(P(B)) \to P(IV(B))$ is strongly monotonic and complete. This is the core of the proof of the existence of the distributive law. The other details are basically the same.

Lemma 4.5.16. *Let $\langle B, \lhd \rangle$ be an abstract basis and $IV(B)$ be endowed with $\prec_P$. Let $P(B)$ be endowed with one of the three AB-relations: Hoare, Egli-Milner, Smyth. In all three cases the function $a_B : IV(P(B)) \to P(IV(B))$ is strongly monotonic and complete.*

Lemma 4.5.17. *Let $\langle B, \lhd \rangle$ be an abstract basis and $IV(B)$ be endowed with $\prec_S$. Let $P(B)$ be endowed with the Smyth AB-relation. Then the function $a_B : IV(P(B)) \to P(IV(B))$ is strongly monotonic and complete. If $P(B)$ is endowed with the Hoare or the Egli-Milner AB-relation, then a_B is not weakly monotonic.*

The following table shows in which cases a_B is strongly monotonic and complete.

	Hoare Indexed Valns.	Plotkin Indexed Valns.	Smyth Indexed Valns.
Hoare Powerdomain	Y	Y	N
Plotkin Powerdomain	N	Y	N
Smyth Powerdomain	N	Y	Y

We can conclude by saying that the case we have studied in detail is the most interesting, because:

- Hoare indexed valuations are the only ones that project surjectively onto continuous valuations.
- We can prove a distributive law only between Hoare indexed valuations and the Hoare powerdomain.

We have not ruled out the possibility that other distributive laws exist between Hoare indexed valuations and the other two powerdomains, but even if they existed, we feel they would not correspond to the equational distributivity.

4.6 In search of a concrete characterisation

The Hoare powerdomain, and the powerdomain of valuations are freely generated by an equational theory, but they have also an alternative, more concrete, characterisation. It would be interesting to find an analogous characterisation of indexed valuations. Unfortunately our attempts have not been successful so far. We outline in this section the ideas we had and the difficulties we have encountered.

4.6.1 The leading idea

Our intuition is the following. Finite valuations on a set X are in fact continuous valuations on the discrete topology of X. A finite indexed valuation is an indexing function together with a finite valuation on the indexing set. We could then generalise this and provide the indexing set with a topology. An indexed valuation on a topological space X would then be a continuous indexing function together with a continuous valuation on the indexing set. Remember, though, that finite indexed valuations are defined up to the equivalence relation $\sim$. We have to define a analogous relation on general indexed valuations. Bijective renaming is not a problem. More problematic is the irrelevance of indices with weight 0. One possibility is to define the support of a valuation and then to identify valuations with homeomorphic supports.

If a ν is a continuous valuation on the topological space (X, τ), we define the *irrelevance set* to be the union of all open sets of weight 0. We denote it by $Irr(\nu)$. The irrelevance set is open. The set of open sets of weight 0 is directed, because if O, O' have weight 0, then $O \cup O'$ has also weight 0 by modularity. By continuity the irrelevance set has also weight 0, so that it is the biggest open set of weight 0. The *support* of ν is the complement of its irrelevance set. We denote it by $\mathit{Supp}(\nu)$. An open set O of $\mathit{Supp}(\nu)$ is of the form $O' \cap \mathit{Supp}(\nu)$ for some $O' \in \tau$. The restriction of ν to its support is the function $\hat{\nu}$ defined by $\hat{\nu}(O) = \nu(O \cup \mathit{Irr}(\nu))$. Note that $O \cup \mathit{Irr}(\nu)$ is open in τ because for any O' with $O = O' \cap \mathit{Supp}(\nu)$ we have that $O \cup \mathit{Irr}(\nu) = O' \cup \mathit{Irr}(\nu)$. It is easy to check that $\hat{\nu}$ is a valuation on $\mathit{Supp}(\nu)$.

The problem is now to put an order on indexed valuations and obtain a continuous domain. On finite indexed valuations the order is defined using witnesses, which are surjective functions satisfying some properties. In the topological context we can define the witnesses to be surjective continuous functions satisfying some properties. Let's try to formalise these ideas.

An indexed valuation on a topological space D is given by a valuation ν on a topological space $\langle X, \tau \rangle$ together with a continuous function $f : X \to D$. If $\zeta = (\nu, X, f)$ and $\zeta = (\nu', X', f')$ are two indexed valuations on D we say that $\zeta \sqsubseteq \zeta'$ if there exists a continuous surjective function $h : X' \to X$ such that for every open subset O of X,

- $f(O) \supseteq f'(h^{-1}(O))$;
- $\nu(O) \leq \nu'(h^{-1}(O))$.

Clearly this defines a preorder, but is it a DCPO? What is the limit of a chain of indexed valuations? The idea is to construct the topological space by taking the limit in the category **TOP** of topological spaces and similarly define the valuation at the limit. Unfortunately a chain of indexed valuations could be witnessed by different functions. We should be able to prove that the limits of two different witnessing chains are, in some sense, equivalent. We argue next why this is not possible.

4.6.2 A counterexample

Consider the following counterexample. The topological space D is just a singleton. We build a chain of finite indexed valuations ν_n (defined on discrete finite topologies). Since the indexing function is always trivial in this case, we will not mention it. For every n we have $X_n = I_n \cup \{*\}$, with $p(n) = \frac{1}{2^n}$ and $p(*) = 1$. Clearly $\nu_n \sqsubseteq \nu_{n+1}$ but there are two choices of witness. The first choice is $f_n : X_{n+1} \to X_n$ defined as $f_n(n) = n, f_n(*) = *, f_n(n+1) = n$.

$$\begin{tikzcd}
{} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[dl, dashed] & {} \arrow[d, dashed] \\
1 \arrow[d] & 2 \arrow[d] & 3 \arrow[d] & 4 \arrow[dl] & & * \arrow[d] \\
1 \arrow[d] & 2 \arrow[d] & 3 \arrow[dl] & & & * \arrow[d] \\
1 \arrow[d] & 2 \arrow[dl] & & & & * \arrow[d] \\
1 & & & & & *
\end{tikzcd}$$

The second choice is $g_n : X_{n+1} \to X_n$ defined as $g_n(n) = n, g_n(*) = *, g_n(n+1) = *$.

$$\begin{tikzcd}
{} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[d, dashed] & {} \arrow[dr, dashed] & {} \arrow[d, dashed] \\
1 \arrow[d] & 2 \arrow[d] & 3 \arrow[d] & 4 \arrow[drr] & & * \arrow[d] \\
1 \arrow[d] & 2 \arrow[d] & 3 \arrow[drrr] & & & * \arrow[d] \\
1 \arrow[d] & 2 \arrow[drrrr] & & & & * \arrow[d] \\
1 & & & & & *
\end{tikzcd}$$

The limit of the chain f_n is $\mathbb{N} \cup \{*\}$ with the discrete topology, while the limit of the chain g_n is $\mathbb{N} \cup \{*\}$, where the open neighbours of $*$ are the sets of the form $\{m \mid m > n\} \cup \{*\}$. In both cases the limit valuation is (basically) defined as $p_\infty(n) = \frac{1}{2^n}, p_\infty(*) = 1$, so that every non empty set has positive weight. The two limit valuations do not have homeomorphic support. Note, though, that they generate the same Borel measures.

One idea could be to use measures instead of valuations, but this may force us to work in a smaller category than **CONT** as it is not known whether all continuous valuations on a continuous domain extend to Borel measures.

4.7 Layered indexed valuations

An indexed valuation $\nu = (x_i, p_i)_{i\in I}$ is less than $\xi = (y_j, q_j)_{j\in J}$ if the indices in ν split into the indices of ξ, the corresponding elements increase, and *the weights* increase. In giving semantics to the language, we never use the latter feature. Notice also that the counterexample at the end of the previous section uses the possibility of increasing the weights.

Achim Jung observed that, while the splitting of indices and the increment of the value has an intuitive interpretation, the increment of weights has not. Indices represent computations, and the order relation represents the amount of information we have. One way of obtaining more information is by letting the computation progress. Splitting indices represents the forking of a computation into several more, because we have performed some probabilistic choice. The probability of a computation is split among all its possible futures, but it does not increase as a whole.

When we use normal valuations we must allow the increase of weights, because different computations may take different paths, or progress at different speed, and yet give the same observation. Therefore the weight of the observation may increase in time. But the weight of the computation does not increase as time progresses.

The possibility of increasing the weights is matched, equationally, by the continuity in the first argument of the scalar multiplication. Removing this requirement generates a different functor, which has the extra nice property of preserving the category of algebraic domains.

4.7.1 Definition

Let $(X, \lhd)$ be an abstract basis. We define a relation $\prec$ on $IV(X)$ in such a way that $(IV(X), \prec)$ is an abstract basis:

Definition 4.7.1. For $(x_i, p_i)_{i\in I}, (y_j, q_j)_{j\in J} \in IV(X)$

$$(x_i, p_i)_{i\in I} \prec (y_j, q_j)_{j\in J}$$

if and only if there exists a total surjective function $f : J \to I$, s.t.:

$$x_{f(j)} \lhd y_j\,,$$

$$p_i = \sum_{f(j)=i} q_j\,.$$

Note that in this case it is not equivalent to use partial functions, because we cannot increase the weights of indices, hence we cannot add dummy indices with 0 weight to transform a partial function into a total function.

Proposition 4.7.2. $(IV(X), \prec)$ *is an abstract basis.*

Definition 4.7.3. If D is a continuous domain generated by the abstract basis B, let $\mathcal{IV}_l(D)$ be the ideal completion of $(IV(B), \prec)$. Its elements are called *Layered indexed valuations.*

Note that, if $(B, \lhd)$ is a partial order, then so is $(IV(B)), \prec)$. Therefore the functor $\mathcal{IV}_l(D)$ preserves the category of algebraic domains.

Definition 4.7.4. A *layered continuous quasi-cone* is a structure $(D, \sqsubseteq, \oplus, \odot)$ such that

- $(D, \sqsubseteq)$ is a continuous domain;
- $\oplus : D \times D \to D$ is continuous;
- for every $p \in [0, +\infty[$ the function $\lambda d \in D.pd$ is continuous;
- axioms (1)–(7) + (HV) are satisfied.

Let **CONT** be the category of continuous domains, and **LQCONT** be the category of layered continuous quasi-cones and continuous homomorphisms.

Proposition 4.7.5. *If D is a continuous domain then $\mathcal{IV}_l(D)$ is a layered continuous quasi-cone.*

Proposition 4.7.6. *The operator $\mathcal{IV}_l$ extends to a functor* **CONT** $\to$ **LQCONT** *which is left adjoint to the forgetful functor.*

Proof: For every continuous function $g : D \to \mathcal{E}$ where $\mathcal{E} \in$ **LQCONT** there is a unique $g^\dagger : \mathcal{IV}(D) \to \mathcal{E}$ (in the category **LQCONT**) s.t.

$$\begin{tikzcd}
D \arrow[d, "\eta"'] \arrow[dr, "g"] & \\
\mathcal{IV}(D) \arrow[r, dashed, "g^\dagger"'] & \mathcal{E}.
\end{tikzcd}$$

where $\eta(d) = \{(b,1)_{*\in\{*\}} \,|\, b \in d\}$.

Take the "restriction" of g to B_D, defined as $g(b) = g(\iota_B(b))$. It has a unique homomorphic extension $\overline{g} : IV(B_D) \to \mathcal{E}$. Defined by

$$\begin{aligned} \overline{g}\left((b,1)_{*\in\{*\}}\right) &:= g(b)\,; \\ \overline{g}\left((b_i,p_i)_{i\in I}\right) &:= \bigoplus_{i\in I} p_i g(b_i)\,. \end{aligned}$$

Lemma 4.7.7. *if $\nu \prec \xi$, then $\overline{g}(\nu) \sqsubseteq \overline{g}(\xi)$.*

Proof: (of the lemma) First suppose that $(b,p)_{*\in\{*\}} \prec (c_j,q_j)_{j\in J}$. Then $p = \sum_{j\in J} q_j$ and for every j, $b \lhd c_j$. Applying iteratively the inequation (HV), we can show that:

$$\overline{g}\left((b,p)_{*\in\{*\}}\right) = pg(b) \sqsubseteq \bigoplus_{j\in J} q_j g(b)\,.$$

Then, by monotonicity of the operations, of g and of ι_B,

$$\bigoplus_{j\in J} q_j g(b) \sqsubseteq \bigoplus_{j\in J} q_j g(c_j) = \overline{g}\left((c_j,q_j)_{j\in J}\right)\,.$$

Now suppose $(b_i,p_i)_{i\in I} \prec (c_j,q_j)_{j\in J}$ with witness f. Let $J_i = f^{-1}(i)$. Clearly $(b_i,p_i)_{*\in\{*\}} \prec (c_j,q_j)_{j\in J_i}$ for every $i \in I$. Therefore

$$\overline{g}\left((b_i,p_i)_{*\in\{*\}}\right) \sqsubseteq \overline{g}\left((c_j,q_j)_{j\in J_i}\right)\,.$$

Notice that $(b_i,p_i)_{i\in I} = \bigoplus_{i\in I}(b_i,p_i)_{*\in\{*\}}$ and $(c_j,q_j)_{j\in J} = \bigoplus_{i\in I}(c_j,q_j)_{j\in J_i}$. Monotonicity of the sum, and the homomorphism condition on $\overline{g}$ imply that

$$\overline{g}\left((b_i,p_i)_{i\in I}\right) \sqsubseteq \overline{g}\left((c_j,q_j)_{j\in J}\right)\,.$$

□(lemma)

Note again that we cannot derive $\underline{0} \sqsubseteq A$, so the domain $\mathcal{IV}(D)$ does not have a minimum, in general.

Let us call $g^\dagger$ the extension of $\overline{g}$ to $\mathcal{IV}_l(D)$ (the ideal completion of $(IV(B_D))$. We recall that $g^\dagger(\mathcal{I}) := \bigsqcup^{\uparrow}{}_{\nu\in\mathcal{I}}\, \overline{g}(\nu)$. We know that the function $g^\dagger$ is continuous. The continuity of the operations implies that $g^\dagger$ is also an homomorphism. Thus it is a morphism of the category.

It remains to show that $g^\dagger\left(\{(b,1)_{*\in\{*\}} \,|\, b \in d\}\right) = g(d)$.

Now

$$\begin{aligned} g^\dagger\left(\{(b,1)_{*\in\{*\}} \,|\, b \in d\}\right) &= \bigsqcup_{b\in d}{}^{\uparrow}\, \overline{g}((b,1)_{*\in\{*\}}) \\ &= \bigsqcup_{b\in d}{}^{\uparrow}\, g(b) = g(d)\,. \end{aligned}$$

Uniqueness is proved in the usual way. □

We observe that

Lemma 4.7.8. *If $\langle B, \lhd\rangle$ is an abstract basis $P(B)$ is endowed with the Hoare AB-relation and $IV(B)$ is endowed with $\prec$, then the function $a_B : IV(P(B)) \to (P(IV(B))$ is strongly monotonic and complete.*

We therefore have a distributive law between $\mathcal{IV}_l$ and $\mathcal{P}_H$.

4.7.2 Relation with Normalised Valuations

The domain of layered indexed valuations is structured in layers. Every layer is characterised by the weight assigned to the whole domain. Elements belonging to different layers are not comparable. It therefore makes sense to study only the "normal" layer, because the others are obtained by scalar multiplication. In [Eda95a], Edalat studies the notion of normalised valuations. We can compare normalised layered indexed valuations with normalised valuations and ask whether a result holds which correspond to theorem 4.3.1. We can still define $\mathit{Flat}_D : \mathcal{IV}_l(D) \to \mathcal{V}(D)$. Is such a function surjective onto $\mathcal{V}^1(D)$? The answer is no in general. Take an infinite set X with the flat order. Then every normalised layered indexed valuation has finite support.

If the domain has a bottom, the answer could be yes. The idea is that instead of increasing the weights of the indices, we could generate new indices that take their weight from the bottom. Note though that the way below relations don't match.

Recall that the way below relation on $\mathcal{V}^1(D)$ has the following property. Suppose D has a bottom element $\perp$. Then For two simple valuations $\nu := \sum_{b\in B} r_b\eta_b$ and $\xi := \sum_{c\in C} s_c\eta_c$ in $\mathcal{V}^1(D)$ we have that $\nu \ll \xi$ if and only if $\perp \in B$ with $r_\perp \neq 0$ and there exists "transport numbers" $t_{b,c}$ such that

- $t_{\perp,c} \neq 0$;
- $\sum_{c\in C} t_{b,c} = r_b$;
- $\sum_{b\in B} t_{b,c} \ll s_c$;
- $t_{b,c} > 0 \Longrightarrow b \sqsubseteq c$.

Therefore in general $\nu \prec \xi$ does not imply $\mathit{Flat}(\nu) \ll \mathit{Flat}(\xi)$.

4.8 The Hoare convex powercone of Tix and Mislove

Recall the theory (1)–(12)+(HV)+(HP), which corresponds to the combination of the Hoare indexed valuation with the Hoare powerdomain. If instead of the axiom (HV) we include the more standard

13. $(p + q)A = (pA \oplus qA)$

the resulting free construction is the one of Tix [Tix99] and Mislove [Mis00], which still includes the equational distributive laws, but without any corresponding categorical distributive law.

We recall that a *continuous d-cone* is a continuous quasi-cone satisfying $(p + q)A = pA \oplus qA$. The corresponding category is called **CCONE**.

Definition 4.8.1. A *continuous join TM-cone* is a continuous domain algebra for the theory (1)–(13)+(HP) for which the scalar multiplication is continuous also in the first argument. The corresponding category is called **CJTM**. A subset X of a continuous d-cone is *convex* if for every $x, y \in X$ and for every $p \in [0, 1]$, we have that $px \oplus (1-p)y \in X$. If H is a continuous d-cone, we define

$$\mathcal{P}_{TM}(H) := \{X \subseteq H \mid X \neq \emptyset, \text{ convex, Scott closed}\}$$

Tix calls this construction the *convex Hoare powercone*.

With the sets ordered by inclusion, the addition and multiplication defined (essentially) pointwise, and the union defined as union followed by convex closure and by topological closure, it is shown that $\mathcal{P}_{TM}(H)$ is a continuous TM-cone. In fact, Tix defines the $A \uplus B$ as the least upper bound of $\{A, B\}$. But her definition is equivalent to ours. On the one hand, the least upper bound operation is associative, commutative, idempotent, continuous and satisfies the Hoare inequality. On the other hand, if $\uplus$ is defined as an associative, commutative, idempotent, continuous operation satisfying the Hoare inequality, then $A \uplus B$ is indeed the least upper bound of $\{A, B\}$.

Jones (and Kirch for our setting) showed that the powerdomain of valuations functor $\mathcal{V} : \mathbf{CONT} \to \mathbf{CCONE}$ is left adjoint of the forgetful functor. Tix in her thesis showed that the functor $\mathcal{P}_{TM} : \mathbf{CCONE} \to \mathbf{CJTM}$ is left adjoint of the forgetful functor. Therefore the functor $\mathcal{P}_{TM} \circ \mathcal{V} : \mathbf{CONT} \to \mathbf{CJTM}$ is left adjoint of the forgetful functor.

We observe that if B is a basis for D, finite valuations on B with the AB-relation induced by the splitting lemma are a basis for $\mathcal{V}(D)$. If B is a basis for the continuous d-cone H, finitely generated convex subsets of $\overline{B}$ with a Hoare-like AB-relation are a basis for $\mathcal{P}_{TM}(H)$. We are now going to prove that formally.

Let B be a basis for the continuous domain D. Consider the set $V(B)$ endowed with the standard addition and scalar multiplication. Note that every finite valuation ν can be written as

$$\nu = \bigoplus_{b \in Supp(\nu)} \nu(b)\eta_b \,.$$

Define $\nu \prec \xi$ if for every $b \in Supp(\nu), c \in Supp(\xi)$ there exist $t_{b,c} \in \overline{\mathbb{R}^+}$ such that

$$\sum_{c \in Supp(\xi)} t_{b,c} = \nu(b) \,,$$

$$\sum_{b \in Supp(\nu)} t_{b,c} \ll \xi(c) \,,$$

and $t_{b,c} \neq 0 \Longrightarrow b \ll c$.

Proposition 4.8.2. *The structure $\langle V(B), \prec \rangle$ is an abstract basis. Its ideal completion $Idl(V(B))$ is isomorphic to $\mathcal{V}(D)$ in **CCONES**. Moreover $\iota : V(B) \to Idl(V(B))$ is an homomorphism of real cones.*

Proof: Consider the set of simple valuations in $\mathcal{V}(D)$ which are defined using elements of B only. We call this set $\mathcal{S}(D)$. The function

$$j(\nu) = \bigoplus_{b \in Supp(\nu)} \nu(b)\eta_b \,,$$

define a bijection j between $V(B)$ and $\mathcal{S}(D)$ such that $j(\nu) \ll j(\xi)$ if and only if $\nu \prec \xi$. Bijectivity is straightforward, while the second property follows from the splitting lemma. The extension of j defines a continuous bijection between $Idl(V(B))$ and $\mathcal{V}(D)$. We have just to check that it is a homomorphism. First we notice that j is an homomorphism. Then $j^\dagger$ is an homomorphism because of the continuity of the operations in $\mathcal{V}(D)$. Indeed

$$j^\dagger\big(I \oplus J\big) = \bigsqcup^{\uparrow}_{\zeta \in (I \oplus J)} j(\zeta)$$

Recall that j preserves the AB-relation. This, and the roundness of I, J imply that the upper bounds of $\{j(\zeta) \,|\, \zeta \in (I \oplus J)\}$ are exactly the same as the upper bounds for $\{j(\nu) \oplus j(\xi) \,|\, \nu \in I, \xi \in J\}$. Therefore

$$\bigsqcup^{\uparrow}_{\zeta \in (I \oplus J)} j(\zeta) = \bigsqcup^{\uparrow}_{\nu \in I, \xi \in J} j(\nu) \oplus j(\xi)$$

By continuity of $\oplus$ we have that

$$\bigsqcup^{\uparrow}_{\nu \in I, \xi \in J} j(\nu) \oplus j(\xi) = \bigsqcup^{\uparrow}_{\nu \in I} j(\nu) \oplus \bigsqcup^{\uparrow}_{\xi \in J} j(\xi) = j^\dagger(I) \oplus j^\dagger(J)$$

Similarly for the scalar multiplication and $\underline{0}$.

We observe also that the construction $\mathcal{P}_{TM}(\mathcal{V}(\mathcal{D}))$ can be obtained via abstract bases. Consider a basis B for D. We define an AB-relation on $P_{TM}(V(B))$ as follows

$$X \prec Y \text{ if } \forall \nu \in X \exists \xi \in Y.\ \nu \prec \xi$$

We can prove, similarly to the previous work, that the ideal completion of such AB-basis is the free functor **CONT** → **CJTM** and therefore it is natural isomorphic to $\mathcal{P}_{TM}(\mathcal{V}(\mathcal{D}))$. Also the function $\iota : P_{TM}(V(B)) \to \mathcal{P}_{TM}(\mathcal{V}(\mathcal{D}))$ preserves all operations. □

Tix and Mislove define analogous notion of Plotkin and Smyth convex powerdomains. We refer the interested reader to their work.

4.9 Conclusions and future work

We have presented a denotational model for probabilistic computation designed to be combined with nondeterminism. In the category of sets and functions we have the full picture: we characterise indexed valuations both as a free construction for an equational theory, and we give a more concrete representation. Finally we show the existence of a categorical distributive law between indexed valuations and the powerset. This categorical distributive law corresponds to an equational distributive law.

In the category of continuous domains the work is not completed yet. We have characterised indexed valuations as free construction for different equational theories, we have discussed the relations between different versions of indexed valuations and continuous valuations. We have shown the existence of some categorical distributive laws between indexed valuations and powerdomains, and we have presented the cases in which we could not prove such laws exist.

Future work should mainly address the problem of a concrete characterisation of indexed valuations. Besides its intrinsic interest, a concrete notion would allow us to work in a less tedious environment than that of abstract bases. To this aim, layered indexed valuations deserve to be studied further. It would also be interesting to have a definitive proof that, in the cases where we were not able to define a distributive law, such a law does not actually exist. It may also be interesting to study the other distributive combinations of the monads. We have said that there is no distributive law $PV \to VP$ either, but the use of multisets instead of sets could provide a solution. What could be the operational reading of this model?

Chapter 5

Semantics of Programs

We give an example of how to use the constructions of the previous chapters by giving a denotational semantics to a simple imperative language with probabilistic and nondeterministic primitives. We first introduce the language with only probabilistic primitives. We present an operational semantics, a denotational semantics and we show an adequacy theorem that relates them. We then extend the language with a nondeterministic choice operator. We give the extended language an operational semantics in terms of a simplified version of probabilistic automata. We present two denotational semantics: one in terms of indexed valuations and standard powerdomains, the other in term of standard valuations and the convex powerdomain. We show adequacy theorems relating the first semantics to *deterministic* schedulers, and the second semantics to *probabilistic* schedulers. Finally we discuss the computational intuition lying behind the mathematics.

5.1 A purely probabilistic language

In this section we give an operational and a denotational semantics to a small imperative language. It is the language **IMP** of [Win93] extended with a random assignment. The denotational semantics is a simplified version of the one in [Jon90] where it was used in relation to a probabilistic Hoare logic. A similar language was given a probabilistic semantics in the early work [Koz81].

The language, which we call **PL**, has the following (abstract) syntactic categories:

- locations **Loc**, ranged over by X;
- subprobability distributions over the integers, ranged over by χ;
- arithmetical expressions **Aexp**, ranged over by a;
- boolean expressions **Bexp**, ranged over by b;
- commands **Comm**, ranged over by c.

The (abstract) BNF for the last three syntactic categories are as follows:

$$a ::= n \in \mathbb{N} \mid X \mid a + a \mid a - a \mid a * a$$

$$b ::= \mathbf{true} \mid \mathbf{false} \mid a \leq a \mid \neg b \mid b \wedge b$$

$$c ::= \mathbf{skip} \mid X := a \mid X := \chi \mid c; c \mid \mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c \mid \mathbf{while}\ b\ \mathbf{do}\ c\,.$$

During the proof of the adequacy theorem, we will find it useful to have another command, which we call "tagged" or "bounded" while:

$$c ::= \ldots \mid \mathbf{while}_i\ b\ \mathbf{do}\ c \quad (i = 0, 1, 2, \ldots)\,.$$

We also need the notion of *state*. A state is any function **Loc**→**Num**. We call Σ the set of states, ranged over by σ. We call any pair $\langle c, \sigma \rangle$ a *configuration*. We denote the set of all configurations by Γ. The set Γ is ranged over by γ.

To make the notation more uniform we introduce the symbol ϵ representing the "empty command". We use it with the following meaning:

$$\langle \epsilon, \sigma \rangle \equiv \sigma\,,$$

$$\epsilon; c \equiv c; \epsilon \equiv c\,.$$

We extend consequently the notion of configuration so that a state σ is a configuration $\langle c, \sigma \rangle$ where $c = \epsilon$.

When σ is a state, by $\sigma[n/X]$ we denote a state such that:

$$\sigma[n/X](X') = \begin{cases} \sigma(X') & \text{if } X' \neq X \\ n & \text{if } X' = X \end{cases}$$

5.1.1 The operational semantics

The operational semantics for expressions is completely standard. The relations for expression evaluation have the following form:

$$\langle a, \sigma \rangle \to n \in \mathbb{N}\,;$$

$$\langle b, \sigma \rangle \to t \quad (t = \mathbf{true}, \mathbf{false})\,.$$

The intended meaning is that the expression a in state σ evaluates to the number n, and similarly for booleans. We skip their defining rules because they are exactly as in [Win93], and behave as expected.

As for commands, we give a semantics in terms of unlabelled generative transition systems, whose states are configurations. Transitions have the following form:

$$\langle c, \sigma \rangle \quad \xrightarrow{p} \quad \langle c', \sigma' \rangle$$

where $p \in [0, 1]$. The intended meaning is: in state σ the command c produces the state σ' and passes the control to the command c' (the "residual" program) with probability p. When the residual command is ϵ, the execution terminates, producing a (final) state σ'.

The rules to derive transitions are as follows:

$$\langle \mathbf{skip}, \sigma \rangle \xrightarrow{1} \sigma$$

The command **skip** has a deterministic behaviour, does not change the state and stops after the execution.

$$\frac{\langle a, \sigma\rangle \to n}{\langle X := a, \sigma\rangle \xrightarrow{1} \sigma[n/X]}$$

The usual assignment is deterministic, and stops after the execution.

$$\langle X := \chi, \sigma\rangle \xrightarrow{\chi(n)} \sigma[n/X]$$

The random assignment has a probabilistic behaviour, and stops after execution.

$$\frac{\langle c_0, \sigma\rangle \xrightarrow{p} \langle c_0', \sigma'\rangle}{\langle c_0; c_1, \sigma\rangle \xrightarrow{p} \langle c_0'; c_1, \sigma'\rangle}$$

Note that for $c_0' = \epsilon$ the rule reads:

$$\frac{\langle c_0, \sigma\rangle \xrightarrow{p} \sigma'}{\langle c_0; c_1, \sigma\rangle \xrightarrow{p} \langle c_1, \sigma'\rangle}$$

In a sequence, the first command releases the control after terminating the execution.

$$\frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_0, \sigma\rangle} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_1, \sigma\rangle}$$

$$\frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \langle c; \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \sigma}$$

The conditional and the while are deterministic.

Finally the tagged while. It behaves like the while, except that every loop decreases the tag, unless the tag is 0, where it blocks.

$$\frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{while}_i\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \sigma} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{while}_{i+1}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \langle c; \mathbf{while}_i\ b\ \mathbf{do}\ c, \sigma\rangle}$$

Note that the last rule does not apply for the case of $\mathbf{while}_0$. In this case the program does not correctly terminate, but cannot continue either. This is a *blocked* configuration. The above rules are deterministic in the following sense: if $\gamma \xrightarrow{p} \gamma'$ and $\gamma \xrightarrow{p'} \gamma'$ then $p = p'$. Moreover such a transition has a unique derivation.

Given a configuration $\langle c, \sigma\rangle$ we represent the computations it yields as paths. Every path is obtained by gluing together derivable transitions.

$$\langle c_0, \sigma_0\rangle \xrightarrow{p_0} \langle c_1, \sigma_1\rangle \xrightarrow{p_1} \langle c_2, \sigma_2\rangle \xrightarrow{p_2} \dots$$

If s is a path, with $l(s)$ we indicate the last configuration of the path. A path is *maximal* if it is either infinite, or it is not further extensible. (This is the case, for example, when the last configuration is a final state.) Finite maximal paths ending in a configuration represent blocked computations. The set of maximal paths starting from γ is called $\mathcal{B}(\gamma)$. Assuming probabilistic independence of all random assignments, for every $s \in \mathcal{B}(\gamma)$ we define the *weight* or *probability* of s, $\Pi(s)$, as the product of all the probability labels in s.

We can evaluate the probability of reaching a final state σ', by summing up the probabilities of all paths ending in σ'. We therefore give the following definition.

Definition 5.1.1. Let $\gamma := \langle c, \sigma\rangle$. Then:

$$c(\sigma, \sigma') := \sum_{\substack{s\in\mathcal{B}(\gamma)\\ l(s)=\sigma'}} \Pi(s)\,.$$

Thus $c(\sigma, \sigma')$ represents the probability that the program c in state σ yields state σ'. The following result shows the behaviour of $\sigma' \mapsto c(\sigma, \sigma')$ with respect to the sequential composition.

Theorem 5.1.2. *For every* $c_0, c_1, \sigma, \sigma'$

$$(c_0; c_1)(\sigma, \sigma') = \sum_{\sigma''\in\Sigma} c_0(\sigma, \sigma'') \cdot c_1(\sigma'', \sigma')\,.$$

Proof: For the sake of clarity let us put

$$\gamma := \langle c_0; c_1,\ \sigma\rangle,\ \gamma_0 := \langle c_0,\ \sigma\rangle,\ \gamma_1(\sigma'') := \langle c_1,\ \sigma''\rangle\,.$$

First we show that

$$(c_0; c_1)(\sigma, \sigma') \leq \sum_{\sigma''\in\Sigma} c_0(\sigma, \sigma'') \cdot c_1(\sigma'', \sigma').$$

For this to hold it is enough to show that

$$\sum_{\substack{s\in\mathcal{B}(\gamma)\\ l(s)=\sigma'}} \Pi(s) \leq \sum_{\sigma''\in\Sigma} \sum_{\substack{s'\in\mathcal{B}(\gamma_0)\\ l(s')=\sigma''}} \sum_{\substack{s''\in\mathcal{B}(\gamma_1(\sigma''))\\ l(s'')=\sigma'}} \Pi(s')\Pi(s'')\,.$$

Every path $s \in \mathcal{B}(\gamma)$ is obtained by "concatenating" a path $s' \in \mathcal{B}(\gamma_0)$ and a path $s'' \in \mathcal{B}(\gamma_1(\sigma''))$ for some $\sigma'' \in \Sigma$. To be precise we should say that it is a concatenation together with a change of labels in s': every label $\langle c, \tilde{\sigma}\rangle$ in s' becomes $\langle c; c_1, \tilde{\sigma}\rangle$. Note that $\langle c, \sigma\rangle \xrightarrow{p} \langle c', \sigma'\rangle$ is derivable if and only if $\langle c; c_1, \sigma\rangle \xrightarrow{p} \langle c'; c_1, \sigma'\rangle$.

In this case obviously $\Pi(s) = \Pi(s')\Pi(s'')$. Thus for every member of the left-hand side summation there is a corresponding identical member in the right-hand side summation.

To prove the converse inequation we have to argue that a path $s \in \mathcal{B}(\gamma)$ is generated in a unique way as a concatenation of paths s', s'' as above. Suppose there are two maximal paths t', t'' whose concatenation gives us s. If the length of t' is the same as the length of s', then clearly $t' = s'$ (and $t'' = s''$). If the length of t' is, say, smaller than the length of s' then t' is a prefix of s'. But no maximal path can be prefix of another path. □

The following result shows that $\sigma' \mapsto c(\sigma, \sigma')$ has the actual property of a probability distribution.

Proposition 5.1.3. *For every c, σ*

$$\sum_{\sigma' \in \Sigma} c(\sigma, \sigma') \in [0,1] \,.$$

This is a consequence of the adequacy theorem (Theorem 5.1.4).

5.1.2 The denotational semantics

Given an arithmetic expression a, its denotation $[\![a]\!]$ is a function $\Sigma \to \mathbb{N}$. Similarly the denotation of a boolean expression b is a function $[\![b]\!] : \Sigma \to \{\mathbf{true}, \mathbf{false}\}$. It is defined by

$$[\![a]\!]\sigma = n \iff \langle a, \sigma\rangle \to n \,,$$

$$[\![b]\!]\sigma = t \iff \langle b, \sigma\rangle \to t \,.$$

(Alternatively they could be defined by structural induction and the above definitions would be an easy theorem)

We recall here that $V_\infty^{\leq 1}$ defines a monad in **SET**. Its Kleisli extension is defined as follows. Given $f : X \to V_\infty^{\leq 1}(Y)$, the function $f^\dagger : V_\infty^{\leq 1}(X) \to V_\infty^{\leq 1}(Y)$ is defined as

$$f^\dagger(\nu)(y) := \sum_{x \in X} \nu(x) \cdot f(x)(y) \,.$$

We are now ready to define the denotation of commands in **PL**. For every $c \in \mathbf{Comm}$ we define the denotation $[\![c]\!] : \Sigma \to V_\infty^{\leq 1}(\Sigma)$ as follows.

$$[\![\mathbf{skip}]\!]\sigma = \eta_\sigma$$

$$[\![X := a]\!]\sigma = \eta_{\sigma[n/X]} \text{ where } n = [\![a]\!]\sigma$$

$$[\![X := \chi]\!]\sigma = \xi \text{ where } \xi(\sigma') = \begin{cases} p & \text{if } \sigma' = \sigma[n/X] \;\&\; \chi(n) = p \\ 0 & \text{otherwise} \end{cases}$$

$$[\![c_0; c_1]\!] = [\![c_1]\!]^\dagger \circ [\![c_0]\!]$$

$$[\![\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1]\!]\sigma = \begin{cases} [\![c_0]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \\ [\![c_1]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{false} \end{cases}$$

$$[\![\mathbf{while}_0\ b\ \mathbf{do}\ c]\!]\sigma = \begin{cases} \eta_\sigma & \text{if } [\![b]\!]\sigma = \mathbf{false} \\ \lambda\sigma' \in \Sigma.0 & \text{if } [\![b]\!]\sigma = \mathbf{true} \end{cases}$$

$$[\![\mathbf{while}_{i+1}\ b\ \mathbf{do}\ c]\!]\sigma = \begin{cases} \eta_\sigma & \text{if } [\![b]\!]\sigma = \mathbf{false} \\ [\![c; \mathbf{while}_i\ b\ \mathbf{do}\ c]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \end{cases}$$

We can prove that $[\![c]\!]$ is well defined by well founded induction. The well order is defined as follows. Let $maxt(c)$ be the maximum tag in a while command occurring in c (0 if there are no while commands). We say that $c_0 \preceq c_1$ if (1) $maxt(c_0) < maxt(c_1)$ or if (2) $maxt(c_0) = maxt(c_1)$ and c_0 is a subterm of c_1.

To define the denotation of the un-tagged while we observe that for any set X, the set $V_\infty^{\leq 1}(X)$ with the pointwise order is a DCPO with bottom element the distribution constant on 0. It is easy to check that the combinators used in the definition of the semantics correspond to continuous functions for this DCPO. The DCPO structure carries over to the set of functions $f : X \to V_\infty^{\leq 1}(X)$. We denote this set by $[X \to V_\infty^{\leq 1}(X)]$, with the order defined as $f \sqsubseteq g$ if for every $x \in X$, $f(x) \sqsubseteq g(x)$. The bottom element is the function that sends every element to the bottom of $V_\infty^{\leq 1}(X)$.

We have seen (Theorem 2.3.2) that a continuous function $[X \to V_\infty^{\leq 1}(X)] \to [X \to V_\infty^{\leq 1}(X)]$ has a (least) fixed point. Let us define a continuous function $\Phi : [\Sigma \to V_\infty^{\leq 1}(\Sigma)] \to [\Sigma \to V_\infty^{\leq 1}(\Sigma)]$ as follows:

$$\Phi(f)(\sigma) = \begin{cases} \eta_\sigma & \text{if } [\![b]\!]\sigma = \mathbf{false}\,, \\ (f^\dagger \circ [\![c]\!])\sigma & \text{if } [\![b]\!]\sigma = \mathbf{true}\,. \end{cases}$$

This function is continuous because all the operations involved in its definition are. In particular the function $(-)^\dagger$ is continuous as a function $[\Sigma \to V_\infty^{\leq 1}(\Sigma)] \to [V_\infty^{\leq 1}(\Sigma) \to V_\infty^{\leq 1}(\Sigma)]$.

We use such fixed point to define the denotation of the while command, by putting:

$$[\![\mathbf{while}\ b\ \mathbf{do}\ c]\!] = \mathbf{Fix}(\Phi)\,.$$

5.1.3 The equivalence of the semantics

Theorem 5.1.4 (Adequacy). *For every command c of* **PL***, and for every pair of states $\sigma, \sigma' \in \Sigma$,*

$$c(\sigma, \sigma') = [\![c]\!]\sigma\sigma'\,.$$

In order to prove the result for the full language, we first prove adequacy for the language $\mathbf{PL}^-$ which is the original language without the un-tagged while.

Definition 5.1.5. The language $\mathbf{PL}^-$ has the same syntax as **PL** except that it does not include the constructor **while** b **do** c. the operational and denotational semantics of $\mathbf{PL}^-$ are defined as for **PL**, removing all the rules which refer to **while** b **do** c.

Lemma 5.1.6. *If c is a command of $\mathbf{PL}^-$ then for every $\sigma, \sigma' \in \Sigma$,*

$$c(\sigma, \sigma') = [\![c]\!]\sigma\sigma'\,.$$

Proof: By well founded induction using the same well order as defined in the previous section.

The interesting case is the sequencing $c_0; c_1$: by induction hypothesis we have that

$$c_i(\sigma, \sigma') = [\![c_i]\!]\sigma\sigma' \ \text{ for } i \in \{0,1\}, \quad \sigma, \sigma' \in \Sigma \ \ (*)\,.$$

By Theorem 5.1.2

$$(c_0; c_1)(\sigma, \sigma') = \sum_{\sigma'' \in \Sigma} c_0(\sigma, \sigma'') \cdot c_1(\sigma'', \sigma')\,.$$

By $(*)$

$$\sum_{\sigma'' \in \Sigma} c_0(\sigma, \sigma'') \cdot c_1(\sigma'', \sigma') = \sum_{\sigma'' \in \Sigma} [\![c_0]\!]\sigma\sigma'' \cdot [\![c_1]\!]\sigma''\sigma'\,.$$

By the definition of the Kleisli extension and the definition of the denotation of the sequencing

$$\sum_{\sigma'' \in \Sigma} [\![c_0]\!]\sigma\sigma'' \cdot [\![c_1]\!]\sigma''\sigma' = [\![c_1]\!]^\dagger \circ [\![c_0]\!]\sigma\sigma' = [\![c_0; c_1]\!]\sigma\sigma' \,.$$

□

Definition 5.1.7. [1] Let c be a command of **PL**. Then $c^{(i)}$ is the command we obtain substituting in c all the occurrences of **while** b **do** c' with **while**$_i$ b **do** c'.

Clearly $c^{(i)}$ is a command of $\mathbf{PL}^-$, for every i. Therefore, by Lemma 5.1.6, we have:

Observation 5.1.8. *For every c, σ, σ'*

$$c^{(i)}(\sigma, \sigma') = [\![c^{(i)}]\!]\sigma\sigma' \,.$$

Next we observe that:

Proposition 5.1.9. *For every c, σ, σ'*

$$\sup_{i \in \mathbb{N}} [\![c^{(i)}]\!]\sigma\sigma' = [\![c]\!]\sigma\sigma' \,.$$

Proof: By structural induction. It works because all combinators used in defining the denotational semantics are continuous (continuity in the sense of real numbers implies Scott-continuity). The only non-trivial case is the untagged while.

Define

$$e := \sup_{i \in \mathbb{N}} [\![\mathbf{while}_i \ b \ \mathbf{do} \ c^{(i)}]\!] \,.$$

By continuity we have that

$$e = \sup_{i \in \mathbb{N}} \sup_{j \in \mathbb{N}} [\![\mathbf{while}_i \ b \ \mathbf{do} \ c^{(j)}]\!] \,.$$

By induction hypothesis and by continuity,

$$e = \sup_{i \in \mathbb{N}} [\![\mathbf{while}_i \ b \ \mathbf{do} \ c]\!] \,.$$

Now we recall the definition of the operator Φ:

$$\Phi(f)(\sigma) = \begin{cases} \eta_\sigma & \text{if } [\![b]\!]\sigma = \mathbf{false}\,, \\ (f^\dagger \circ [\![c]\!])\sigma & \text{if } [\![b]\!]\sigma = \mathbf{true}\,. \end{cases}$$

By induction on i one can prove that:

$$[\![\mathbf{while}_i \ b \ \mathbf{do} \ c]\!] = \Phi^{i+1}(\lambda\sigma \in \Sigma.0)$$

which gives us the final result:

$$e = \mathbf{Fix}(\Phi) = [\![\mathbf{while} \ b \ \mathbf{do} \ c]\!] \,.$$

□

Furthermore we have:

[1] This informal definition can be turned into a definition by structural induction.

Theorem 5.1.10. *For every c, σ, σ'*

$$\sup_{i \in \mathbb{N}} c^{(i)}(\sigma, \sigma') = c(\sigma, \sigma') .$$

Proof: Let us give the following definition:

$$c_{[k]}(\sigma, \sigma') := \sum_{\substack{s \in \mathcal{B}(\langle c, \sigma \rangle),\ l(s) = \sigma' \\ length(s) \leq k}} \Pi(s) .$$

It is clear that

$$\sup_{k \in \mathbb{N}} c_{[k]}(\sigma, \sigma') = c(\sigma, \sigma') .$$

Therefore we have just to show that the two increasing sequences $c^{(i)}(\sigma, \sigma')$ and $c_{[k]}(\sigma, \sigma')$ are cofinal. Since the max length of a path in $\mathcal{B}(\langle c^{(i)}, \sigma \rangle)$ is finite, there exists k such that $c^{(i)}(\sigma, \sigma') \leq c_{[k]}(\sigma, \sigma')$. For the other direction take any $k \in \mathbb{N}$. We have to find i such that $c_{[k]}(\sigma, \sigma') \leq c^{(i)}(\sigma, \sigma')$. In a maximal path of length k there cannot be more that k different occurrences of a while command. Since during every step the tag decreases at most of 1, we can simulate the same computation using while commands with tag k. This means that putting $i = k$ we get just $c_{[k]}(\sigma, \sigma') \leq c^{(i)}(\sigma, \sigma')$.

Combining all the previous results we get

$$c(\sigma, \sigma') = [\![c]\!] \sigma \sigma' .$$

□

5.2 Nondeterministic and probabilistic choice

We now extend the language with a nondeterministic choice operator. We will give an operational semantics in terms of probabilistic automata. We will give a denotational semantics in terms of indexed valuation. We show an adequacy theorem relating the two semantics.

5.2.1 Probabilistic automata

Probabilistic automata were introduced in Section 2.6. We are going to adapt that general framework to our needs. We recall that if Y is a subset of $V^1_\infty(X)$, by $\overline{Y}$ we denote the set of convex combinations of elements of Y.

In general a probabilistic automaton is a coalgebra for the functor $P_\perp(A \times V^1_\infty(-)) : \mathbf{SET} \to \mathbf{SET}$. In this chapter will not make use of labels, therefore a probabilistic automaton on a set of states X will be a function $k : X \to P_\perp(V^1_\infty(X))$ together with an initial state $x_0 \in X$. We will use the notation of [HP00]. Whenever $\nu \in k(x)$ we will write

$$x(\xrightarrow{p_i} x_i)_{i \in I}$$

where $x_i \in X$, $i \neq j \Longrightarrow x_i \neq x_j$, and $\nu(x_i) = p_i$. A *finite path* of a probabilistic automaton is an element in $(X \times V^1_\infty(X))^* X$, written as $x_0 \nu_0 \ldots x_{n-1} \nu_{n-1} x_n$, such that $\nu_i(x_{i+1}) > 0$. The path is *deterministic* if $\nu_i \in k(x_i)$. It is *probabilistic* if $\nu_i \in \overline{k(x_i)}$. The last state of a path s is denoted by $l(s)$. The probability of a path $s := x_0\nu_0 \dots x_{n-1}\nu_{n-1}x_n$ is defined as

$$\Pi(s) = \prod_{i<n} \nu_i(x_{i+1})\,.$$

We do not want to allow schedulers to block a computation, therefore we use a slightly different definition, than the one of section 2.6.

A *probabilistic scheduler* for a probabilistic automaton k is a partial function $\mathcal{S} : (X \times V^1_\infty(X))^* X \to V^1_\infty(X)$ such that

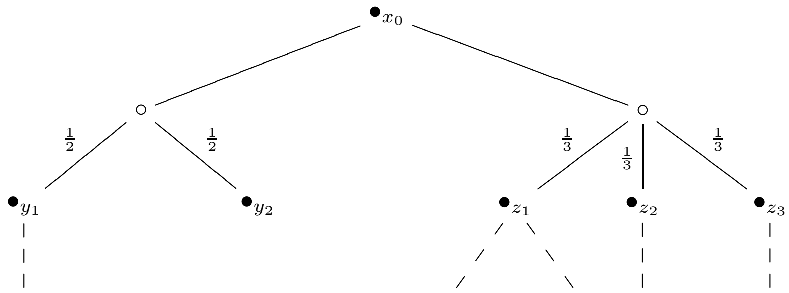
- if $k(l(r)) \neq \emptyset$ then $\mathcal{S}(r)$ is defined;
- $\mathcal{S}(r) \in \overline{k(l(r))}$;

Equivalently we could define a probabilistic scheduler to be a partial function $\mathcal{S} : (X \times V^1_\infty(X))^* X \to V^1(V^1_\infty(X))$, requiring that $Supp(\mathcal{S}(r)) \subseteq k(l(r))$.

A *deterministic* scheduler is a probabilistic scheduler that does not make use of the convex combinations. That is for a deterministic scheduler we have $\mathcal{S}(r) \in k(l(r))$.

Now given a state $x \in X$ and a scheduler $\mathcal{S}$ for k, we consider the set $\mathcal{B}(k, \mathcal{S})$ of maximal paths, obtained from k by the action of $\mathcal{S}$. That is the paths $x_0\nu_0 \dots x_{n-1}\nu_{n-1}x_n$ such that $\nu_i = \mathcal{S}(x_0\nu_0 \dots x_i)$. A deterministic scheduler generates deterministic paths, a probabilistic scheduler generates probabilistic paths.

A good way of visualising probabilistic automata is by using alternating trees [Han91]. Black nodes represent states, hollow nodes represent probability distributions. The use of trees instead of graphs is a way of keeping track of the paths: a deterministic scheduler is thus a function that, for every black node, chooses one of its hollow sons.



5.2.2 The operational semantics

The language **NPL** has the same syntax as **PL** with one more constructor.

$$c ::= \dots \mid c \ \mathbf{or} \ c\,.$$

The operational semantics is given in terms of unlabelled probabilistic automata on the set of configurations (and final states). For every configuration γ_0 we have the probabilistic automaton $\mathcal{M}(\gamma_0) = (\Gamma, k, \gamma_0)$ where the steps in k are derived inductively using the following rules.

We write $\xrightarrow{p}\sigma$ for $(\xrightarrow{p}\sigma)_{i\in\{*\}}$.

$$\langle \mathbf{skip}, \sigma\rangle \xrightarrow{1} \sigma$$

$$\frac{\langle a, \sigma\rangle \to n}{\langle X := a, \sigma\rangle \xrightarrow{1} \sigma[n/X]}$$

$$\langle X := \chi, \sigma\rangle (\xrightarrow{\chi(n)} \sigma[n/X]\ )_{n\in\mathbf{Num}}$$

$$\frac{\langle c, \sigma\rangle (\xrightarrow{p_i} \langle c_i, \sigma_i\rangle)_{i\in I}}{\langle c; c', \sigma\rangle (\xrightarrow{p_i} \langle c_i; c', \sigma_i\rangle)_{i\in I}}$$

where some c_i could be ϵ.

$$\frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_1, \sigma\rangle} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma\rangle \xrightarrow{1} \langle c_0, \sigma\rangle}$$

$$\frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{while}_i\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \sigma} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{while}_{i+1}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \langle c; \mathbf{while}_i\ b\ \mathbf{do}\ c, \sigma\rangle}$$

$$\frac{\langle b, \sigma\rangle \to \mathbf{false}}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \sigma} \qquad \frac{\langle b, \sigma\rangle \to \mathbf{true}}{\langle \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle \xrightarrow{1} \langle c; \mathbf{while}\ b\ \mathbf{do}\ c, \sigma\rangle}$$

$$\frac{\langle c, \sigma\rangle (\xrightarrow{p_i} \gamma_i)_{i\in I}}{\langle c\ \mathbf{or}\ c', \sigma\rangle (\xrightarrow{p_i} \gamma_i)_{i\in I}} \qquad \frac{\langle c', \sigma\rangle (\xrightarrow{p_j} \gamma_j)_{j\in J}}{\langle c\ \mathbf{or}\ c', \sigma\rangle (\xrightarrow{p_j} \gamma_j)_{j\in J}}$$

Definition 5.2.1. Le $\mathcal{S}$ be a scheduler for $\mathcal{M}(\langle c, \sigma\rangle)$. We define $\mathcal{B}(\langle c, \sigma\rangle, \mathcal{S})$ to be the set of finite paths $s \in \mathcal{B}(\mathcal{M}(\langle c, \sigma\rangle))$ such that $l(s)$ is a state. We define $Val(\mathcal{S}, c, \sigma)$ to be the probability distribution such that

$$Val(\mathcal{S}, c, \sigma)(\sigma') = \sum_{\substack{s\in\mathcal{B}(\langle c,\sigma\rangle,\mathcal{S}) \\ l(s)=\sigma'}} \Pi(s)\,.$$

We define $Ival(\mathcal{S}, c, \sigma)$ to be the discrete indexed valuation

$$(l(s), \Pi(s))_{s\in\mathcal{B}(\langle c,\sigma\rangle,\mathcal{S})}\,.$$

Note that $Val(\mathcal{S}, c, \sigma) = Flat\big(Ival(\mathcal{S}, c, \sigma)\big)$. Paths in $\mathcal{B}(\langle c, \sigma\rangle, \mathcal{S})$, are paths that end properly. We assume we cannot observe blocked computations.

5.2.3 Adequacy for the finite fragment

In the sequel we shall write $(x_i, p_i)_{i\in I}$ also to denote $\iota\big((x_i, p_i)_{i\in I}\big) \in \mathcal{IV}(X)$. We will use this notation even when I is not finite, to denote the lub of all its finite "truncations".

Definition 5.2.2. The language $\mathbf{NPL}^-$ has the same syntax as **NPL** except that it does not include the constructor **while** b **do** c and all valuations χ involved have finite support. Operational semantics of $\mathbf{NPL}^-$ is defined like for **NPL**.

The denotational semantics

$$\mathcal{F}[\![c]\!] : \Sigma \to P(IV(\Sigma))$$

is defined as follows. The indexed valuation $(\sigma, p)_{*\in\{*\}}$ will be denoted as (σ, p).

$$\mathcal{F}[\![\mathbf{skip}]\!]\sigma = \{(\sigma, 1)\}$$

$$\mathcal{F}[\![X := a]\!]\sigma = \{(\sigma[n/X], 1)\} \text{ where } n = [\![a]\!]\sigma$$

$$\mathcal{F}[\![X := \chi]\!]\sigma = \{(\sigma[n/X], \chi(n))_{n\in Supp(\chi)}\}$$

$$\mathcal{F}[\![c_0; c_1]\!] = \mathcal{F}[\![c_1]\!]^\dagger \circ \mathcal{F}[\![c_0]\!]$$

$$\mathcal{F}[\![c_0 \ \mathbf{or} \ c_1]\!]\sigma = \mathcal{F}[\![c_1]\!]\sigma \cup \mathcal{F}[\![c_0]\!]\sigma$$

$$\mathcal{F}[\![\mathbf{if} \ b \ \mathbf{then} \ c_0 \ \mathbf{else} \ c_1]\!]\sigma = \begin{cases} \mathcal{F}[\![c_0]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \\ \mathcal{F}[\![c_1]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{false} \end{cases}$$

$$\mathcal{F}[\![\mathbf{while}_0 \ b \ \mathbf{do} \ c]\!]\sigma = \begin{cases} \{(\sigma, 1)\} & \text{if } [\![b]\!]\sigma = \mathbf{false} \\ \{\underline{0}\} & \text{if } [\![b]\!]\sigma = \mathbf{true} \end{cases}$$

$$\mathcal{F}[\![\mathbf{while}_{i+1} \ b \ \mathbf{do} \ c]\!]\sigma = \begin{cases} \{(\sigma, 1)\} & \text{if } [\![b]\!]\sigma = \mathbf{false} \\ \mathcal{F}[\![c; \mathbf{while}_i \ b \ \mathbf{do} \ c]\!](\sigma) & \text{if } [\![b]\!]\sigma = \mathbf{true} \end{cases}$$

We now show that there is a very tight correspondence between the denotational and the operational semantics. In the sequel we write $\mathcal{S}$ *is a scheduler for* $\langle c, \sigma\rangle$ to mean that $\mathcal{S}$ is a scheduler for $\mathcal{M}(\langle c, \sigma\rangle)$.

Theorem 5.2.3 (Adequacy). *Let c be a command of* $\mathbf{NPL}^-$ *and ν be a finite indexed valuation $\in IV(\Sigma)$. Then $\nu \in \mathcal{F}[\![c]\!]\sigma$ if and only if there exists a scheduler $\mathcal{S}$ for $\mathcal{M}(\langle c, \sigma\rangle)$ s.t. $\nu = Ival(\mathcal{S}, c, \sigma)$.*

Proof: By well founded induction, the ordering being essentially the one defined in section 5.1.2 (lexicographic on tags×structure). The nontrivial case is the sequential composition. A path in $\mathcal{B}(\mathcal{M}(\langle c_0; c_1, \sigma\rangle))$ is the concatenation of a path r in $\mathcal{B}(\mathcal{M}(\langle c_0, \sigma\rangle))$ together with a path t in $\mathcal{B}(\mathcal{M}(\langle c_1, l(t)\rangle))$, renaming the configurations of the first part. Therefore a scheduler $\mathcal{S}$ for $\langle c_0; c_1, \sigma\rangle$ can be thought of as a scheduler $\mathcal{S}_0$ for $\langle c_0, \sigma\rangle$ together with schedulers $\mathcal{S}_r$ for $\langle c_1, l(r)\rangle$ for every finite $r \in \mathcal{B}(\mathcal{M}(\langle c_0, \sigma\rangle))$. (In the sequel we write $\mathcal{B}(c_0, \sigma, \mathcal{S})$ for $\mathcal{B}(\langle c_0, \sigma\rangle, \mathcal{S})$).

By the induction hypothesis $(l(r), \Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)} \in \mathcal{F}[\![c_0]\!]\sigma$ and for every r, $(l(t), \Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)} \in \mathcal{F}[\![c_1]\!]l(r)$. We have to show that

$$(l(s), \Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})} \in \mathcal{F}[\![c_1]\!]^\dagger(\mathcal{F}[\![c_0]\!]\sigma)\,.$$

Recalling the definition of $f^\dagger$, it is enough to show that

$$(l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}\in\mathcal{F}[\![c_1]\!]^\dagger\left(\{\,(l(r),\Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}\,\}\right)\,.$$

Let us define $h:\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\to IV(\Sigma)$ as

$$h(r)=(l(t),\Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}\in\mathcal{F}[\![c_1]\!]l(r)\,.$$

Therefore by definition of $f^\dagger$:

$$(l(t),\Pi(r)\Pi(t))_{\substack{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}}\in\mathcal{F}[\![c_1]\!]^\dagger\left(\{\,(l(r),\Pi(r))_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}\,\}\right)\,.$$

Since a path in $\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})$ is the concatenation of a path r in $\mathcal{B}(\mathcal{M}(\langle c_0,\sigma\rangle))$ together with a path t in $\mathcal{B}(\mathcal{M}(\langle c_1,l(t)\rangle))$, we have

$$(l(t),\Pi(r)\Pi(t))_{\substack{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}}=$$

$$=(l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}\,.$$

Conversely suppose $(\sigma_i,p_i)_{i\in I}\in\mathcal{F}[\![c_1]\!]^\dagger(\mathcal{F}[\![c_0]\!]\sigma)$. By definition of the Kleisli extension, there exist $(\tau_j,q_j)_{j\in J}\in\mathcal{F}[\![c_0]\!]\sigma$ and $h:J\to IV(\Sigma)$ such that $h(j)\in\mathcal{F}[\![c_1]\!]\tau_j$ and

$$(\sigma_i,p_i)_{i\in I}=\mu^{IV}\left((h(j),q_j)_{j\in J}\right)\,.$$

By the induction hypothesis there exists a scheduler $\mathcal{S}_0$, such that

$$J=\mathcal{B}(c_0,\sigma,\mathcal{S}_0),q_r=\Pi(r),\tau_r=l(r)\,.$$

And for every $r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)$, there is a scheduler $\mathcal{S}_r$ such that

$$h(r)=(l(t),\Pi(t))_{t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}\,.$$

Combining $\mathcal{S}_0$ with the $\mathcal{S}_r$ we obtain a scheduler $\mathcal{S}$ for $\langle c_0;c_1,\sigma\rangle$. In order to obtain an overall scheduler $\mathcal{S}$, formally we have to define it also for the paths not in $\mathcal{B}(c_0,\sigma,\mathcal{S}_0)$. But this choice can be arbitrary, because it does not influence the definition $\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})$. Recalling the definition of μ^{IV} we get $(\sigma_i,p_i)_{i\in I}=(l(s),\Pi(s))_{s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}$. □

5.2.4 Adequacy for the full language

The adequacy theorem we are going to prove states that, if the denotation of a configuration contains a valuation, then there is a scheduler that (almost) realises that valuation, or perhaps does better.

Theorem 5.2.4. *Let c be a command of* **NPL** *and let $\nu\in\mathcal{IV}(\Sigma)$. Then $\nu\in[\![c]\!]\sigma$ iff for every $\xi\ll\nu$ there exists a scheduler $\mathcal{S}$ for (c,σ) s.t. $Ival(\mathcal{S},c,\sigma)\sqsupseteq\xi$.*

We need some preliminary lemmas.

Lemma 5.2.5. *Let c be a command of* **NPL**$^-$. *Then $[\![c]\!]\sigma=\iota(\mathcal{F}[\![c]\!]\sigma)$.*

Proof: By structural induction. Notice that the definitions for $[\![c]\!]$ and $\mathcal{F}[\![c]\!]$ go in parallel with each other, and that ι preserves all the operations.

Proposition 5.2.6. *Let c be a command of* $\mathbf{NPL}^-$. *Then $\nu \in [\![c]\!]\sigma$ iff there exists $\nu' \in \mathcal{F}[\![c]\!]\sigma$ s.t. $\nu \sqsubseteq \iota(\nu')$.*

Proof: By the previous lemma and the characterisation of the Hoare powerdomain in terms of Scott closed sets. □

Definition 5.2.7. Let $\chi \in V_\infty^{\leq 1}(\mathbb{N})$. With $\chi^{(i)}$ we denote the element of $V_\infty^{\leq 1}(\mathbb{N})$ such that

$$\chi^{(i)}(n) := \begin{cases} \chi(n) & \text{if } n < i \\ 0 & \text{otherwise} \end{cases}$$

All valuations of the form $\chi^{(i)}$ have finite support.

Definition 5.2.8. Let c be a command of **NPL**. Then $c^{(i)}$ is the command we obtain by substituting in c all the occurrences of **while** b **do** c' with **while**$_i$ b **do** c' and all occurrences of χ with $\chi^{(i)}$.

Clearly $c^{(i)}$ is a command of $\mathbf{NPL}^-$, for every i. Therefore we have:

Proposition 5.2.9. *Let c be a command of* **NPL** *and ν be a finite indexed valuation $\in IV(\Sigma)$. For every i we have:*
$\iota(\nu) \in [\![c^{(i)}]\!]\sigma$ iff there exists a scheduler $\mathcal{S}$ for (c,σ) s.t. $Ival(\mathcal{S},c,\sigma) \sqsupseteq \nu$.

Proof: Clearly a scheduler $\mathcal{S}$ for (c,σ) restricts to a scheduler $\mathcal{S}_i$ for $(c^{(i)},\sigma)$. Moreover $Ival(\mathcal{S},c,\sigma) \sqsupseteq Ival(\mathcal{S}_i,c^{(i)},\sigma)$. Conversely a scheduler $\mathcal{S}_i$ for $(c^{(i)},\sigma)$ can be extended (possibly in many different ways) to a scheduler $\mathcal{S}$ for (c,σ), with the same inequality as above. This together with Proposition 5.2.6, gives us the statement. □

Proposition 5.2.10. *For every c,σ*

$$\sup_{i\in\mathbb{N}} [\![c^{(i)}]\!]\sigma = [\![c]\!]\sigma\,.$$

Proof: By structural induction, using the continuity of the operators defining the semantics. □

Coming to the proof of Theorem 5.2.4, we show the "only if" direction. For $\nu \in [\![c]\!]\sigma$ there are two cases.

1. There is $\nu' \in \mathcal{F}[\![c^{(i)}]\!]\sigma$ for some i, such that $\nu \sqsubseteq \iota(\nu')$. Then we invoke Proposition 5.2.9, and we are done.

2. There is a sequence $\nu_i \in [\![c^{(i)}]\!]\sigma$ converging to ν. By Proposition 5.2.6 it is no restriction to assume that $\nu_i = \iota(\tilde{\nu}_i)$ for some $\tilde{\nu}_i \in \mathcal{F}[\![c^{(i)}]\!]\sigma$. Then we have a sequence of schedulers $\mathcal{S}_i$ such that $\tilde{\nu}_i = (l(s_i),\Pi(s_i))_{s_i\in\mathcal{B}(\langle c^{(i)},\sigma\rangle,\mathcal{S}_i)}$. So $\tilde{\nu}_i = Ival(\mathcal{S}_i,c^{(i)},\sigma)$.

   Therefore $Ival(\mathcal{S}_i,c^{(i)},\sigma)$ converges to ν. And since $\xi \ll \nu$ there is a k such that $(Ival(\mathcal{S}_k,c^{(k)},\sigma)) \sqsupseteq \xi$. Now we can extend $\mathcal{S}_k$ to an $\mathcal{S}$ for (c,σ) with $(Ival(\mathcal{S},c,\sigma)) \sqsupseteq \xi$.

The "if" direction is similar. □

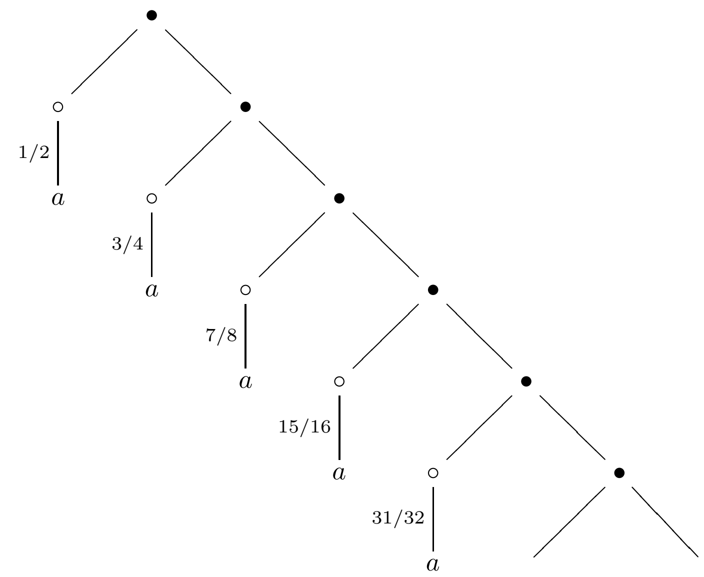
We cannot hope that there always exists a scheduler which attains the limit valuation as the following example shows.

Let χ be the probability distribution s.t. $\chi(n) = 1/2^{n+1}$. Assume also $a \neq 0$. Define

$$\mathbf{loop} \equiv \mathbf{while\ true\ do\ skip}\,;$$

$$c \equiv Y := 0; Z := 0; \mathbf{while}\ Z = 0\ \mathbf{do}\ c_0\,;$$

$$c_0 \equiv (X := \chi; \mathbf{if}\ X \leq Y\ \mathbf{then}\ Z := a\ \mathbf{else\ loop})\ \mathbf{or}\ (Y := Y + 1)\,.$$



Here there is no scheduler $\mathcal{S}$ for which $Ival(\mathcal{S}, c, \sigma)$ assigns probability 1 to a state where $Z = a$, but we can get as close to this as we want.

5.3 Randomised schedulers

The main feature of the above adequacy theorem is that it uses deterministic schedulers. A semantics in terms of the convex powerset functor is adequate with respect to probabilistic schedulers.

We first have to define $\mathcal{F}[\![c]\!] : \Sigma \to P_{TM}(V(\Sigma))$.

$$\mathcal{F}[\![\mathbf{skip}]\!]\sigma = \{\eta_\sigma\}$$

$$\mathcal{F}[\![X := a]\!]\sigma = \{\eta_{\sigma[n/X]}1\} \text{ where } n = [\![a]\!]\sigma$$

$$\mathcal{F}[\![X := \chi]\!]\sigma = \lambda\sigma' \in \Sigma. \begin{cases} \chi(n) & \text{if } \sigma' = \sigma[n/X] \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{F}[\![c_0; c_1]\!] = \mathcal{F}[\![c_1]\!]^\dagger \circ \mathcal{F}[\![c_0]\!]$$

$$\mathcal{F}[\![c_0\ \mathbf{or}\ c_1]\!]\sigma = \mathcal{F}[\![c_1]\!]\sigma \uplus \mathcal{F}[\![c_0]\!]\sigma$$

$$\mathcal{F}[\![\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1]\!]\sigma = \begin{cases} \mathcal{F}[\![c_0]\!](\sigma) & \text{if } [\![b]\!]\sigma = \textbf{true} \\ \mathcal{F}[\![c_1]\!](\sigma) & \text{if } [\![b]\!]\sigma = \textbf{false} \end{cases}$$

$$\mathcal{F}[\![\textbf{while}_0\ b \textbf{ do } c]\!]\sigma = \begin{cases} \{\eta_\sigma\} & \text{if } [\![b]\!]\sigma = \textbf{false} \\ \{\underline{0}\} & \text{if } [\![b]\!]\sigma = \textbf{true} \end{cases}$$

$$\mathcal{F}[\![\textbf{while}_{i+1}\ b \textbf{ do } c]\!]\sigma = \begin{cases} \{(\sigma, 1)\} & \text{if } [\![b]\!]\sigma = \textbf{false} \\ \mathcal{F}[\![c; \textbf{while}_i\ b \textbf{ do } c]\!](\sigma) & \text{if } [\![b]\!]\sigma = \textbf{true} \end{cases}$$

Theorem 5.3.1 (Adequacy). *Let c be a command of $\mathbf{NPL}^-$ and ν be a discrete valuation in $V(\Sigma)$. Then $\nu \in \mathcal{F}[\![c]\!]\sigma$ if and only if there exists a probabilistic scheduler $\mathcal{S}$ for $\mathcal{M}(\langle c, \sigma\rangle)$ s.t. $\nu = Val(\mathcal{S}, c, \sigma)$*

Proof: By well founded induction. Note that the probabilistic schedulers are necessary for the semantics of the nondeterministic choice, because the operator $\uplus$ is defined as union followed by convex closure.

Again the nontrivial case is sequential composition. Take a scheduler $\mathcal{S}$ for $\langle c_0; c_1, \sigma\rangle$. Such an $\mathcal{S}$ can be thought of as a scheduler $\mathcal{S}_0$ for $\langle c_0, \sigma\rangle$ together with schedulers $\mathcal{S}_r$ for $\langle c_1, l(r)\rangle$ for every finite $r \in \mathcal{B}(\langle c_0, \sigma\rangle, \mathcal{S}_0)$.

By the induction hypothesis we have that $Val(\mathcal{S}_0, c_0, \sigma) \in [\![c_0]\!]\sigma$ and for every r, $Val(\mathcal{S}_r, c_1, l(r)) \in [\![c_1]\!]l(r)$.

We have to show that

$$\lambda\sigma'. \sum_{\substack{l(s)=\sigma' \\ s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s) \in [\![c_1]\!]^\dagger([\![c_0]\!]\sigma)\,.$$

Recall the statement of Proposition 3.4.9 characterising the Kleisli extension: if $f : X \to P_{TM}(V(Y))$, then

$$f^\dagger(A) = \left\{ \sum_{x\in X} \xi(x)h(x) \mid h : X \to V(Y), h(x) \in f(x), \xi \in A \right\}$$

To prove our claim it is then enough to show that

$$\lambda\sigma'. \sum_{\substack{l(s)=\sigma' \\ s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s) \in [\![c_1]\!]^\dagger\left(\{\, Val(\mathcal{S}_0, c_0, \sigma)\,\}\right)\,.$$

Let us define $h : \Sigma \to V(\Sigma)$ as

$$h(\sigma'') = \sum_{\substack{l(r)=\sigma'' \\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{Val(\mathcal{S}_0, c_0, \sigma)(\sigma'')} Val(\mathcal{S}_r, c_1, \sigma'')\,.$$

Remember that, by definition:

$$\sum_{\substack{l(r)=\sigma'' \\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r) = Val(\mathcal{S}_0, c_0, \sigma)(\sigma'')\,.$$

Therefore

$$\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')} = 1$$

Since $[\![c_1]\!]\sigma''$ is convex, then $h(\sigma'') \in ([\![c_1]\!]\sigma'')$. Therefore by Proposition 3.4.9:

$$\sum_{\sigma''\in\Sigma} Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')h(\sigma'') \in [\![c_1]\!]^\dagger\left(\{\,Val(\mathcal{S}_0,c_0,\sigma)\,\}\right) .$$

But

$$\sum_{\sigma''\in\Sigma} Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')h(\sigma'')(\sigma')$$

$$= \sum_{\sigma''\in\Sigma}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r)\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')} Val(\mathcal{S}_r,c_1,\sigma'')(\sigma')\right)\right)$$

$$= \sum_{\sigma''\in\Sigma}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r)Val(\mathcal{S}_r,c_1,\sigma'')(\sigma')\right)\right)$$

$$= \sum_{\sigma''\in\Sigma}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \frac{\Pi(r)}{Val(\mathcal{S}_0,c_0,\sigma)(\sigma'')}\right)\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r)Val(\mathcal{S}_r,c_1,\sigma'')(\sigma')\right)$$

$$= \sum_{\sigma''\in\Sigma}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r)Val(\mathcal{S}_r,c_1,\sigma'')(\sigma')\right)$$

$$= \sum_{\sigma''\in\Sigma}\left(\sum_{\substack{l(r)=\sigma''\\ r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)}} \Pi(r)\left(\sum_{\substack{l(t)=\sigma'\\ t\in\mathcal{B}(c_1,\sigma'',\mathcal{S}_r)}} \Pi(t)\right)\right)$$

$$= \sum_{r\in\mathcal{B}(c_0,\sigma,\mathcal{S}_0)} \Pi(r)\left(\sum_{\substack{l(t)=\sigma'\\ t\in\mathcal{B}(c_1,l(r),\mathcal{S}_r)}} \Pi(t)\right)$$

$$= \sum_{\substack{l(s)=\sigma'\\ s\in\mathcal{B}(c_0;c_1,\sigma,\mathcal{S})}} \Pi(s)\,,$$

and the claim is proved. For the last step, note that a path $s \in \mathcal{B}(c_0;c_1,\sigma,\mathcal{S})$ is the concatenation of a path $r \in \mathcal{B}(c_0,\sigma,\mathcal{S}_0)$ together with a path $t \in \mathcal{B}(c_1,l(r),\mathcal{S}_r)$.

Vice versa suppose that $\nu \in [\![c_1]\!]^\dagger([\![c_0]\!]\sigma)$. Then there exist $\xi \in [\![c_0]\!]\sigma$ and $h : \Sigma \to V(\Sigma)$ such that $h(\sigma'') \in [\![c_1]\!]\sigma''$ and $\nu = \sum_{\sigma''} \xi(\sigma'')h(\sigma'')$. By the induction hypothesis there exist schedulers $\mathcal{S}_0$, $\mathcal{S}_{\sigma''}$ such that $\xi = Val(\mathcal{S}_0, c_0, \sigma)$, and $h(\sigma'') = Val(\mathcal{S}_{\sigma''}, c_1, \sigma'')$. Similar to what we did with deterministic schedulers, we combine them to get a scheduler $\mathcal{S}$ such that $\nu = Val(\mathcal{S}, c_0; c_1, \sigma)$. Notice that in this case the combined scheduler has some memoryless character: it behaves the same for every subautomata starting at a configuration $\langle c_1, \sigma''\rangle$, regardless of the previous history.

□

We can interpret the semantics for the full language as being of the form

$$[\![c]\!] : \Sigma \to \mathcal{P}_{TM}(\mathcal{V}(\Sigma))$$

Using the characterisation of the convex Hoare powercone as ideal extension of the convex powerset we can show an adequacy result similar to the previous one, which makes use of probabilistic schedulers.

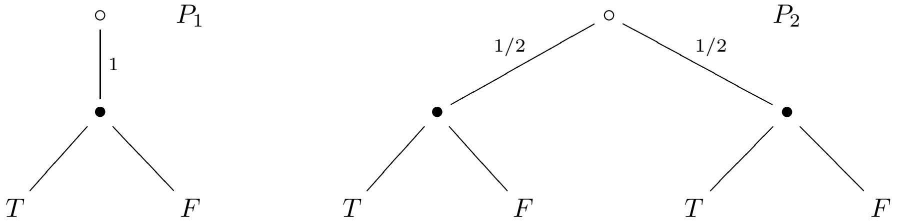
Theorem 5.3.2. *Let c be a command of* **L** *and let $\zeta \in \mathcal{V}(\Sigma)$. Then $\zeta \in [\![c]\!]\sigma$ iff for every $\epsilon > 0$ there exists a probabilistic scheduler $\mathcal{S}$ for (c, σ) s.t. $Val(\mathcal{S}, c, \sigma) \sqsupseteq (1 - \epsilon)\zeta$.*

5.4 Discussion

Using probabilistic automata and schedulers we can give another motivation for our axiom (HV) and corresponding definition 4.1.1, which implies

$$\eta_x \sqsubseteq \tfrac{1}{2}\eta_x + \tfrac{1}{2}\eta_x \ .$$

Consider the following example.



The figure represents two probabilistic processes. The process P_1 allows two different ways of resolving the nondeterminism. The corresponding probability valuations are η_T and η_F. The process P_2 allows four different ways of resolving the nondeterminism, two of which give the probability valuations $\frac{1}{2}\eta_T + \frac{1}{2}\eta_F$. The process P_2 offers more opportunities, so in a Hoare fashion, we consider it better than P_1. Formally this is implied by $1\eta_x \sqsubseteq \frac{1}{2}\eta_x + \frac{1}{2}\eta_x$.

This assumes that

- probabilistic choice is visible: in our language every probabilistic choice entails the assignment of some variable. Even if we then decide to ignore that variable, an omniscient scheduler sees the difference;
- the schedulers are deterministic. A probabilistic scheduler for P_1 can simulate a deterministic scheduler for P_2.

The example above shows why Hoare indexed valuations combine well with the Hoare powerdomain. More indices give more power to a deterministic scheduler, therefore more indices provide more information when the scheduler is under our control.

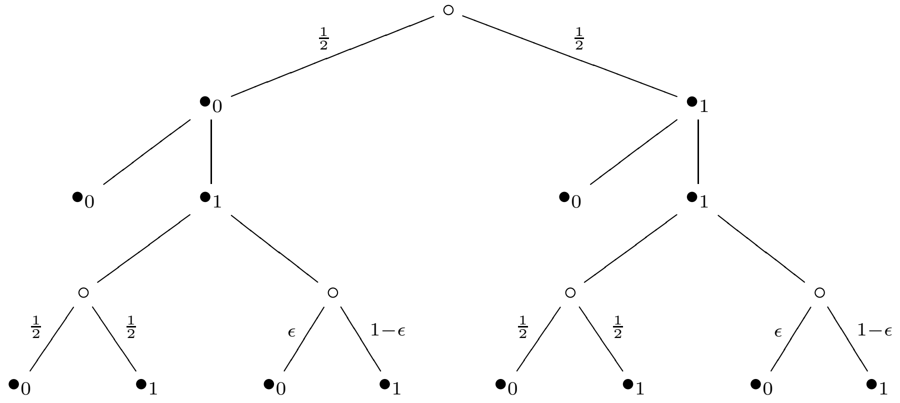
We have seen the mathematical reasons why there is no distributive law between the functors P and V. We can exemplify this with a program in our language. Suppose the denotation of a command c is to be defined as a function $[\![c]\!] : \Sigma \to P(V(\Sigma))$. If we want it to be compositional, we have to define $[\![c_1; c_2]\!]$ in terms of $[\![c_1]\!], [\![c_2]\!]$. The first intuitive idea would be to define it as

$$[\![c_1; c_2]\!](\sigma) = \big\{\lambda\sigma'. \sum_{\sigma''\in\Sigma} h(\sigma'')[\![c_2]\!](\sigma'')(\sigma') \;| h : \Sigma \to P(\Sigma), h(\sigma'') \in [\![c_1]\!](\sigma'')\big\}$$

This definition makes the sequential composition non-associative. Let

- c_1 be the command $X := \chi$, where $\chi(0) = 1/2, \chi(1) = 1/2$;
- c_2 be the command $X := 0$ **or** $X := 1$;
- c_3 be the command **if** $X = 0$ **then skip else** $X := \chi_1$ **or** $X := \chi_2$, where $\chi_1(0) = 1/2, \chi_1(1) = 1/2$, $\chi_2(0) = \epsilon, \chi_2(1) = 1 - \epsilon$;

and consider the program $c_1; c_2; c_3$.



In this example we can assume that there are only two states: $\Sigma = \{0, 1\}$ We have

- $[\![c_1]\!](i) = \{\frac{1}{2}\eta_0 + \frac{1}{2}\eta_1\}$ for $i = 0, 1$;
- $[\![c_2]\!](i) = \{\eta_0, \eta_1\}$ for $i = 0, 1$;
- $[\![c_1; c_2]\!](n) = \{\eta_0, \frac{1}{2}\eta_0 + \frac{1}{2}\eta_1, \eta_1\}$ for $i = 0, 1 \in \mathbb{N}$;
- $[\![c_3]\!](0) = \{\eta_0\}$, $[\![c_3]\!](1) = \{\frac{1}{2}\eta_0 + \frac{1}{2}\eta_1, \epsilon\eta_0 + (1 - \epsilon)\eta_1\}$;
- $[\![c_2; c_3]\!](i) = \{\eta_0, \frac{1}{2}\eta_0 + \frac{1}{2}\eta_1, \epsilon\eta_0 + (1 - \epsilon)\eta_1\}$ for $i = 0, 1$.

If we read $c_1; c_2; c_3$ as $c_1; (c_2; c_3)$, then

$$\begin{aligned}&[\![c_1; c_2; c_3]\!](i)\\&\quad = \{\eta_0, \tfrac{1}{2}\eta_0 + \tfrac{1}{2}\eta_1, \epsilon\eta_0 + (1-\epsilon)\eta_1, \tfrac{3}{4}\eta_0 + \tfrac{1}{4}\eta_1, \tfrac{1+\epsilon}{2}\eta_0 + \tfrac{1-\epsilon}{2}\eta_1, \tfrac{1+\epsilon}{4}\eta_0 + \tfrac{3-\epsilon}{4}\eta_1\}\,.\end{aligned}$$

If we read $c_1; c_2; c_3$ as $(c_1; c_2); c_3$, then

$$\begin{aligned}&[\![c_1; c_2; c_3]\!](i)\\&\qquad = \{\eta_0, \tfrac{1}{2}\eta_0 + \tfrac{1}{2}\eta_1, \epsilon\eta_0 + (1-\epsilon)\eta_1, \tfrac{3}{4}\eta_0 + \tfrac{1}{4}\eta_1, \tfrac{1+\epsilon}{2}\eta_0 + \tfrac{1-\epsilon}{2}\eta_1\}\,.\end{aligned}$$

In the second case the function h (which roughly speaking does the job of the scheduler), when choosing a valuation in $[\![c_3]\!](1)$ does not “remember” that the process has reached the state 1 by two different paths. Therefore we miss one valuation in the final set. When the denotation is given in terms of indexed valuations, the function h is given enough information to remember this. Indeed in the case of indexed valuations, h chooses looking at the paths, rather than only at the state.

However, a memoryless scheduler can simulate the combination of schedulers with memory by flipping a coin. That is why the semantics in terms of probabilistic schedulers does not need to be given in terms of indexed valuations.

5.5 Conclusions and future work

In this chapter we have shown the computational intuition behind the notion of indexed valuations by giving semantics to an imperative language with nondeterministic and probabilistic primitives. We have compared a denotational semantics in terms of indexed valuations with an operational semantics in terms of deterministic schedulers. We have argued that a compositional semantics in terms of deterministic schedulers requires them to make their decisions knowing the whole history of the process. This feature is reflected in the denotation by the use of indices to represent the histories.

A semantics in terms of probabilistic scheduler does not require the use of indices, but requires all the sets involved to be geometrically convex.

5.5.1 Coalgebra and bisimulation

Which notion of bisimulation is induced if we use indexed valuation to define transition systems?

Definition 5.5.1. An *indexed* probabilistic automaton on a set of labels A is a coalgebra for the functor $X \mapsto P_\perp(IV(A \times X))$ in the category **SET**.

In order to define the notion of bisimulation, we first need the notion of *lifting* of a relation to indexed valuations.

Definition 5.5.2. Let $R \subseteq X \times Y$ be a relation. Let $\nu = (x_i, p_i)_{i\in I} \in IV(X), \xi = (y_j, q_j)_{j\in J} \in IV(Y)$. We say that $\nu R \xi$ if $I = J$ and for every $i \in I$, $p_i = q_i$ and $x_i R y_i$.

Definition 5.5.3. A bisimulation between two indexed probabilistic automata $(X, \alpha), (Y, \beta)$ is a relation $R \subseteq X \times Y$ such that whenever $(x, y) \in R$.

- for all $\nu \in \alpha(x)$ there exists $\xi \in \beta(x)$ such that $\nu R \xi$.
- and symmetrically

Two states $x \in X, y \in Y$ are bisimilar if they are related by some bisimulation.

Using standard techniques ([BSdV03, RT94]) one can show the following.

Theorem 5.5.4. *Two states of two indexed probabilistic automata* $(X, \alpha), (Y, \beta)$ *are bisimilar iff they are coalgebraic bisimilar.*

Indexed probabilistic automata look very much like standard probabilistic automata. The difference is that an IPA that flips a coin and then chooses regardless of the outcome of the coin is NOT equivalent to an IPA that does not flip the coin at all.

Bisimilarity between IPAs seems thus to be too fine an equivalence. We don't try too escape this criticism. However one one could argue that probabilistic bisimulation is too fine already for probabilistic automata (see [DGJP99]). Moreover, distinctions of the kind seen above appear in the literature [HV99].

Note also that indexed valuations were introduced in order to define a computational monad. Although most of the functors used to define transition systems coalgebraically are monads too, this feature is not used in the standard literature (I am indebted to Bartek Klin for this observation). It should not be completely surprising then, if indexed valuations do not provide a good coalgebraic model.

5.5.2 Semantics of a functional language

We would like to use our construction to give a semantics to a functional language, an extension of PCF with probabilistic and nondeterministic choice. In order to do that we need to work in a cartesian closed category. Unfortunately the category **CONT** is not cartesian closed [AJ94]. The powerdomain of valuation does not preserve some important cartesian closed category of continuous domains, and it is not known to preserve any cartesian closed category besides **DCPO**.

We did not try to prove that the indexed valuations functor preserves any cartesian closed of continuous domains. This is still an interesting problem to work on, although Jung and Tix ([JT98]) warn us that it is not an easy problem.

5.5.3 Presheaf models

We could also try to find other denotational models beyond domain theory. The notion of indexed valuation, with its explicit reference to computational paths, seems to lead toward a presheaf semantics, known to be a nice framework for concurrent and higher order processes [NW03]. Can the notions studied in this thesis help in defining a probabilistic semantics in terms of presheaves?

Part III

Probability and Concurrency

Chapter 6

Valuations on Event Structures

The vast majority of models for probabilistic concurrent computation follow the interleaving approach. The only exception, to our knowledge, are Katoen's probabilistic event structures [Kat96]. The model we are going to present here is, in some sense, a special case of Katoen's. First of all Katoen builds on the notion of bundle labelled event structures, while our model is based on the more primitive notion of prime event structures, and does not contain labels. More importantly we restrict our attention to the class of *confusion free* event structures.

However, in this restricted setting, we are able to produce many interesting original contributions. In Katoen's model, as in every interleaving model, all probabilistic choices are assumed to be (probabilistically) independent. We are able to go beyond this limitation by constructing a model that accounts for correlation between choices.

Then we show a connection between our notion of probabilistic event structure and domain theory, extending the classic result of [Win80, NPW81]. This result can be interpreted as a representation theorem for continuous valuations on a certain class of domains.

We are able to define a notion of run for a probabilistic event structure, which encompasses and extends the corresponding notion in the interleaving framework. Finally we are able to show a confluence result for such runs, which also generalise classic results in the theory of event structures.

At the end we will discuss more on the implications and limitations of our work.

In this chapter we introduce the notions of event structure and of configuration of an event structure. We define the notion of confusion free event structure, arguing that they are suitable to be endowed with probability. We introduce probabilities on confusion free event structures using the notion of *global valuation.* A global valuation is a function assigning a weight to every finite configuration. The name is chosen because every global valuation generates a normalised continuous valuation on the domain of configurations. We start by defining a notion of global valuation which assumes probabilistic independence of all choices. We then remove this assumption and prove the main theorem relating global valuation and continuous valuations. We further generalise the definition of global valuation in order to include subprobability distributions. This allows us to characterise completely the normalised continuous valuations on the domain of configurations. Finally we perform an analysis of our definitions using the notion of morphism of event structures.

6.1 Event structures

Event structures were introduced by Nielsen, Plotkin and Winskel [NPW81, Win80, Win87] as a model for concurrency. Their main feature, which distinguishes them from transition systems, is the ability of recording causality and concurrency of events. States of event structures are represented by the notion of configuration. The set of configurations of an event structure form a DCPO.

6.1.1 Prime Event Structures

Definition 6.1.1. A *prime event structure* is a triple $\mathcal{E} = \langle E, \leq, \# \rangle$ such that

- E is a countable set of *events*;
- $\langle E, \leq \rangle$ is a partial order;
- for every $e \in E$, $\downarrow e$ is finite;
- $\#$ is an irreflexive and symmetric relation satisfying the following: for every $e_1, e_2, e_3 \in E$ if $e_1 \geq e_2$ and $e_2 \# e_3$ then $e_1 \# e_3$.

The relation $\#$ is called the *conflict* relation. Two events e_1, e_2 are *concurrent*, written $e_1 \bowtie e_2$, if $\neg e_1 \leq e_2, \neg e_2 \leq e_1, \neg e_1 \# e_2$.

Since prime event structures are the only kind of event structures we deal with in this thesis, we will refer to them simply as event structures.

The order relation of an event structure represents causality: an event e must happen before another event e' can occur ($e \leq e'$). Nondeterministic choice is represented by the conflict relation($e \# e'$). Concurrency between events is represented by absence of causality and conflict.

In what follows, the set of events of an event structure $\mathcal{E}$ will always be denoted by E, possibly carrying indices or dashes.

6.1.2 Configurations of an Event Structure

A state of an event structure is a set of events that have happened. If an event has happened, so must have all the events that it causally depends on. If two events are in conflict, at most one of them has happened.

Definition 6.1.2. A *configuration* x of an event structure $\mathcal{E}$ is a conflict-free downward closed subset of E, that is a subset x of E satisfying:

- whenever $e \in x$ and $e' \leq e$ then $e' \in x$.
- for every $e, e' \in x$, it is not the case that $e \# e'$

The set of configurations of $\mathcal{E}$, partially ordered by inclusion, is denoted as $\mathcal{L}(\mathcal{E})$. The set of finite configurations is denoted as $\mathcal{L}_{fin}(\mathcal{E})$.

The following is a well known fact that connects event structures with domain theory [NPW81].

Theorem 6.1.3. *The partial order $\langle \mathcal{L}(\mathcal{E}), \subseteq \rangle$ is an algebraic DCPO, whose compact elements are the finite configurations.*

This theorem can be refined by saying that such a DCPO is in fact a coherent dI-domain. Conversely every coherent dI-domain can be represented as the set of configurations of a prime event structure. Details can be found in [Win80].

Sometimes, in order to avoid ambiguity, we will use lattice notation for configurations. That is, we will write $x \leq y$ for $x \subseteq y$, $x \vee y$ for $x \cup y$, and $\perp$ for the empty configuration.

If x is a configuration and e is an event such that $e \not\in x$ and $x \cup \{e\}$ is a configuration, then we say that e is *enabled* at x. Two configurations x, x' are said to be *compatible* if $x \cup x'$ is a configuration. A special kind of configuration is the principal lower set generated by one event.

Definition 6.1.4. For every event e of an event structure $\mathcal{E}$, we define $[e] := \downarrow e$, and $[e) := [e] \setminus \{e\}$.

It is easy to see that both $[e]$ and $[e)$ are configurations for every event e. An event e is enabled at x if $[e) \subseteq x$ and for every $e' \in x$, e' is not in conflict with e.

An useful observation is the following:

Lemma 6.1.5. *Let x be a configuration of an event structure and let e be a maximal event in x. Then $x' := x \setminus \{e\}$ is a configuration and e is enabled in x'?*

Proof: Any subset of a conflict free set is conflict free. Moreover, since e is maximal, then x' is still downward closed. □

The *depth* of an event e is defined as follows. If $[e) = \emptyset$ then $depth(e) = 0$. Otherwise $depth(e) = \max\{depth(e') \mid e' < e\} + 1$. Since $\downarrow e$ is finite, it is clear that every event has a finite depth. This allows us to perform proofs by induction on the depth.

6.1.3 Linear Runs

Configurations represent both a state and a "non linear" run of an event structure, where the order in which concurrent events have happened is not recorded. We can also give a linear notion of a run.

Definition 6.1.6. Let x, x' be two configurations of an event structure. We write $x \xrightarrow{e} x'$ when $e \not\in x$, and $x' = x \cup \{e\}$. A sequence $\sigma = e_1 \dots e_n \in E^*$ is a *string* of the event structure $\mathcal{E}$, if there exist $x_1, \dots, x_n$ such that

$$\emptyset \xrightarrow{e_1} x_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} x_{n-1} \xrightarrow{e_n} x_n \, .$$

We define $Conf(\sigma) := x_n$. Notice that $Conf(\sigma) = \{e_1, \dots, e_n\}$. The *language* of an event structure $\mathcal{E}$ is the set of its strings, and it is denoted by $Str(\mathcal{E})$.

6.1.4 Immediate Conflict

Two events of an event structure may be in conflict by inheriting the conflict from previous events. We want to characterise the "minimal" conflicts that generate all other conflicts by inheritance.

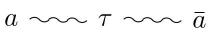
Definition 6.1.7. The *immediate conflict* relation $\#_\mu$ on an event structure $\mathcal{E} = \langle E, \leq \# \rangle$ is defined as follows: for every $e, e' \in E$, $e \#_\mu e'$ iff $e \# e'$ and $[e] \cup [e'), [e) \cup [e']$ are configurations.

Trivially the immediate conflict relation is symmetric.

6.2 Confusion free event structures

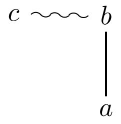
The idea for adding probabilities to event structures is to resolve the conflicts probabilistically. Whenever there is a set of events in immediate conflict, a die is rolled and, depending on the outcome, one of the events is chosen. However there are event structures where things do not go so smoothly. Consider the event structure $\mathcal{E}_{sym} = \langle E_{sym}, \leq, \# \rangle$ where $E_{sym} = \{a, \bar{a}, \tau\}$, the order relation is trivial, and $a \# \tau$, $\bar{a} \# \tau$.

Using a standard notation, we represent the partial order with the usual Hasse diagrams, while immediate conflict is represented by a curly line.



Note that $a, \bar{a}$ are concurrent. Which conflict do we resolve? If we flip a coin to choose between a and τ, the outcome also involves $\bar{a}$, contradicting the intuition that a and $\bar{a}$ are concurrent, and therefore unable to interfere with each other. This is an example of *symmetric confusion*. The first requirement we add is that the immediate conflict be transitive, so as to rule out symmetric confusion.

Next consider the event structure $\mathcal{E} = \langle E, \leq, \# \rangle$ where $E = a, b, c$, with $a \leq b$, and $b \# c$.



at the empty configuration c is enabled, but b is not, so there is no conflict to resolve between them. Once a has happened, though, b is also enabled, and a coin must be flipped. Again, the happening of a interferes with the concurrent event c. This is an example of *asymmetric confusion*. To rule out asymmetric confusion we require that whenever an event e is enabled, all the events in immediate conflict with e are enabled as well.

6.2.1 Confusion Freeness

The combination of the above requirements is known as *confusion freeness*. This notion originates from the theory of Petri Nets. For a discussion of its importance we refer also to [RT86].

Definition 6.2.1. An event structure $\mathcal{E}$ is *confusion free* if the following conditions are satisfied:

- $\#_\mu \cup 1_E$ (the reflexive closure of immediate conflict) is an equivalence;
- whenever $e \#_\mu e'$, then $[e) = [e')$.

Confusion free event structures are also known as *concrete data structures* [KP93] and are studied as a computational model in relation with the notion of *dataflow networks* [Kah74, KM77] and as a datatype to characterise sequentiality [Ong95].

The equivalence classes of $\#_\mu \cup 1_E$ are called *cells*. The set of cells of $\mathcal{E}$ is denoted by $\mathit{cell}(\mathcal{E})$. If c is a cell and $e \in c$ then we say that e is *located at* c. The second condition in the definition implies that whenever an event e located at c is enabled at a configuration x, all the events located at c are enabled as well. In such a case we say that the cell c is enabled. The depth of a cell is the depth of its events. We say that a configuration x *fills* a cell c if there exists $e \in x$ such that $e \in c$. We say that the cell c is *accessible* at x if c is enabled at x but not filled by x. The set of accessible cells at x is denoted by $\mathit{Acc}(x)$. We extend the partial order notation by writing $e < c'$ if for some event $e' \in c'$ (and therefore for all such) $e < e'$. We write $c < c'$ if for some (unique) event $e \in c$, $e < c'$. By $[c)$ we denote the set of events e such that $e < c$.

Lemma 6.2.2. *In an event structure, $e \# e'$ if and only if there exist e_0, e'_0 such that $e_0 \leq e, e'_0 \leq e', e_0 \#_\mu e'_0$.*

Proof: Consider the set $([e] \times [e']) \cap \#$ consisting of the pairs of conflicting events, and order it componentwise. Consider a minimal such pair (e_0, e'_0). By minimality any event in $[e_0)$ is not in conflict with any event in $[e'_0]$. Since they are both lower sets we have that $[e_0) \cup [e'_0]$ is a configuration. Analogously for $[e_0] \cup [e'_0)$. By definition $e_0 \#_\mu e'_0$. The other direction follows from the definition of $\#$. □

This allows us to characterise compatibility of configurations as follows.

Proposition 6.2.3. *In a confusion-free event structure, two configurations x, x' are compatible if and only if for every $e_0 \in x, e'_0 \in x'$, if e_0, e'_0 are located at the same cell, they are equal.*

Proof: The configuration x, x' are incompatible if and only if there are $e \in x, e' \in x'$ such that $e \# e'$. By Lemma 6.2.2 this happens if and only if there are e_0, e'_0 such that $e_0 \leq e, e'_0 \leq e', e_0 \#_\mu e'_0$. When the event structure is confusion-free, e_0, e'_0 belong to the same cell. □

The domain of configurations of a confusion free event structure is a distributive concrete domain [KP93]. Conversely every distributive concrete domain can be represented as the domain of configurations of a confusion free event structure.

6.2.2 Coverings

We will need the following notions.

Definition 6.2.4. Given two configurations $x, x' \in \mathcal{L}(\mathcal{E})$ we say that x' *covers* x (written $x \lhd x'$) if there exists $e \in E$ such that $x' = x \cup \{e\}$.

For every finite configuration x of a confusion-free event structure, a *covering* at x is a set C of configurations such that:

- for every $x' \in C$, $x \lhd x'$;
- for every $x', x'' \in C$, x', x'' are incompatible;
- for every configuration y, if $x \lhd y$ then there is $x' \in C$ such that y, x' are compatible.

Coverings are a device to define cells without referring to events.

Proposition 6.2.5. *If C is a covering at x, then $c = \{e \,|\, x' \in C, x' = x \cup \{e\}\}$ is a cell accessible at x. Conversely if $c \in Acc(x)$, then $C := \{x \cup \{e\} \,|\, e \in c\}$ is a covering.*

Proof: Let C be a covering at x, and let c be defined as above. Then for every distinct $e, e' \in c$, we have $e \# e'$, otherwise $x \cup \{e\}$ and $x \cup \{e'\}$ would be compatible. Moreover as $[e), [e') \subseteq x$, we have that $[e] \cup [e') \subseteq x \cup \{e\}$ so that $[e] \cup [e')$ is a configuration. Analogously $[e) \cup [e']$ is a configuration so that $e \#_\mu e'$. Now take $e \in c$ and suppose there is $e' \notin c$ such that $e \#_\mu e'$. Since $\#_\mu$ is transitive, then for every $e'' \in c$, $e' \#_\mu e''$. Therefore $x \cup \{e'\}$ is incompatible with every configuration in C, and $x \lhd x \cup \{e'\}$. Contradiction.

Conversely, take a cell $c \in Acc(x)$, and define C as above. Then clearly for every $x' \in C$, $x \lhd x'$ and also for every $x', x'' \in C$, x', x'' are incompatible. Now consider a configuration y, such that $x \lhd y$. This means $y = x \cup \{e\}$ for some e. If $e \in c$ then $y \in C$ and y is compatible with itself. If $e \notin c$ then for every $e' \in c$, e, e' are not in immediate conflict. Suppose $e \# e'$, then, by lemma 6.2.2 there are $d \leq e, d' \leq e'$ such that $d \#_\mu d'$. Suppose $d < e$ then $[e) \cup [e']$ would not be a conflict free. But that is not possible as $[e) \cup [e'] \subseteq x \cup \{e'\}$ and the latter is a configuration. Analogously it is not the case that $d' < e'$. Therefore $e \#_\mu e'$ contradiction. Therefore for every $x \in C$, y and x are compatible. $\Box$

Because of the previous lemma, every covering C uniquely determines a cell c. In this case we say that C is a c-covering.

For more details on event structures, one possible reference is [Win87].

6.3 Valuations on Event Structures

We define two notions of global valuation on a confusion free event structure. The first one assuming independence of all random choices, the second, more general one, removing this assumption.

6.3.1 Valuations with independence

In a confusion free event structure, every choice is localised at some cell. We can resolve such choice by a probability distribution over the events in that cell.

Definition 6.3.1. A *local valuation* on a confusion-free event structure $\mathcal{E} := \langle E, \leq, \# \rangle$ is a function $p : E \to]0, 1]$ such that for every cell c, $\sum_{e \in c} p(e) = 1$.

Note that no event gets probability 0. Intuitively this is justified by the fact that an event with probability 0 cannot happen, so we may as well dispense with it.

If we assume that all the probabilistic choices are independent, the probability of a configuration is the product of the probabilities of its events. Given a local valuation p we can define a function $v_p : \mathcal{L}_{fin}(\mathcal{E}) \to]0, 1]$ by putting $v_p(x) = \prod_{e \in x} p(e)$. We are now going to characterise more abstractly the functions arising in this way.

Definition 6.3.2. A *global valuation with independence* on a confusion free event structure $\mathcal{E}$ is a function $v : \mathcal{L}_{fin}(\mathcal{E}) \to]0, 1]$ such that:

a) $v(\emptyset) = 1$;

b) if C is a covering at x, then $\sum_{x' \in C} v(x') = v(x)$;

c) if x, y are compatible, then $v(x \cup y) = v(x) \cdot v(y)/v(x \cap y)$.

The following lemma points out an important feature of global valuations with independence. Its statement applies to more general notions, and we will use it later.

Lemma 6.3.3. *If $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0, 1]$ satisfies condition b) above, then it is contravariant, i.e.:*

$$x \subseteq x' \Longrightarrow v(x) \geq v(x')$$

Proof: By induction on the cardinality of $x' \setminus x$. If $x = x'$ then $v(x) = v(x')$. Take $x \subseteq x'$ and consider a maximal event e in $x' \setminus x$. Let $x'' := x' \setminus \{e\}$. By induction hypothesis $v(x) \geq v(x'')$. Let c be the cell of e and C be the c-covering of x''. By condition b), $\sum_{y \in C} v(y) = v(x'')$. Since for every $y \in C$ we have that $v(y) \geq 0$, then it must also be that $v(y) \leq v(x'')$. But $x' \in C$ so that $v(x') \leq v(x'') \leq v(x)$. □

Proposition 6.3.4. *Let $\mathcal{E}$ be a confusion free event structure. Then*

i) a local valuation p on $\mathcal{E}$ determines a global valuation with independence v by taking $v_p(x) = \Pi_{e \in x} p(e)$;

ii) a global valuation with independence v on $\mathcal{E}$ determines a local valuation by taking $p_v(e) = v([e])/v([e))$.

Moreover, the operations of (i) and (ii) are mutually inverse.

Proof: Part i) is straightforward. As for part ii), because of contravariance we have that $v([e]) \leq v([e))$, so that $p_v(e) \in]0, 1]$. Consider now a cell c. Then the set $C := \{[c) \cup \{e\} \mid e \in c\}$ is a covering at $[c)$. Remember that if $e \in c$, then $[e) = [c)$. Therefore

$$\sum_{e \in c} p_v(e) = \sum_{e \in c} v([e])/v([e))$$

$$= \sum_{e \in c} v([e])/v([c)) = \sum_{x \in C} v(x)/v([c)) = 1\,.$$

We want now to show that $p_{v_p} = p$. Take an event e.

$$p_{v_p}(e) = v_p([e])/v_p([e))$$
$$= \prod_{e' \in [e]} p(e') / \prod_{e' \in [e)} p(e') = p(e)\,.$$

In order to show that $v_{p_v} = v$ we proceed by induction on the size of the configurations. Because of property a), we have that

$$v_{p_v}(\emptyset) = \prod_{e \in \emptyset} p_v(e) = 1 = v(\emptyset)\,.$$

Now assume that for every configuration y of size n, $v_{p_v}(y) = v(y)$, take a configuration x of size $n+1$. Take a maximal event $e \in x$ so that $y := x \setminus \{e\}$ is still a configuration. Since x is a configuration, it must be that $[e] \subseteq x$ and thus $[e) \subseteq y$. Therefore $[e) = y \cap [e]$. Now

$$v_{p_v}(x) = \prod_{e' \in x} p_v(e') = p_v(e) \cdot \prod_{e' \in y} p_v(e')$$
$$= \big(v([e])/v([e))\big) \cdot v_{p_v}(y)$$

By induction hypothesis this is equal to

$$= \big(v([e])/v([e))\big) \cdot v(y) = v([e]) \cdot v(y)/v([e))$$
$$= v([e]) \cdot v(y)/v(y \cap [e])$$

And because of property c) this is equal to

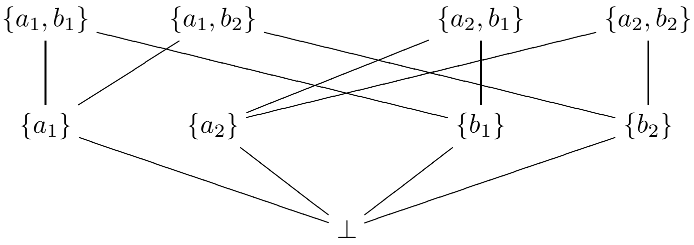
$$= v(y \cup [e]) = v(x)\,.$$

□

We show an example. Take the following confusion-free event structure $\mathcal{E}_{a,b}$: $E_{a,b} = \{a_1, a_2, b_1, b_2\}$ with the flat ordering and with $a_1 \# a_2$ and $b_1 \# b_2$.

$$a_1 \sim\!\sim\!\sim a_2 \qquad\qquad b_1 \sim\!\sim\!\sim b_2$$

Then $\mathcal{L}(\mathcal{E}_{a,b})$ is as follows



We define a local valuation on $\mathcal{E}_{a,b}$ by $p(a_1) = 1/3, p(a_2) = 2/3, p(b_1) = 1/4, p(b_2) = 3/4$. The corresponding global valuation is defined as

$$v_p(\perp) = 1$$

$$v_p(\{a_1\}) = 1/3, v_p(\{a_2\}) = 2/3, v_p(\{b_1\}) = 1/4, v_p(\{b_2\}) = 3/4$$

$$v_p(\{a_1, b_1\}) = 1/12, v_p(\{a_2, b_1\}) = 1/6, v_p(\{a_1, b_2\}) = 1/4, v_p(\{a_2, b_2\}) = 1/2$$

In the above example, a covering at $\perp$ is $\{a_1\}, \{a_2\}$. A covering at $\{a_1\}$ is $\{a_1, b_1\}, \{a_1, b_2\}$.

Definition 6.3.5. A *probabilistic event structure with independence* is a confusion free event structure together with a global valuation with independence (or, equivalently, a local valuation).

6.3.2 Valuations without independence

We will see later that every global valuation with independence v on $\mathcal{E}$ can be extended to a continuous valuation ν on $\mathcal{L}(\mathcal{E})$ with the property that for every finite configuration x, $v(x) = \nu(\uparrow x)$. Not every continuous valuation arises in this way. As an example, consider the following continuous valuation on $\mathcal{E}_{a,b}$. Define

- $\nu(\uparrow \emptyset) = 1$
- $\nu(\uparrow\{a_1\}) = \nu(\uparrow\{a_2\}) = \nu(\uparrow\{b_1\}) = \nu(\uparrow\{b_2\}) = 1/2$
- $\nu(\uparrow\{a_1, b_1\}) = \nu(\uparrow\{a_2, b_2\}) = 0$
- $\nu(\uparrow\{a_1, b_2\}) = \nu(\uparrow\{a_2, b_1\}) = 1/2$

and extend ν to all open sets by modularity. Define a function $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0, 1]$ by $v(x) := \nu(\uparrow x)$. This is clearly not a global valuation with independence. For a start it takes on the value 0. More importantly it does not satisfy condition c). If we consider the compatible configurations $x := \{a_1\}, y := \{b_1\}$ then $v(x \cup y) = 0 < 1/4 = v(x) \cdot v(y)/v(x \cap y)$.

Condition c) characterises independence. In the example above, the probabilistic choices in the two cells are not independent: there is a positive correlation between the occurrence of a_1 and the occurrence of b_1. We can think of the above probabilistic event structure as representing two entangled bits. Once one of them is observed, we also know the state of the other.

This observation leads us to a more general definition of probabilistic event structure.

Definition 6.3.6. A *global valuation* on a confusion-free event structure $\mathcal{E}$ is a function $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0, 1]$ such that:

a) $v(\emptyset) = 1$;

b) if C is a covering at x, then $\sum_{x' \in C} v(x') = v(x)$.

Note, in particular, that a global valuation can take the value 0. This more general notion is not reducible to a function on events only.

Definition 6.3.7. A *probabilistic event structure* is a confusion free event structure together with a global valuation.

This is now a good level of generality in a sense to be justified in the next section.

6.4 Probabilistic Event Structures and Domains

Theorem 6.1.3 showed a connection between event structures and domain theory. The main result of this chapter extends that theorem drawing a connection between global valuations and continuous valuations on the domain of configurations of an event structure.

In order to give a continuous valuation, we have to give a weight to open sets. Intuitively, open sets represent observations. (A good discussion of this point of view can be found, for example, in [Abr87].) A principal open set $\uparrow x$ represents the observation of x. A global valuation provides a weight for finite configurations. It is reasonable to ask that the weight of $\uparrow x$ be the weight of x. It turns out that this assignment can be extended in a unique way to all open sets.

Theorem 6.4.1. *For every global valuation v on $\mathcal{E}$ there is a unique continuous valuation ν on $\mathcal{L}(\mathcal{E})$ such that for every finite configuration x, $\nu(\uparrow x) = v(x)$.*

We will characterise later the continuous valuations arising in this way as the *maximal* elements of $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$.

The proof of Theorem 6.4.1 will require various intermediate results. In the following proofs we will write $\widehat{x}$ for $\uparrow x$. To avoid complex case distinctions we also introduce a special element $\top$ representing an impossible configuration. If x, y are incompatible, the expression $x \vee y$ will denote $\top$. Also, for every global valuation v, $v(\top) = 0$, finally $\widehat{\top} = \emptyset$. The finite configurations together with $\top$ form a $\vee$-semilattice.

We have to define a function from the Scott open sets of $\mathcal{L}(\mathcal{E})$ to the unit interval. This value of ν on the principal open sets is determined by $\nu(\widehat{x}) = v(x)$. We first define ν on finite unions of principal open sets. Since $\mathcal{L}(\mathcal{E})$ is algebraic, such sets form a basis of the Scott topology of $\mathcal{L}(\mathcal{E})$. We will then be able to define ν on all open sets by continuity.

Let Pn be the set of principal open subsets of $\mathcal{L}(\mathcal{E})$. That is

$$Pn = \{\widehat{x} \mid x \in \mathcal{L}_{fin}(\mathcal{E})\} \cup \{\emptyset\}\,.$$

Notice that Pn is closed under finite intersection because $\widehat{x} \cap \widehat{y} = \widehat{x \vee y}$. (If x, y are not compatible then $\widehat{x} \cap \widehat{y} = \emptyset = \widehat{\top} = \widehat{x \vee y}$.) The family Pn is, in general, not closed under finite union.

Let Bs be the set of finite unions of elements of Pn. That is

$$Bs = \{\widehat{x_1} \cup \ldots \cup \widehat{x_n} \mid \widehat{x_i} \in Pn,\ 1 \leq i \leq n\}\,.$$

Using distributivity of intersection over union it is easy to prove the following.

Lemma 6.4.2. *The structure $\langle Bs, \cup, \cap\rangle$ is a distributive lattice with top and bottom.*

Since the ν has to be modular, it will also satisfy the inclusion-exclusion principle (Proposition 2.5.5). We exploit this to define ν. Let us define $\nu_0 : Bs \to \mathbb{R}$ as follows

$$\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) = \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right)\,.$$

We have first to make sure that ν_0 is well defined: If two expressions $\widehat{x_1} \cup \ldots \cup \widehat{x_n}$ and $\widehat{y_1} \cup \ldots \cup \widehat{y_m}$ represent the same set, then

$$\sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) = \sum_{\emptyset \neq J \subseteq I_m} (-1)^{|J|-1} v\left(\bigvee_{j \in J} y_j\right) .$$

Lemma 6.4.3. *We have $\widehat{x} \subseteq \widehat{x_1} \cup \ldots \cup \widehat{x_n}$ if and only if there exists i such that $x_i \leq x$.*

Proof: Straightforward. □

Lemma 6.4.4. *If $x_n \leq x_{n+1}$ then*

$$\sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) = \sum_{\emptyset \neq I \subseteq I_{n+1}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) .$$

Proof: When $x_n \leq x_{n+1}$ we have that $x_n \vee x_{n+1} = x_{n+1}$. Now

$$\begin{aligned}
&\sum_{\emptyset \neq I \subseteq I_{n+1}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) \\
&= \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) \\
&+ \sum_{\substack{I \subseteq I_{n+1} \\ n, n+1 \in I}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) \\
&+ \sum_{\substack{I \subseteq I_{n+1} \\ n \notin I, n+1 \in I}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) .
\end{aligned}$$

We claim that

$$\sum_{\substack{I \subseteq I_{n+1} \\ n, n+1 \in I}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\substack{I \subseteq I_{n+1} \\ n \notin I, n+1 \in I}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) = 0$$

and this would prove our lemma.

To prove the claim

$$\begin{aligned}
&\sum_{\substack{I\subseteq I_{n+1}\\ n,n+1\in I}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} x_i\right)\\
&= \sum_{I\subseteq I_{n-1}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} x_i \vee x_n \vee x_{n+1}\right)\\
&= \sum_{I\subseteq I_{n-1}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} x_i \vee x_{n+1}\right)\\
&= -\sum_{I\subseteq I_{n-1}} (-1)^{|I|} v\left(\bigvee_{i\in I} x_i \vee x_{n+1}\right)\\
&= -\sum_{\substack{I\subseteq I_{n+1}\\ n\notin I, n+1\in I}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} x_i\right)
\end{aligned}$$

□

Therefore we can safely remove "redundant" components from a finite union until we are left with a minimal expression. The next lemma says that such minimal expression is unique, up to the order of the components.

Lemma 6.4.5. *Let $\widehat{x_1} \cup \ldots \cup \widehat{x_n} = \widehat{y_1} \cup \ldots \cup \widehat{y_m}$, and let such expressions be minimal. Then $n = m$ and there exists a permutation σ of I_n such that $x_i = y_{\sigma(i)}$.*

Proof: By lemma 6.4.3, for every $i \in I_n$ there exist some $j \in I_m$ such that $y_j \leq x_i$. Let $\sigma : I_n \to I_m$ be a function choosing one such j. Symmetrically let $\tau : I_m \to I_n$ be such that $x_{\tau(j)} \leq y_j$. Now I claim that for every i, $\tau(\sigma(i)) = i$. In fact $x_{\tau(\sigma(i))} \leq y_{\sigma(i)} \leq x_i$. The minimality of the x_i's implies the claim. Symmetrically $\sigma(\tau(j)) = j$, so that σ is indeed a bijection. □

Finally we observe that in the definition of ν_0, the order of the x_i does not matter. This concludes the proof of that ν_0 is well-defined.

Next we state a lemma saying that $\nu_0 : Bs \to \mathbb{R}$ is a valuation on the lattice $\langle Bs, \cup, \cap\rangle$. This is the crux of the proof of Theorem 6.4.1.

Lemma 6.4.6. *The function $\nu_0 : Bs \to \mathbb{R}$ satisfies the following properties:*

- *(Strictness)*
  $\nu_0(\emptyset) = 0$;
- *(Monotonicity)*
  $U \subseteq V \Longrightarrow \nu_0(U) \leq \nu_0(V)$;
- *(Modularity)*
  $\nu_0(U) + \nu_0(V) = \nu_0(U \cup V) + \nu_0(U \cap V)$.

In particular, since $\widehat{\bot} = \mathcal{L}(\mathcal{E})$, for every $U \in Bs$, we have $0 = \nu_0(\emptyset) \leq \nu_0(U) \leq \nu_0(\mathcal{L}(\mathcal{E})) = \nu_0(\widehat{\bot}) = v(\bot) = 1$. So in fact $\nu_0 : Bs \to [0, 1]$.

Proof: Strictness is obvious.

We prove monotonicity in steps. First we prove a special case. That is for every n-tuple of finite configurations (x_i) and for every finite configuration y, if $\widehat{x_1} \cup \ldots \cup \widehat{x_n} \subseteq \widehat{y}$, then $\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \leq \nu_0(\widehat{y})$. We will do it by induction on n. The basis requires that $0 = \nu_0(\emptyset) \leq \nu_0(\widehat{y}) = v(y)$ which is true.

Suppose now that $\widehat{x_1} \cup \ldots \cup \widehat{x_{n+1}} \subseteq \widehat{y}$. Fix y and consider all $n+1$-tuples (z_i) such that $\widehat{z_1} \cup \ldots \cup \widehat{z_{n+1}} \subseteq \widehat{y}$ and order them componentwise. That is $(z_i) \leq (z_i')$ if for every i, $z_i \leq z_i'$. Note that if $(z_i) > (z_i')$ then some of the (z_i') must be strictly smaller than some of the z_i. As every z_i is finite this order is well founded.

Suppose by contradiction that there exist an $n+1$-tuples for which

$$\nu_0\left(\widehat{z_1} \cup \ldots \cup \widehat{z_{n+1}}\right) > \nu_0(\widehat{y})$$

and take a minimal such. If this is the case, then all z_i must be strictly greater than y. We argue that there is a cell c, such that y does not fill c, some of the z_i's fill c and for all z_i that do, the event $e \in c \cap z_i$ is maximal in z_i. Consider a maximal event $e_1 \in z_1 \setminus y$. If the cell c_1 of e_1 is maximal in all z_j that fill c_1, then we are done. Otherwise consider the first z_j that fills c_1 but for which c_1 is not maximal. Consider a maximal event in z_j lying above c_1. Consider its cell c_2. Since c_2 is above c_1, clearly c_2 cannot be filled by any of the z_i for $i < j$ because, either they do not fill c_1, or if they do, then c_1 is maximal. Continue this process until you reach z_{n+1} at which point we will have found a cell c with the properties above.

Consider all the events $e_1, \ldots, e_h, \ldots \in c$.[1] For every $h \geq 1$ let $I^h = \{i \in I_{n+1} \mid e_h \in z_i\}$. Since c is maximal and it is not filled by y, then we have that for every $i \in I^h$, $z_i' := z_i \setminus \{e_h\}$ is still a configuration and it is still above y.

For every $i \in I_{n+1}$ let w_i be z_i' if i belongs to some I^h, and otherwise let w_i be z_i. For what we have said, all w_i are greater than y so that $\widehat{w_1} \cup \ldots \cup \widehat{w_{n+1}} \subseteq \widehat{y}$. Also the tuple (w_i) is strictly below (z_i) in the well order defined above. We now show that

$$\nu_0\left(\widehat{w_1} \cup \ldots \cup \widehat{w_{n+1}}\right) > \nu_0(\widehat{y})$$

which contradicts minimality.

To do that we show that

$$\nu_0\left(\widehat{w_1} \cup \ldots \cup \widehat{w_{n+1}}\right) \geq \nu_0\left(\widehat{z_1} \cup \ldots \cup \widehat{z_{n+1}}\right) .$$

That is

$$\sum_{\emptyset \neq I \subseteq I_{n+1}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} w_i\right) \geq \sum_{\emptyset \neq I \subseteq I_{n+1}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i\right) .$$

We can start erasing summands that do not change. Let $\tilde{I} = I_{n+1} \setminus \bigcup_{h \geq 1} I^h$ For every $i \in \tilde{I}$, $w_i = z_i$, thus if $I \subseteq \tilde{I}$ then $\bigvee_{i \in I} w_i = \bigvee_{i \in I} z_i$. So that

$$v\left(\bigvee_{i \in I} w_i\right) = v\left(\bigvee_{i \in I} z_i\right) .$$

[1] Cells can be finite or countable. We do the proof for the countable case, the finite case being analogous and, in fact, simpler.

Removing the summands of the above shape, it is enough to prove that

$$\sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} w_i\right) \geq \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i\right) .$$

Also note that if for two different $h, h' \geq 1$ we have that, if $I \cap I^h \neq \emptyset$ and $I \cap I^{h'} \neq \emptyset$ then $\bigvee_{i \in I} z_i = \top$, that is $v\left(\bigvee_{i \in I} z_i\right) = 0$, because it is the join of incompatible configurations. Therefore we can rewrite the right-hand member of the inequation above as

$$\sum_{h \geq 1} \sum_{\emptyset \neq I \setminus \tilde{I} \subseteq I^h} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i\right) .$$

For every $i \notin \tilde{I}$ we can define z_i^h to be $w_i \cup \{e_h\}$. All such z_i^h are indeed configurations because if $i \notin \tilde{I}$ then c is accessible at w_i. For every I such that $\emptyset \neq I \setminus \tilde{I}$ we have that $\bigvee_{i \in I} z_i^h = \top$ if and only if $\bigvee_{i \in I} w_i = \top$ as e_h is the only event in its cell appearing in any configuration, so its introduction cannot cause an incompatibility that was not already there. Now condition b) in the definition of global valuation says exactly that

$$v\left(\bigvee_{i \in I} w_i\right) = \sum_{h \geq 1} v\left(\bigvee_{i \in I} z_i^h\right) .$$

(Where both members may be 0 if $\bigvee_{i \in I} w_i$ is already $\top$.) Therefore

$$\sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} \sum_{h \geq 1} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i^h\right) = \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} w_i\right) .$$

Now, the left hand member is absolutely convergent, because v is a nonnegative function and

$$\sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} \sum_{h \geq 1} v\left(\bigvee_{i \in I} z_i^h\right) = \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} v\left(\bigvee_{i \in I} w_i\right) < +\infty .$$

Therefore we can rearrange the terms as we like, in particular we can swap the two summations symbols. Thus

$$\sum_{h \geq 1} \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i^h\right) = \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} w_i\right) .$$

So to prove our claim it is enough to show that

$$\sum_{h \geq 1} \sum_{\emptyset \neq I \setminus \tilde{I} \subseteq I^h} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i\right) \leq \sum_{h \geq 1} \sum_{\substack{\emptyset \neq I \subseteq I_{n+1} \\ I \setminus \tilde{I} \neq \emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} z_i^h\right) .$$

Note that if $I \setminus \tilde{I} \subseteq I^h$ then $\bigvee_{i\in I} z_i = \bigvee_{i\in I} z_i^h$. Therefore we can rewrite the inequation as:

$$\sum_{h\geq 1} \sum_{\emptyset\neq I\setminus\tilde{I}\subseteq I^h} (-1)^{|I|-1} v\left(\bigvee_{i\in I} z_i^h\right) \leq \sum_{h\geq 1} \sum_{\substack{\emptyset\neq I\subseteq I_{n+1}\\ I\setminus\tilde{I}\neq\emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} z_i^h\right).$$

To prove the inequation holds, it is then enough to show that for any $h \geq 1$.

$$\sum_{\emptyset\neq I\setminus\tilde{I}\subseteq I^h} (-1)^{|I|-1} v\left(\bigvee_{i\in I} z_i^h\right) \leq \sum_{\substack{\emptyset\neq I\subseteq I_{n+1}\\ I\setminus\tilde{I}\neq\emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} z_i^h\right).$$

Subtracting the same quantity from both members we get equivalently

$$0 \leq \sum_{\substack{\emptyset\neq I\subseteq I_{n+1}\\ I\setminus(\tilde{I}\cup I^h)\neq\emptyset}} (-1)^{|I|-1} v\left(\bigvee_{i\in I} z_i^h\right).$$

Let $\tilde{I}^h := \bigcup_{l\neq h} I^l$. We can rewrite the sum above as

$$\sum_{\emptyset\neq J\subseteq\tilde{I}^h} \sum_{H\subseteq\tilde{I}\cup I^h} (-1)^{|H|+|J|-1} v\left(\bigvee_{i\in H\cup J} z_i^h\right)$$

$$= \sum_{\emptyset\neq J\subseteq\tilde{I}^h} (-1)^{|J|-1} \sum_{H\subseteq\tilde{I}\cup I^h} (-1)^{|H|} v\left(\bigvee_{i\in H\cup J} z_i^h\right).$$

Using BSV lemma (6.App.2) we can rewrite this as

$$\sum_{\emptyset\neq K\subseteq\tilde{I}^h} \sum_{K\subseteq J\subseteq\tilde{I}^h} (-1)^{|J|+|K|} \sum_{H\subseteq\tilde{I}\cup I^h} (-1)^{|H|} v\left(\bigvee_{i\in H\cup J} z_i^h\right)$$

$$= \sum_{\emptyset\neq K\subseteq\tilde{I}^h} \sum_{K\subseteq J\subseteq\tilde{I}^h} \sum_{H\subseteq\tilde{I}\cup I^h} (-1)^{|K|+|J\cup H|} v\left(\bigvee_{i\in H\cup J} z_i^h\right)$$

Fix K. Consider a set I such that $K \subseteq I \subseteq I_{n+1}$. Since $\tilde{I}^h, \tilde{I}\cup I^h$ are a partition of I_{n+1}, we have that $H := I\cap(\tilde{I}\cup I^h)$ and $J := I\cap\tilde{I}^h$ are a partition of I. We use this to rewrite the term above.

$$= \sum_{\emptyset\neq K\subseteq\tilde{I}^h} \sum_{K\subseteq I\subseteq I_{n+1}} (-1)^{|I|+|K|} v\left(\bigvee_{i\in I} z_i^h\right).$$

For every K, and defining $L := I \setminus K$, we have that

$$\begin{aligned}
&\sum_{K\subseteq I\subseteq I_{n+1}} (-1)^{|I|+|K|} v\left(\bigvee_{i\in I} z_i^h\right)\\
&= \sum_{L\subseteq I_{n+1}\setminus K} (-1)^{|L|+2|K|} v\left(\bigvee_{i\in K} z_i^h \vee \bigvee_{j\in L} z_j^h\right)\\
&= (-1)^{0+2|K|} v\left(\bigvee_{i\in K} z_i^h\right) + \sum_{\emptyset\neq L\subseteq I_{n+1}\setminus K} (-1)^{|L|+2|K|} v\left(\bigvee_{j\in L}(z_j^h \vee \bigvee_{i\in K} z_i^h)\right)\\
&= v\left(\bigvee_{i\in K} z_i^h\right) + \sum_{\emptyset\neq L\subseteq I_{n+1}\setminus K} (-1)^{|L|} v\left(\bigvee_{j\in L}(z_j^h \vee \bigvee_{i\in K} z_i^h)\right)\\
&= v\left(\bigvee_{i\in K} z_i^h\right) - \sum_{\emptyset\neq L\subseteq I_{n+1}\setminus K} (-1)^{|L|-1} v\left(\bigvee_{j\in L}(z_j^h \vee \bigvee_{i\in K} z_i^h)\right) .
\end{aligned}$$

If $\bigvee_{i\in K} z_i^h = \top$ then the whole sum is equal to 0. Otherwise it is equal to

$$\nu_0\left(\widehat{\bigvee_{i\in K} z_i^h}\right) - \nu_0\left(\bigcup_{j\in I_{n+1}\setminus K} \widehat{z_j^h \vee \bigvee_{i\in K} z_i^h}\right) .$$

Note that for every j is

$$\widehat{z_j^h \vee \bigvee_{i\in K} z_i^h} \subseteq \widehat{\bigvee_{i\in K} z_i^h}$$

so that

$$\bigcup_{j\in I_{n+1}\setminus K} (\widehat{z_j^h \vee \bigvee_{i\in K} z_i^h}) \subseteq \widehat{\bigvee_{i\in K} z_i^h} .$$

Moreover observe that $|I_{n+1} \setminus K| < n+1$. By induction hypothesis

$$\nu_0\left(\widehat{\bigvee_{i\in K} z_i^h}\right) - \nu_0\left(\bigcup_{j\in I_{n+1}\setminus K} \widehat{z_j^h \vee \bigvee_{i\in K} z_i^h}\right) \geq 0 .$$

Thus we have proved that for every n-tuple of finite configurations (x_i) and for every finite configuration y, if $\widehat{x_1} \cup \ldots \cup \widehat{x_n} \subseteq \widehat{y}$, then $\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \leq \nu_0(\widehat{y})$.

Monotonicity now follows from the following lemma:

Lemma 6.4.7. *If $x_1, \ldots, x_{n+1}$ are finite configurations*

$$\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \leq \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{x_{n+1}}\right) .$$

Proof:

$$
\begin{aligned}
&\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{x_{n+1}}\right)\\
&= \sum_{\emptyset \neq I \subseteq I_{n+1}} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right)\\
&= \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + v(x_{n+1}) - \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(x_{n+1} \vee \bigvee_{i \in I} x_i\right)\\
&= \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + v(x_{n+1}) - \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_{n+1} \vee x_i\right)\\
&= \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) + \nu_0\left(\widehat{x_{n+1}}\right) - \nu_0\left(\widehat{x_{n+1} \vee x_1} \cup \ldots \cup \widehat{x_{n+1} \vee x_n}\right)\\
&\geq \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) .
\end{aligned}
$$

Therefore, by induction on m,

$$\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \leq \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{y_1} \cup \ldots \cup \widehat{y_m}\right) .$$

Finally, to show monotonicity of ν_0, suppose that

$$\widehat{x_1} \cup \ldots \cup \widehat{x_n} \subseteq \widehat{y_1} \cup \ldots \cup \widehat{y_m} .$$

Then

$$\widehat{y_1} \cup \ldots \cup \widehat{y_m} = \widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{y_1} \cup \ldots \cup \widehat{y_m} .$$

By the above observation we have

$$
\begin{aligned}
\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) &\leq \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{y_1} \cup \ldots \cup \widehat{y_m}\right)\\
&= \nu_0\left(\widehat{y_1} \cup \ldots \cup \widehat{y_m}\right) .
\end{aligned}
$$

□(6.4.7) To prove modularity take $\widehat{x_1} \cup \ldots \cup \widehat{x_n}$ and $\widehat{y_1} \cup \ldots \cup \widehat{y_m}$, we want to prove that

$$
\begin{aligned}
&\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) + \nu_0\left(\widehat{y_1} \cup \ldots \cup \widehat{x_m}\right)\\
&= \nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n} \cup \widehat{y_1} \cup \ldots \cup \widehat{x_m}\right) + \nu_0\left(\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \cap \left(\widehat{y_1} \cup \ldots \cup \widehat{x_m}\right)\right) .
\end{aligned}
$$

By distributivity we have that

$$
\begin{aligned}
&\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \cap \left(\widehat{y_1} \cup \ldots \cup \widehat{x_m}\right)\\
&= \left(\widehat{x_1} \cap \widehat{y_1}\right) \cup \left(\widehat{x_1} \cap \widehat{y_2}\right) \cup \ldots \cup \left(\widehat{x_n} \cap \widehat{y_m}\right) .
\end{aligned}
$$

Using the definitions, we have to prove that

$$R := \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\emptyset \neq J \subseteq I_m} (-1)^{|I|-1} v\left(\bigvee_{i \in I} y_j\right)$$

is equal to

$$
\begin{aligned}
L \quad := \quad & \sum_{\substack{\emptyset \neq I \uplus J \\ I \subseteq I_n, J \subseteq I_m}} (-1)^{|I \uplus J|-1} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right)\\
+ \quad & \sum_{\emptyset \neq K \subseteq I_n \times I_m} (-1)^{|K|-1} v\left(\bigvee_{(i,j) \in K} (x_i \vee y_j)\right)
\end{aligned}
$$

We can split the various $I \uplus J$ in three classes: when J is empty, when I is empty, and when both are not empty. So we can rewrite L as

$$\begin{aligned}
L &= \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) \\
&+ \sum_{\emptyset \neq J \subseteq I_m} (-1)^{|I|-1} v\left(\bigvee_{j \in J} y_j\right) \\
&+ \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} (-1)^{|I \uplus J|-1} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right) \\
&+ \sum_{\emptyset \neq K \subseteq I_n \times I_m} (-1)^{|K|-1} v\left(\bigvee_{(i,j) \in K} (x_i \vee y_j)\right)
\end{aligned}$$

The first two summands of this expression are equal to R, so we have just to prove that the last two are equal to 0.

For every $\emptyset \neq I \subseteq I_n$, $\emptyset \neq J \subseteq I_m$ consider all $K \subseteq I_n \times I_m$ such that $\pi_1(K) = I, \pi_2(K) = J$. We argue that for all such K,

$$\bigvee_{(i,j) \in K} (x_i \vee y_j) = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j$$

In fact using commutativity, associativity and idempotency of the join, we can group all the x_i and y_j on the left hand member. So that

$$\bigvee_{(i,j) \in K} (x_i \vee y_j) = \bigvee_{i \in \pi_1(K)} x_i \vee \bigvee_{j \in \pi_2(K)} y_j$$

We can rewrite the the last two summands of the above expression as

$$\begin{aligned}
& \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} (-1)^{|I \uplus J|-1} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right) \\
& + \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} \sum_{\substack{\emptyset \neq K \subseteq I_n \times I_m \\ \pi_1(K)=I, \pi_2(K)=J}} (-1)^{|K|-1} v\left(\bigvee_{(i,j) \in K} (x_i \vee y_j)\right) \\
= & \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} (-1)^{|I \uplus J|-1} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right) \\
& + \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} \sum_{\substack{\emptyset \neq K \subseteq I_n \times I_m \\ \pi_1(K)=I, \pi_2(K)=J}} (-1)^{|K|-1} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right) \\
= & \sum_{\substack{\emptyset \neq I \subseteq I_n \\ \emptyset \neq J \subseteq I_m}} v\left(\bigvee_{i \in I} x_i \vee \bigvee_{j \in J} y_j\right) \left((-1)^{|I \uplus J|-1} + \sum_{\substack{\emptyset \neq K \subseteq I_n \times I_m \\ \pi_1(K)=I, \pi_2(K)=J}} (-1)^{|K|-1}\right)
\end{aligned}$$

So it is enough to prove that for every finite sets I, J

$$(-1)^{|I \uplus J|-1} + \sum_{\substack{\emptyset \neq K \subseteq I_n \times I_m \\ \pi_1(K)=I, \pi_2(K)=J}} (-1)^{|K|-1} = 0$$

which is the statement of Lemma 6.App.1, to be proved in the appendix. □

Now we are ready to define ν on all Scott open sets.

Lemma 6.4.8. *For every Scott open $O \subseteq \mathcal{L}(\mathcal{E})$, we have*

$$O = \bigsqcup_{\substack{U \subseteq O \\ U \in Bs}}^{\uparrow} U \,.$$

Proof: Directedness is straightforward. Moreover, since $\mathcal{L}(\mathcal{E})$ is algebraic, Pn is a basis for the Scott topology (and so is, a fortiori, Bs). □

Now, for every Scott open set O, define

$$\nu(O) = \sup_{\substack{U \subseteq O \\ U \in Bs}} \nu_0(U) \,.$$

We then have the following proposition, which concludes the proof of Theorem 6.4.1.

Proposition 6.4.9. *The function ν is a valuation on the Scott-topology of $\mathcal{L}(\mathcal{E})$ such that for every finite configuration x, $\nu(\uparrow x) = v(x)$.*

Continuity follows from an exchange of suprema, strictness and monotonicity are obvious. Modularity follows from the modularity of ν_0 and continuity of the addition. Finally, because of the monotonicity of ν_0, we have that $\nu(\uparrow x) = \nu_0(\uparrow x) = v(x)$. □

6.5 A representation theorem

Using the results of the previous section, we are now going to characterise completely the normalised valuations on the domain of configurations on an event structure. We first note that not all normalised continuous valuations arise from global valuation. To overcome this problem we define the notion of an event structure with "invisible events". This notion allows us to state the representation theorem for continuous valuations.

6.5.1 Leaking valuations

Some continuous valuations "leak" probability. The simplest example of that is the event structure composed of one event $*$ only, and the continuous valuation defined as $\nu(\emptyset) = 0$, $\nu(\uparrow \bot) = 1$, $\nu(\uparrow\{*\}) = 1/2$. The above continuous valuation is not generated by any global valuation. This suggests that we should generalise the definition of global valuation, allowing the use of subprobability distributions. At a first sight it seems that it is enough to relax condition b) to the following

b') If C is a covering at x, then $\sum_{x' \in C} v(x') \leq v(x)$.

It turns out that this is not the right generalisation, as the following example shows. Consider the event structure $\mathcal{E}$ where $E = \{d, e\}$ with the flat ordering and no conflict. Define a "leaking global valuation" on $\mathcal{E}$ by,

- $v(\emptyset) = 1$
- $v(\{d\}) = v(\{e\}) = 1$
- $v(\{d, e\}) = 0$

The function v satisfy conditions a) and b'), but it cannot be extended to a continuous valuation on the domain of configurations. Suppose it did, and call such valuation ν. Then ν is modular so that $\nu(\uparrow\{a\} \cup \uparrow\{b\}) = \nu(\uparrow\{a\}) + \nu(\uparrow\{b\}) - \nu(\uparrow\{a, b\}) = 1 + 1 - 0 = 2$. This would contradicts monotonicity, as $\nu(\uparrow \bot) = 1$.

6.5.2 Invisible events

In fact, the leaking of probability can be attributed to an "invisible" event, as we are now going to show.

Definition 6.5.1. Consider a confusion free event structure $\mathcal{E} = \langle E, \leq, \# \rangle$. Let $\mathit{cell}(\mathcal{E})$ be the set of the cells of $\mathcal{E}$. For every $c \in \mathit{cell}(\mathcal{E})$ we consider a new "invisible" event ∂_c such that $\partial_c \notin E$ and if $c \neq c'$ then $\partial_c \neq \partial_{c'}$. Let $\partial = \{\partial_c \mid c \in \mathit{cell}(\mathcal{E})\}$. We define $\mathcal{E}_\partial$ to be $\langle E_\partial, \leq_\partial, \#_\partial \rangle$, where

- $E_\partial = E \cup \partial$;
- $\leq_\partial$ is $\leq$ extended by $e \leq_\partial \partial_c$ if for all $e' \in c$, $e \leq e'$;
- $\#_\partial$ is $\#$ extended by $e \#_\partial \partial_c$ if there exists $e' \in c$, $e' \leq e$.

Roughly speaking $\mathcal{E}_\partial$ is $\mathcal{E}$ where every cell contains an extra invisible event which does not enable anything. Invisible events, like kitchen paper, suck up all leaking probability, as the following theorem shows.

Definition 6.5.2. A pre-valuation on a confusion-free event structure $\mathcal{E}$ is a function $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0, 1]$ such that $v(\emptyset) = 1$.

Theorem 6.5.3. *Let $\mathcal{E}$ be the confusion-free event structure. Let v be a pre-valuation on $\mathcal{E}$. A necessary and sufficient condition for there to be a unique normalised continuous valuation ν on $\mathcal{L}(\mathcal{E})$ with $v(x) = \nu(\uparrow x)$, is that v can be extended to a global valuation v_∂ on $\mathcal{E}_\partial$.*

Proof: Sufficiency. During the proof of Theorem 6.4.1, we have used condition b) only to prove monotonicity of ν_0. In particular we only use it to prove that for every n-tuple of configurations (x_i) and for every configuration y, if $\widehat{x_1} \cup \ldots \cup \widehat{x_n} \subseteq \widehat{y}$, then $\nu_0\left(\widehat{x_1} \cup \ldots \cup \widehat{x_n}\right) \leq \nu_0(\widehat{y})$.

Note that this statement can be formulated without referring to open set. It is equivalent to saying that if $y \leq x_1, \ldots, x_n$ then

$$v(y) \geq \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) .$$

Since v_∂ is a global valuation on $\mathcal{E}_\partial$, we can prove the above statement for it. That is for every configuration $y, x_1, \ldots, x_n$, if $y \leq x_1, \ldots, x_n$ then

$$v_\partial(y) \geq \sum_{\emptyset \neq I \subseteq I_n} (-1)^{|I|-1} v_\partial \left(\bigvee_{i \in I} x_i \right) .$$

Now, restricting our attention to the configurations not involving the invisible events, we get the result for $\mathcal{E}$. The rest of the proof of Theorem 6.4.1 does not make use of property b) nor of contravariance. □

To prove necessity we proceed in steps using some accessory lemmas. Consider a normalised continuous valuation ν on $\mathcal{L}(\mathcal{E})$. If we define $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0,1]$ by $v(x) = \nu(\uparrow x)$, clearly we have $v(\emptyset) = 1$. We need to show that such function can be extended to a global valuation on $\mathcal{E}_\partial$. Note that if such extension exists, it is unique.

We need some notation. For a configuration x, let $c_1, \ldots, c_n$ be distinct cells accessible at x. Let $c_i = \{e_i^1, \ldots, e_i^{m_i}\}$. Consider a subset $J \subseteq I_n$ and consider the set X_J of functions $f : J \to \mathcal{N}$ such that $f(j) \in I_{m_j}$. Roughly speaking a function in X_J chooses an event for every cell whose index is in J. For every $f \in X_J$ we define x^f to be $x \cup \bigcup_{j \in J} \{e_j^{f(j)}\}$. Thus x^f is simply x augmented with the events chosen by f. Clearly x^f is a configuration. For every $J \subseteq I_n$, we define x^{∂_J} to be $x \cup \bigcup_{j \in J} \{\partial_{c_j}\}$. We have that $x^{\partial_J} \in \mathcal{L}_{fin}(\mathcal{E}_\partial)$. Moreover every configuration in $\mathcal{L}_{fin}(\mathcal{E}_\partial)$ has this form for some x, because the invisible events are always maximal. Notice that if $J \cap J' = \emptyset$ then, for $f \in X_J$ we can write both $x^{f \partial_{J'}}$ and $x^{\partial_{J'} f}$ and that they denote the same configuration of $\mathcal{L}_{fin}(\mathcal{E}_\partial)$, that is

$$x \cup \bigcup_{j \in J} \{e_j^{f(j)}\} \cup \bigcup_{j' \in J'} \{\partial_{c_{j'}}\} .$$

To extend v to configurations of the form x^{∂_J}, we start by observing what happens when J is a singleton. What is the value of $v_\partial(x \cup \{\partial_c\})$? Since we want this extension to be a global valuation, we must have $v_\partial(x \cup \{\partial_c\}) + \sum_{e \in c} v_\partial(x \cup \{e\}) = v_\partial(x)$. This implies $v_\partial(x \cup \{\partial_c\}) = v(x) - \sum_{e \in c} v(x \cup \{e\})$, which we take as definition. Generalising:

Lemma 6.5.4. *If v_∂ is a global valuation on $\mathcal{E}_\partial$, x a finite configuration of $\mathcal{E}$ and $c_1, \ldots, c_n$ distinct cells accessible at x, then*

$$v_\partial(x^{\partial_{I_n}}) = \sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v_\partial(x^f) \right)$$

Proof: By induction on n. □

This also can be taken as definition of the extension of v to $\mathcal{E}_\partial$. The next lemma shows that in order for this extension to be a global valuation, it is enough that $\sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v(x^f) \right)$ belong always to $[0,1]$.

Lemma 6.5.5. *Suppose v is a pre-valuation on $\mathcal{E}$. Then v_∂ as defined above is a global valuation on $\mathcal{E}_\partial$ if and only if:*

b") for every x finite configuration of $\mathcal{E}$ and $c_1, \ldots, c_n$ distinct cells accessible at x,

$$0 \leq \sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v_\partial(x^f) \right) \leq 1 \,.$$

Lemma 6.5.5 could be used to define "leaking global valuations".

Proof: Clearly $v_\partial : \mathcal{L}_{fin}(\mathcal{E}_\partial) \to [0,1]$ and $v_\partial(\emptyset) = 1$. It remains to show condition b).

Take a configuration $x^{\partial_J} \in \mathcal{L}_{fin}(\mathcal{E}_\partial)$ (where J could be empty, so that $x^{\partial_J} \in \mathcal{L}_{fin}(\mathcal{E})$). Say $J = I_n$. Consider a cell c_{n+1} accessible at $x^{\partial_{I_n}}$. We want that

$$v_\partial(x^{\partial_{I_n}}) = v_\partial(x^{\partial_{I_{n+1}}}) + \sum_{g \in X_{\{n+1\}}} v_\partial(x^{\partial_{I_n} g}) \,.$$

Notice that $x^{\partial_{I_n} g} = x^{g \partial_{I_n}}$. Using the definition of v_∂ we have to show that

$$\sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v(x^f) \right)$$

$$= \sum_{J \subseteq I_{n+1}} \left((-1)^{|J|} \sum_{f \in X_J} v(x^f) \right) + \sum_{g \in X_{\{n+1\}}} \sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v(x^{gf}) \right) .$$

Notice that

$$\sum_{g \in X_{\{n+1\}}} \sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v(x^{gf}) \right) = \sum_{n+1 \in J \subseteq I_{n+1}} \left((-1)^{|J|-1} \sum_{f \in X_J} v(x^f) \right) .$$

If we move this term to the left we have to show

$$\sum_{J \subseteq I_n} \left((-1)^{|J|} \sum_{f \in X_J} v_(x^f) \right) + \sum_{n+1 \in J \subseteq I_{n+1}} \left((-1)^{|J|} \sum_{f \in X_J} v_(x^f) \right)$$

$$= \sum_{J \subseteq I_{n+1}} \left((-1)^{|J|} \sum_{f \in X_J} v_(x^f) \right)$$

which is true as we can partition the subsets of I_{n+1} in two classes: the ones that are included in I_n and the ones containing $n+1$. □

To conclude the proof of Theorem 6.5.3 we have to show that a normalised continuous valuation restricts to a pre-valuation satisfying condition b"). This follows from modularity and monotonicity.

Let ν be a continuous valuation on $\mathcal{L}(\mathcal{E})$. Consider a finite configuration x of $\mathcal{E}$ and $c_1, \ldots, c_n$ distinct cells accessible at x. Then for every $i \in I_n$ and every $f_i \in X_{\{i\}}$ $x \leq x^{f_i}$ so that $\bigcup_{i \in I_n} \bigcup_{f_i \in X_{\{i\}}} \widehat{x^{f_i}} \subseteq \widehat{x}$. By monotonicity it must be $\nu\left(\bigcup_{i\in I_n}\bigcup_{f_i\in X_{\{i\}}}\widehat{x^{f_i}}\right)\leq\nu(\widehat{x})$. Modularity implies the inclusion-exclusion formula so that

$$\nu\left(\bigcup_{i\in I_n}\bigcup_{f_i\in X_{\{i\}}}\widehat{x^{f_i}}\right)=\sum_{\emptyset\neq J\subseteq I_n}\sum_{\emptyset\neq X\subseteq X_J}(-1)^{|J\times X|-1}v\left(\bigvee_{f\in X}x^f\right).$$

Notice that when $X\subseteq X_J$ is not a singleton, then $\bigvee_{f\in X}x^f=\top$ as two different f's choose at least two different events in one of the cells, making the configurations incompatible. Those terms can therefore be removed, leaving the expression

$$R:=\sum_{\emptyset\neq J\subseteq I_n}\sum_{f\in X_J}(-1)^{|J|-1}v\left(x^f\right)$$

Since it denotes the value of ν on some open set $R\geq 0$. By the observation above, we have that $R\leq v(x)$. Remember that for $J=\emptyset$, X_J contains only the empty function $f:\emptyset\to\mathbb{N}$ so that $x=x^f$. We have then

$$0\leq v(x)-\sum_{\emptyset\neq J\subseteq I_n}\sum_{f\in X_I}(-1)^{|I|-1}v\left(x^f\right)\leq 1\,.$$

Which, after some obvious changes becomes

$$0\leq\sum_{J\subseteq I_n}\left((-1)^{|J|}\sum_{f\in X_J}v\left(x^f\right)\right)\leq 1\,.$$

□

6.5.3 A representation theorem

Recall that we can characterise the domains arising as sets of configurations of a confusion free event structure. These are precisely the distributive concrete domains.

The results shown so far allow us to characterise completely the continuous valuations on a coherent dI-domains.

Definition 6.5.6. Let $\mathcal{E}$ be a confusion free event structure. A *generalised global valuation* on $\mathcal{E}$ is a pre-valuation on $\mathcal{E}$ that can be extended to a global valuation on $\mathcal{E}_\partial$.

The representation theorem is the following.

Theorem 6.5.7. *Let $v:\mathcal{L}_{fin}(\mathcal{E})\to[0,1]$ be a pre-valuation. Then there exists a normalised continuous valuation ν on $\mathcal{L}(\mathcal{E})$ satisfying $\nu(\uparrow x)=v(x)$ if and only if v is a generalised global valuation.*

In the next chapter we will also be able to characterise the normalised continuous valuations that corresponds to global valuations. They are precisely the maximal elements in $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$.

6.6 Categorical analysis

We might ask whether the correspondence between valuations on event structures and continuous valuations on the domains of configurations is natural, in the sense of Category Theory. The answer is yes as we will now show.

First, we have to make precise the terms of our question.

Definition 6.6.1. [Win82, WN95] Given two confusion-free event structures $\mathcal{E}, \mathcal{E}'$, a *morphism* $f : \mathcal{E} \to \mathcal{E}'$ is a partial function $f : E \to E'$ such that

- whenever $x \in \mathcal{L}(\mathcal{E})$ then $f(x) \in \mathcal{L}(\mathcal{E}')$
- for every $x \in \mathcal{L}(\mathcal{E})$, for all $e_1, e_2 \in x$ if $f(e_1), f(e_2)$ are both defined and $f(e_1) = f(e_2)$, then $e_1 = e_2$.

Such morphisms define a category **CFES**. The operator $\mathcal{L}$ extends to a functor **CFES** → **ALG** by $\mathcal{L}(f)(x) = f(x)$.

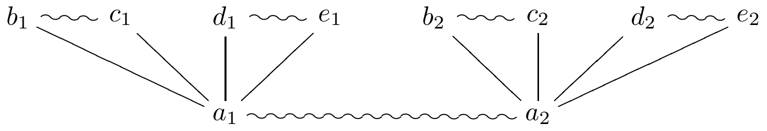
For every confusion free event structure $\mathcal{E}$, let $V(\mathcal{E})$ be the set of generalised global valuations on $\mathcal{E}$. We want to extend this operator to a functor on **CFES**. This is easily done, via continuous valuations.

$$\begin{tikzcd}
\mathcal{E} \arrow[d, "f"] & V(\mathcal{E}) \arrow[r, leftrightarrow] \arrow[d, dashed] & \mathcal{V}^1(\mathcal{L}(\mathcal{E})) \arrow[d, "\mathcal{V}^1(\mathcal{L}(f))"] \\
\mathcal{E}' & V(\mathcal{E}') \arrow[r, leftrightarrow] & \mathcal{V}^1(\mathcal{L}(\mathcal{E}'))
\end{tikzcd}$$

The use of morphisms allows us to make interesting observations.

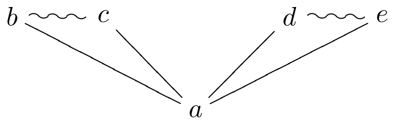
Consider the following event structures $\mathcal{E} = \langle E, \leq, \# \rangle, \mathcal{E}' = \langle E', \leq, \# \rangle$ where

- $E = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, e_1, e_2\}$;
- $a_1 \leq b_1, c_1, d_1, e_1,\ a_2 \leq b_2, c_2, d_2, e_2$;
- $a_1 \#_\mu a_2, b_1 \#_\mu c_1, b_2 \#_\mu c_2, d_1 \#_\mu e_1, d_2 \#_\mu e_2$;



and

- $E' = \{a, b, c, d, e\}$;
- $a \leq b, c, d, e$;
- $b \#_\mu c, d \#_\mu e$;



The map $f : E \to E'$ defined as $f(x_i) = x$, $x = a, b, c, d, e$ and $i = 1, 2$, is a morphism of event structures.

Now suppose we have a global valuation with independence v on $\mathcal{E}$. We can equivalently consider a local valuation p. Let ν be the corresponding continuous valuation on $\mathcal{L}(\mathcal{E})$. Let's now look at $v' := V(f)(v)$. First we observe

$$\begin{aligned} v'(\{a\}) = \mathcal{V}(f)(\nu)(\uparrow\{a\}) = \nu(\uparrow\{a_1\} \cup \uparrow\{a_2\}) \\ = \quad \nu(\uparrow\{a_1\} + \nu(\uparrow\{a_2\}) \\ = \quad v(\{a_1\}) + v(\{a_2\}) = p(a_1) + p(a_2) = 1\,. \end{aligned}$$

Then

$$\begin{aligned} v'(\{a, b\}) = \mathcal{V}(f)(\nu)(\uparrow\{a, b\}) = \nu(\uparrow\{a_1, b_1\} \cup \uparrow\{a_2, b_2\}) \\ = \quad \nu(\uparrow\{a_1, b_1\} + \nu(\uparrow\{a_2, b_2\}) \\ = \quad p(a_1) \cdot p(b_1) + p(a_2) \cdot p(b_2)\,. \end{aligned}$$

Suppose v' were also independent, then it would correspond to a local valuation p'. In that case $\mathcal{V}(f)(\nu)(\uparrow\{a, b\})$ must be equal to $v'(\{a, b\}) = p'(a) \cdot p'(b) = p'(b)$, so that $p'(b) = p(a_1) \cdot p(b_1) + p(a_2) \cdot p(b_2)$. Similarly $p'(d) = p(a_1) \cdot p(d_1) + p(a_2) \cdot p(d_2)$. Now on the one hand

$$\begin{aligned} \mathcal{V}(f)(\nu)(\uparrow\{a, b, d\}) = \nu(\uparrow\{a_1, b_1, d_1\} \cup \uparrow\{a_2, b_2, d_2\}) \\ = \quad \nu(\uparrow\{a_1, b_1, d_1\}) + \uparrow \nu(\{a_2, b_2, d_2\}) \\ = \quad p(a_1) \cdot p(b_1) \cdot p(d_1) + p(a_2) \cdot p(b_2) \cdot p(d_2)\,. \end{aligned}$$

On the other hand

$$\begin{aligned} \mathcal{V}(f)(\nu)(\uparrow\{a, b, d\}) = v'(\{a, b, d\}) \\ = \quad p'(a) \cdot p'(b) \cdot p'(d) = p'(b) \cdot p'(d) \\ = \quad [p(a_1) \cdot p(b_1) + p(a_2) \cdot p(b_2)] \cdot [p(a_1) \cdot p(d_1) + p(a_2) \cdot p(d_2)]\,. \end{aligned}$$

But in general it is not true that

$$\begin{aligned} [p(a_1) \cdot p(b_1) + p(a_2) \cdot p(b_2)] \cdot [p(a_1) \cdot p(d_1) + p(a_2) \cdot p(d_2)] \\ = p(a_1) \cdot p(b_1) \cdot p(d_1) + p(a_2) \cdot p(b_2) \cdot p(d_2)\,. \end{aligned}$$

The valuation v' is not with independence: in fact the correlation between the cell $\{b, c\}$ and the cell $\{d, e\}$ can be interpreted by saying that it is due to a hidden choice between a_1 and a_2. The question arises whether every global valuation is the projection of a global valuation with independence in a similar way. Presently, we do not know the answer.

The use of morphisms allows us also to relate the notion of conflict and the notion of probabilistic correlation. Consider the following event structures $\mathcal{E} = \langle E, \leq, \# \rangle, \mathcal{E}' = \langle E', \leq, \# \rangle$ where

- $E = \{a, b\}$, $a \#_\mu b$;
- $E' = \{a', b'\}$ with no conflict;

The map $f : E \to E'$ defined as $f(a) = a'$, $f(b) = b'$ is a morphism of event structures.

Consider the global valuation v on $\mathcal{E}$ defined as $v(\{a\}) = v(\{b\}) = 1/2$. The valuation $v' = V(f)(v)$ is as follows: $v'(\{a'\}) = v'(\{b'\}) = 1/2, v'(\{a', b'\}) = 0$. Although v is non-leaking, v' is. By theorem 6.5.3, we can extend v' to a non-leaking global valuation on $\mathcal{E}_\partial$:

$$\partial_{a'} \rightsquigarrow a' \qquad\qquad \partial_{b'} \rightsquigarrow b'$$

The (unique) extension is defined as follows:

- $v'(\{\partial_{a'}\}) = v'(\{\partial_{b'}\}) = v'(\{a'\}) = v'(\{b'\}) = 1/2$;
- $v'(\{\partial_{a'}, \partial_{b'}\}) = v'(\{a', b'\}) = 0$;
- $v'(\{\partial_{a'}, b'\}) = v'(\{a', \partial_{b'}\}) = 1/2$.

The conflict between a and b in $\mathcal{E}$ is seen in $\mathcal{E}'$ as a correlation between the cells of a' and b'. We cannot observe a' and b' together.

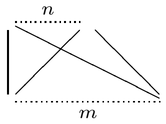
6 Appendix Two Combinatorial Lemmas

We prove here two lemmas used during the proof of Theorem 6.4.1.

Lemma 6.App.1. *For every finite sets I, J with $|I| = n, |J| = m$*

$$\sum_{\substack{\emptyset \neq K \subseteq I \times J \\ \pi_1(K) = I, \pi_2(K) = J}} (-1)^{|K|} = (-1)^{n+m-1}\,.$$

Proof: Without loss of generality we can think of $I = \{1, \ldots, n\}$ and $J = \{1, \ldots, m\}$. Also we observe that a subset $K \subseteq I \times J$ such that $\pi_1(K) = I, \pi_2(K) = J$ is in fact a surjective and total relation between the two sets.



Let

$$t_{n,m} := \sum_{\substack{\emptyset \neq K \subseteq I \times J \\ \pi_1(K) = I, \pi_2(K) = J}} (-1)^|K|\,;$$

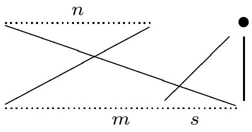
$$t^o_{n,m} := |\{\emptyset \neq K \subseteq I \times J \mid |K| \text{ odd}, \pi_1(K) = I, \pi_2(K) = J\}|\,;$$

$$t^e_{n,m} := |\{\emptyset \neq K \subseteq I \times J \mid |K| \text{ even}, \pi_1(K) = I, \pi_2(K) = J\}|\,.$$

Clearly $t_{n,m} = t^e_{n,m} - t^o_{n,m}$. We want to prove that $t_{n,m} = (-1)^{n+m+1}$. We do this by induction on n. It is easy to check that this is true for $n = 1$. In this case, if m is even then $t^e_{1,m} = 1$ and $t^o_{1,m} = 0$, so that $t^e_{1,m} - t^o_{1,m} = (-1)^{1+m+1}$. Similarly if m is odd.

Now let's assume that for every p, $t_{n,p} = (-1)^{n+p+1}$ and let's try to compute $t_{n+1,m}$. To evaluate $t_{n+1,m}$ we count all surjective and total relations K between I and J together with their"sign". Consider the pairs in K of the form $(n+1, h)$

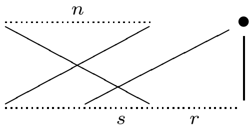
for $h \in J$. What do you get if you remove them? You get a total surjective relation between $\{1, \ldots, n\}$ and a subset J_K of $\{1, \ldots, m\}$.



Consider first the case where $J_K = \{1, \ldots, m\}$. What is the contribution of such K's to $t_{n+1,m}$? There are $\binom{m}{s}$ ways of choosing s pairs of the form $(n+1, h)$. And for every such choice there are $t_{n,m}$ (signed) relations. Adding the pairs $(n+1, h)$ possibly modifies the sign of such relations. All in all the contribution amounts to

$$\sum_{1 \leq s \leq m} \binom{m}{s} (-1)^s t_{n,m} \,.$$

Suppose now that J_K is a proper subset of $\{1, \ldots, m\}$ leaving out r elements.



Since K is surjective, all such elements h must be in a pair of the form $(n+1, h)$. Moreover there can be s pairs of the form $(n+1, h')$ with $h' \in J_K$. What is the contribution of such K's to $t_{n,m}$? There are $\binom{m}{r}$ ways of choosing the elements that are left out. For every such choice and for every s such that $0 \leq s \leq m - r$ there are $\binom{m-r}{s}$ ways of choosing the $h' \in J_K$. And for every such choice there are $t_{n,m-r}$ (signed) relations. Adding the pairs $(n+1, h)$ and $(n+1, h')$ possibly modifies the sign of such relations. All in all, for every r such that $1 \leq r \leq m - 1$, the contribution amounts to

$$\binom{m}{r} \sum_{1 \leq s \leq m-r} \binom{m}{s} (-1)^{s+r} t_{n,m-n} \,.$$

The (signed) sum of all these contribution will give us $t_{n+1,m}$. Now we use the induction hypothesis and we write $(-1)^{n+p+1}$ for $t_{n,p}$.

Thus:

$$\begin{aligned}
t_{n+1,m} &= \sum_{1 \leq s \leq m} \binom{m}{s} (-1)^s t_{n,m} \\
&\quad + \sum_{1 \leq r \leq m-1} \binom{m}{r} \sum_{0 \leq s \leq m-r} \binom{m-r}{s} (-1)^{s+r} t_{n,m-r} \\
&= \sum_{1 \leq s \leq m} \binom{m}{s} (-1)^{s+n+m+1} \\
&\quad + \sum_{1 \leq r \leq m-1} \binom{m}{r} \sum_{0 \leq s \leq m-r} \binom{m-r}{s} (-1)^{s+n+m+1}
\end{aligned}$$

$$
\begin{aligned}
= \quad & (-1)^{n+m+1} \left(\sum_{1 \le s \le m} \binom{m}{s} (-1)^s \right. \\
& \left. + \sum_{1 \le r \le m-1} \binom{m}{r} \sum_{0 \le s \le m-r} \binom{m-r}{s} (-1)^s \right) .
\end{aligned}
$$

By the binomial formula, for $1 \le r \le m-1$ we have

$$0 = (1-1)^{m-r} = \sum_{0 \le s \le m-r} \binom{m-r}{s} (-1)^s .$$

So we are left with

$$
\begin{aligned}
t_{n+1,m} &= (-1)^{n+m+1} \left(\sum_{1 \le s \le m} \binom{m}{s} (-1)^s \right) \\
&= (-1)^{n+m+1} \left(\sum_{0 \le s \le m} \binom{m}{s} (-1)^s - \binom{m}{0} (-1)^0 \right) \\
&= (-1)^{n+m+1} (0-1) \\
&= (-1)^{n+1+m+1} .
\end{aligned}
$$

Which is what we wanted to prove. □

Lemma 6.App.2 (BSV lemma). *Let X be a finite set and let $f : P(X) \to \mathbb{R}$. Then*

$$\sum_{\emptyset \ne J \subseteq X} (-1)^{|J|-1} f(J) = \sum_{\emptyset \ne K \subseteq X} \sum_{K \subseteq J \subseteq X} (-1)^{|J|+|K|} f(J) .$$

Proof: By induction on $|X|$. The base is obvious. Let $X' = X \cup \{*\}$, with $* \notin X$. Consider

$$\sum_{\emptyset \ne K \subseteq X'} \sum_{K \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J)$$

We can split the sum in two, according to whether K contains or does not contain $*$.

$$= \sum_{\emptyset \ne K \subseteq X} \sum_{K \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J) + \sum_{* \in K \subseteq X'} \sum_{K \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J)$$

We now rewrite the second part of the expression, singling out the case where $K = \{*\}$. In all the other cases we can write K as $H \cup \{*\}$ for some nonempty $H \subseteq X$.

$$
\begin{aligned}
&= \sum_{\emptyset \ne K \subseteq X} \sum_{K \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J) \\
&+ \sum_{\emptyset \ne H \subseteq X} \sum_{H \cup \{*\} \subseteq J \subseteq X'} (-1)^{|J|+|H|+1} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|+1} f(J)
\end{aligned}
$$

We now split each of the inner sums in the first line according to whether J contains or does not contain $*$. We have then

$$= \sum_{\emptyset \neq K \subseteq X} \left(\sum_{K \subseteq J \subseteq X} (-1)^{|J|+|K|} f(J) + \sum_{K \cup \{*\} \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J) \right)$$

$$+ \sum_{\emptyset \neq H \subseteq X} \sum_{H \cup \{*\} \subseteq J \subseteq X'} (-1)^{|J|+|H|+1} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|+1} f(J)$$

$$= \sum_{\emptyset \neq K \subseteq X} \sum_{K \subseteq J \subseteq X} (-1)^{|J|+|K|} f(J) + \sum_{\emptyset \neq K \subseteq X} \sum_{K \cup \{*\} \subseteq J \subseteq X'} (-1)^{|J|+|K|} f(J)$$

$$+ \sum_{\emptyset \neq H \subseteq X} \sum_{H \cup \{*\} \subseteq J \subseteq X'} (-1)^{|J|+|H|+1} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|+1} f(J)$$

Now the second and the third member of the expression above cancel out.

$$= \sum_{\emptyset \neq K \subseteq X} \sum_{K \subseteq J \subseteq X} (-1)^{|J|+|K|} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|+1} f(J)$$

We now use the induction hypothesis on the first member

$$= \sum_{\emptyset \neq J \subseteq X} (-1)^{|J|-1} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|+1} f(J)$$

$$= \sum_{\emptyset \neq J \subseteq X} (-1)^{|J|-1} f(J) + \sum_{* \in J \subseteq X'} (-1)^{|J|-1} f(J)$$

Which can be finally joined.

$$= \sum_{\emptyset \neq J \subseteq X'} (-1)^{|J|-1} f(J)\,.$$

□

Chapter 7

Probabilistic Runs of Event Structures

In this chapter we define different notions of run of a probabilistic event structure. The leading idea is that a run of a probabilistic event structure be a probability distribution over nonprobabilistic runs. We define the notion of *test* as a set of configurations which is suitable to represent a probabilistic run. Using tests we can give an alternative characterisation of global valuations. The use of tests allows us also to characterise the continuous valuations arising from global valuations as the maximal elements in $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$.

Finally we prove a confluence theorem showing that in some sense, probabilistic event structures do not feature nondeterminism.

7.1 Event structures as Markov decision processes

We first introduce a notion of "sequential" run for a probabilistic event structure. We do that by seeing an event structure as a Markov decision process. Configurations are the states. At every state, the actions are the accessible cells. A scheduler is a function choosing, at every state, one of the accessible cells.

Instead of following the general framework devised in Section 2.6, we choose to present here an alternative semantics, using ideas from [dAHJ01].

7.1.1 Probabilistic words

Let S be an set. In the following elements of S are called *symbols*, elements of S^* are called *strings*.

Definition 7.1.1. A *(probabilistic) letter* over S is a probability distribution over symbols, i.e. a function $\alpha : S \to [0,1]$ such that $\alpha[S] = 1$. A *(probabilistic) word* of length n is a probability distribution β over strings of length n.

We find it useful to introduce the following notation. We denote a letter α also as $(\alpha(a))_{a\in W}$ where $W = \mathit{Supp}(\alpha)$. We denote a word β of length n also as $(\beta(\sigma))_{\sigma\in W}$ where $W = \mathit{Supp}(\beta)$. We identify letters with words having only strings of length 1 in the support. To ease the reading we will write $(\beta(\sigma))_W$, or $(\beta)_W$ for $(\beta(\sigma))_{\sigma \in W}$, when this does not create confusion.

This notation allows us to define a notion of composition between words.

Definition 7.1.2. Let $(\beta(\sigma))_W$ be a word, let $(\gamma_\sigma(\tau))^W_{Z_\sigma}$ be a family of words of the same length indexed by W. The composition of these two objects, denoted by $\beta; (\gamma_\sigma)^W$ is the word ζ, such that $\zeta(\sigma; \tau) = \beta(\sigma)\gamma_\sigma(\tau)$, where by $\sigma; \tau$ we mean the usual composition of strings.

It is easy to check that the operation is well defined. If $\beta, (\gamma_\sigma)^W$ are words, then $\beta; (\gamma_\sigma)^W$ is a word. We write $\beta; \gamma$ to mean $\beta; (\gamma_\sigma)^W$ when for every $\sigma \in W$, $\gamma_\sigma = \gamma$. The neutral element on the left is the word assigning probability 1 to the string of length 0. We call this word ϵ, as the only string in its support. The neutral elements on the right are the families with all elements equal to ϵ.

Definition 7.1.3. A word β is *prefix* of a word β' if there exists a family $(\gamma_\sigma)^W$ such that $\beta; (\gamma_\sigma)^W = \beta'$. We denote this by $\beta \preceq \beta'$. A *language* is a set of words. A language L is *prefix-closed* iff $\beta' \in L$ and $\beta \preceq \beta'$ implies that $\beta \in L$.

Proposition 7.1.4. *Let β, β' be two probabilistic words over S. We have $\beta \preceq \beta'$ if and only if for every $\sigma \in Supp(\beta)$*

$$\beta(\sigma) = \sum_{\sigma \leq \sigma'} \beta'(\sigma')\,.$$

7.1.2 Schedulers

For simplicity of exposition we will consider only event structures such that at every configuration there is at least one accessible cell. This restriction can be easily avoided by adding a countable chain of independent "idling" events.

Definition 7.1.5. A *fragment scheduler* of a confusion free event structure $\langle E, \leq, \# \rangle$ is a partial function $f : \mathit{Str}(\mathcal{E}) \to \mathit{cell}(\mathcal{E})$ such that $f(\sigma)$ is accessible at $\mathit{Conf}(\sigma)$. A *finite scheduler* f of *length* n is a fragment scheduler defined on words of length $l < n$. The length of f is denoted by $|f|$. A fragment scheduler f' *extends* a finite scheduler f, if $\mathit{Dom}(f) \cap \mathit{Dom}(f') = \emptyset$ and $f \cup f'$ is a finite scheduler. In that case we write $f; f'$ for $f \cup f'$.

Using schedulers, we can give semantics to probabilistic event structures with independence in terms of probabilistic words.

Definition 7.1.6. The *word* of a probabilistic event structure with independence $\langle \mathcal{E}, p \rangle$ *under the scheduler* f, denoted by $b(f)$, is the probabilistic word defined by induction on the length as follows: If $|f| = 0$,

$$b(f)(\sigma) = \begin{cases} 1 & \text{if } \sigma = \epsilon \\ 0 & \text{otherwise}\,. \end{cases}$$

If $|f| = n + 1$ we can write $f = f'; g$ with $|f'| = n$ and with g defined only over words of length n. Consider a string σ. If $|\sigma| \neq n + 1$ we put $b(f)(\sigma) = 0$. If $|\sigma| = n + 1$, say $\sigma = \sigma'; e$ we put

$$b(f)(\sigma'; e) = \begin{cases} b(f')(\sigma') \cdot p(e) & \text{if } e \in g(\sigma) \\ 0 & \text{if } e \notin g(\sigma)\,. \end{cases}$$

The probability of a sequence $\sigma; e$, in $B(f)$, is the probability of σ in the bundle generated by f' times the probability of the last event e, under the condition that e belongs to the cell chosen by f. It is easy to show that $b(f)$ is indeed a probabilistic word. Moreover if $g = f; f'$, then $w(f) \preceq w(g)$.

Definition 7.1.7. The *language* $\mathcal{B}(\mathcal{E})$ of a probabilistic event structure $\mathcal{E}$ is the set of its words.

7.2 Tests

The semantics of the previous section models parallelism by interleaving. We are now going to define a different notion of run, which makes use of configurations. Consequently this notion is more sensitive to the causal nature of event structures, and it does not use interleaving. Moreover it can be given to general probabilistic event structures, without the independence requirement.

7.2.1 Definition

We want the run of an event structure to be a probability distribution over configurations. Which sets of configurations are suitable? Let's look at the interleaving semantics first. A probabilistic word is a probability distribution over strings of the same length. Therefore any two strings in the support are incomparable. This is the first condition we require to our notion of run.

Definition 7.2.1. A *partial test* C of an event structure $\mathcal{E}$ is a set of configurations of $\mathcal{E}$ such that for every x, x' in C, x and x' are incompatible.

Clearly this is not enough. In order to represent a probabilistic run, a set C of configuration must have the property that $\sum_{x \in C} v(x) = 1$. A singleton is a partial test, but it does not, in general, have the above property. It turns out that we need to require *completeness*. In the sequel we will be able to justify this requirement. For the moment we ask the reader to accept the following definition.

Definition 7.2.2. A set C of configurations of an event structure $\mathcal{E}$ is *complete* if for every $y \in \mathcal{L}(\mathcal{E})$ there exists $x \in C$ such that x, y are compatible.

Proposition 7.2.3. *Let C be set of configurations of an event structure $\mathcal{E}$. The set C is complete if and only if for every maximal configuration $z \in \mathcal{L}(\mathcal{E})$, there exists $x \in C$ such that $x \leq z$.*

Proof: $\Longrightarrow$) Take a maximal configuration z. By completeness there exists $x \in C$ such that x, z are compatible. This means that $x \cup z$ is a configuration. But since z is maximal, $x \cup z = z$, that is $x \leq z$.

$\Longleftarrow$) Take any configuration y. Take a maximal configuration z such that $y \leq z$ (it exists by Zorn's Lemma – we could also build it directly by induction using just dependent choice). By hypothesis there exists $x \in C$ such that $x \leq z$. Since z is a common upper bound for x, y, then x, y are compatible. □

Definition 7.2.4. A *test* C of an event structure $\mathcal{E}$ is a complete partial test.

For a partial test C, completeness amounts to saying that no configuration x can be added to C, so that $C \cup \{x\}$ is still a partial test.

Definition 7.2.5. A set C of configurations of an event structure is *finitary* is all its members are finite.

Since we will use tests to represent runs, we also need a notion of *extension* of a run. For this aim, the Egli-Milner ordering is the appropriate one.

Definition 7.2.6. Let C, C' be sets of configurations of an event structure. We say that $C \leq C'$ if for every $x \in C$ there exists $x' \in C'$ with $x \leq x'$ and for every $x' \in C'$ there exists $x \in C$ with $x \leq x'$.

7.2.2 Tests as probabilistic runs

When we endow the event structure with a valuation, tests represent probabilistic runs of the event structure. The partial order relation on tests represents the extension of a probabilistic run. Indeed

Theorem 7.2.7. *If v is a global valuation, and if C is a finitary test, then,*

$$\sum_{x \in C} v(x) = 1\,.$$

It is possible to give a simple proof of this fact for valuations with independence, using elementary measure theory. However, in order to prove the theorem in full generality, we need to develop some tools. We will make use of theorem 6.4.1, relating valuations on event structures and on domains. The first results interpret the above notions within the domain of configurations.

Definition 7.2.8. Let C be a finitary set of configurations of an event structure $\mathcal{E}$. We define $\uparrow (C)$ as the set $\bigcup_{x \in C} \uparrow x$.

Clearly $\uparrow (C)$ is Scott open. All the following properties are straightforward.

Proposition 7.2.9. *Let C be a finitary partial test of $\mathcal{E}$, then the Scott open subsets of $\mathcal{L}(\mathcal{E})$ of the form $\uparrow x$, for $x \in C$ are pairwise disjoint. If C, C' are two finitary sets of configurations of $\mathcal{E}$ and $C \leq C'$ then $\uparrow (C) \supseteq \uparrow (C')$. If C be a finitary complete set of configurations of $\mathcal{E}$, then for every maximal configuration $y \in \mathcal{L}(\mathcal{E})$, we have that $y \in \uparrow (C)$.*

Less trivially,

Proposition 7.2.10. *Let C, C' be finitary tests. Then $C \leq C'$ if and only if $\uparrow (C) \supseteq \uparrow (C')$.*

Proof: of the non-trivial direction. Suppose $\uparrow (C) \supseteq \uparrow (C')$. If $y \in C'$ then $y \in \uparrow (C)$ which means that there exists $x \in C$ such that $x \leq y$. Vice versa if $x \in C$ then by completeness there exists $y \in C'$ such that x, y are compatible. We have just argued that there exists $x' \in C$ such that $x' \leq y$, which implies that x, x' are compatible. Since C is a test, we have that $x = x'$ and $x \leq y$. □

Corollary 7.2.11. *Let ν be a continuous valuation on $\mathcal{L}(\mathcal{E})$. If C is a finitary partial test, then $\nu(\uparrow (C)) = \sum_{x \in C} \nu(\uparrow x)$. If C, C' are finitary sets of configurations and $C \leq C'$ then $\nu(\uparrow (C)) \geq \nu(\uparrow (C'))$.*

7.3 Inductive tests

We introduce a restricted notion of test, more closely related to the interleaving semantics and the theory of probabilistic languages. They will also have a technical role in the proof of Theorem 7.2.7.

7.3.1 Inductive tests and probabilistic words

Definition 7.3.1. Let $\mathcal{E}$ be a confusion-free event structure. If x is a configuration of $\mathcal{E}$, and c is a cell accessible at x we define $x + c$ do be the set $\{x \cup \{e\} \mid e \in c\}$. Let Y, Y' be two sets of configurations of a confusion-free event structure. We write

$$Y \xrightarrow{X,(c_x)} Y'$$

when $X \subseteq Y$, for every $x \in X$, c_x is a cell accessible at x, and

$$Y' = Y \setminus X \cup \bigcup_{x \in X} x + c_x \,.$$

We write $Y \to Y'$ if there are $X, (c_x)$ such that $Y \xrightarrow{X,(c_x)} Y'$. As usual $\to^*$ denotes the reflexive and transitive closure of $\to$.

Definition 7.3.2. An *inductive test* of a confusion-free event structure is a set C of configurations such that

$$\{\emptyset\} \to^* C \,.$$

The idea is that we start the computation with the empty configuration, and, at every step, we choose accessible cells to "activate" and we collect all the resulting configurations. It is easy to see a connection between probabilistic words and inductive tests. Remember we define the semantics in terms of words only for probabilistic event structure with independence.

Proposition 7.3.3. *Let $\langle \mathcal{E}, p\rangle$ be a probabilistic event structure with independence. For every word $\beta \in \mathcal{B}(\mathcal{E})$, $Conf(Supp(\beta))$ is an inductive test of $\mathcal{E}$ and for every $\sigma \in Supp(\beta)$, $\beta(\sigma) = v_p(Conf(\sigma))$. If $\beta \preceq \gamma$ then $Conf(Supp(\beta)) \leq Conf(Supp(\gamma))$.*

Proof: By induction on the length of β: the only word of length 0 corresponds to the test $\{\emptyset\}$. Take a scheduler f of length n and consider the scheduler $f;\chi$ of length $n+1$. Let $X := Supp(b(f))$. For every $\sigma \in X$, $\chi(\sigma)$ is an action at σ, that is a cell accessible at $Conf(\sigma)$. Then

$$Conf(X) \xrightarrow{X,\chi(\sigma)} Conf(Supp(\gamma)) \,.$$

As for the probabilities, clearly $\beta(\sigma)$ and $v_p(Conf(\sigma))$ always coincide, being just the product of the probabilities of the constituting events. In order to prove the last statement, first note that $\beta \preceq \gamma$ implies $Conf(Supp(\beta)) \to^* Conf(Supp(\gamma))$. As a consequence of the forthcoming Proposition 7.3.4, we have $Conf(Supp(\beta)) \leq Conf(Supp(\gamma))$. □

The next proposition is a sanity check for our definitions

Proposition 7.3.4. *If C, C' are inductive tests*

$$C \leq C' \iff C \to^* C' .$$

The direction $\Longleftarrow$) is proved by induction on the derivation $C \to^* C'$. The direction $\Longrightarrow$) is by induction on the derivation $\{\emptyset\} \to^* C$ and is postponed to the Appendix.

7.3.2 Inductive tests are tests

As the choice of the name suggests we have the following result.

Proposition 7.3.5. *Every inductive test is a finitary test.*

Proof: By induction on the derivations. The singleton of the empty configuration is a test. Take an inductive test C, a set $X \subseteq C$ and for every $x \in X$ a cell (c_x) accessible at x. Let $C \xrightarrow{X,(c_x)} C'$. We want to show that C' is a test.

First consider two distinct configurations $x', y' \in C'$. If $x', y' \in C$ then they are incompatible by induction hypothesis. If $x' \in C$, and $y' = y \cup e$ for some $y \in C$, then $x' \neq y$, so that x', y are incompatible. Thus x', y' are incompatible. If $x' = x \cup e_x$ and $y' = y \cup e_y$ for $x, y \in C$ there are two possibilities. If $x \neq y$, then they are incompatible and so are x', y'. If $x = y$, then $e_x \neq e_y$, but they both belong to they same cell, therefore they are in conflict, and x', y' are incompatible.

Now take any configuration z. By induction hypothesis there exists $x \in C$ such that x, z are compatible. If $x \in C'$ we are done. If $x \notin C'$ then there are two possibilities. Either z does not fill c_x, but then for every $e \in c_x$, $z, x \cup e$ are compatible. Or z fills c_x with and event $\bar{e}$ which implies that $z, x \cup \bar{e}$ are compatible. □

Not all test are inductive as the following example shows. Consider the event structure $\mathcal{E} = \langle E, \leq, \# \rangle$ where $E = \{a_1, a_2, b_1, b_2, c_1, c_2\}$, the order is trivial and $a_1 \# a_2, b_1 \# b_2, c_1 \# c_2$. Let's call the three cells a, b, c.

$$a_1 \sim\!\sim\!\sim a_2 \qquad b_1 \sim\!\sim\!\sim b_2 \qquad c_1 \sim\!\sim\!\sim c_2$$

Consider the following set C of configurations

$$\big\{\{a_1, b_2\}, \{b_1, c_2\}, \{a_2, c_1\}, \{a_1, b_1, c_1\}, \{a_2, b_2, c_2\}\big\} .$$

The reader can easily verify that C is a test. If it were an inductive test, we should be able to identify a cell that was chosen at the first step along the derivation. Because of the symmetry of the situation, we can check whether it is a. If a were the first cell chosen, every configuration in C would contain either a_1 or a_2. But this is not the case.

It is now easy to show the following

Proposition 7.3.6. *If v is a global valuation, and if C is an inductive test, then,*

$$\sum_{x \in C} v(x) = 1 .$$

Proof: By induction on the derivation

Suppose $C \xrightarrow{X,c_x} C'$ and $\sum_{x\in C} v(x) = 1$. Consider $\sum_{x'\in C'} v(x')$. We can split this in

$$\sum_{x\in C\setminus X} v(x) + \sum_{x\in X}\sum_{e\in c_x} v(x\cup\{e\}) \, .$$

Since v is a global valuation, property (b) tells us that for every $x \in X$, $\sum_{e\in c_x} v(x\cup\{e\}) = v(x)$. Therefore

$$\sum_{x\in C\setminus X} v(x) + \sum_{x\in X}\sum_{e\in c_x} v(x\cup\{e\})$$

$$= \sum_{x\in C\setminus X} v(x) + \sum_{x\in X} v(x) = \sum_{x\in C} v(x) = 1 \, .$$

□

Alternatively we could obtain Proposition 7.3.6 as a corollary of Proposition 7.3.3.

7.4 Proof of theorem 7.2.7

We recall the statement of the theorem: If v is a global valuation, and if C is a finitary test, then,

$$\sum_{x\in C} v(x) = 1 \, .$$

We show that there exists an enumeration of the cells $(c_n)_{n\in\mathbb{N}}$, such that if $c_m < c_n$, then $m < n$. We build it as follows. Since the cells are countably many, they come equipped already with some enumeration. We start by picking the first cell c. We enumerate all the cells $c' < c$, by layers: first the cells of depth 0, then the cells of depth 1 and so on. There are only finitely many such c', so we stop at some point. Finally we enumerate c. For all the cells enumerated so far $c_m < c_n$ implies $m < n$

At every step, choose the next cell c (in the old enumeration) that has not been enumerated. Repeat the procedure above, enumerating the cells $c' < c$ that have not yet been enumerated. Finally enumerate c. The invariant $c_m < c_n \Longrightarrow m < n$ is preserved.

With this enumeration at hand, consider the following chain of inductive tests: $C_0 = \{\emptyset\}$, $C_n \xrightarrow{X,c_n} C_{n+1}$, where X is the set of configurations $x \in C_n$ such that c_n is accessible at x. We have the following properties:

1. for every C_n, $maxm(\mathcal{L}(\mathcal{E})) \subseteq\uparrow (C_n)$;

2. $\uparrow (C_n) \supseteq\uparrow (C_{n+1})$;

3. if $x \in C_n$ and x fills c_m then $m < n$;

4. if $x \in C_n$ then every cell c_m with $m < n$ enabled at x is filled by x;

5. for every non maximal configuration z there exists n such that $z \not\in\uparrow (C_n)$.

Property (1) comes for the fact the C_n is a test. Property (2) comes from Proposition 7.2.10. Property (3) is by construction. Property (4) is shown by induction on n, using the defining property of the enumeration. Take $x \in C_{n+1}$ and consider a cell c_m with $m < n+1$ enabled at x. If $m < n$ then $c_n \not< c_m$ therefore c_m is enabled at $x' := x \setminus c_n \in C_n$. By induction hypothesis c_m is filled by x', and therefore is filled by x. If $m = n$ then x has just been obtained by adding an event in c_m (otherwise c_m would not be enabled). To show (5), take a non maximal configuration z. There exists a cell c which is accessible at z. Suppose it's c_m. Consider C_{m+1}. Suppose there exists $x \in C_{m+1}$ such that $x \leq z$. Then c_m is not filled by x. By property (4), c is not enabled at x. Consider a minimal event e in $[c) \setminus x$, and say $c_h = \mathit{cell}(e)$. Since $c_h < c = c_m$, then $h < m$. By minimality of e, every event in $[c_h)$ is in x. Therefore c_h is enabled at x. By property (4) c_h is filled by x. Since $[c) \subseteq z$ we have that $e \in z$. Thus the only event in the cell of e that can be in x is e itself. Contradiction.

Therefore, combining (1) and (5)

$$\bigcap_{n \in \mathbb{N}} \uparrow (C_n) = \mathit{maxm}(\mathcal{L}(\mathcal{E}))\ ,.$$

By Theorem 2.5.18, the valuation ν extends to a Borel measure $\bar{\nu}$. We have that $\bar{\nu}(\mathit{maxm}(\mathcal{L}(\mathcal{E}))) = \lim_{n\to\infty} \bar{\nu}(\uparrow (C_n))$. But $\bar{\nu}(\uparrow (C_n)) = \nu(\uparrow (C_n)) = 1$ because C_n is an inductive test. By Theorem 2.5.17 we have $\bar{\nu}(\mathit{maxm}(\mathcal{L}(\mathcal{E}))) = 1$. This implies that for every finitary test C

$$1 \geq \nu(\uparrow (C)) = \bar{\nu}(\uparrow (C)) \geq \bar{\nu}(\mathit{maxm}(\mathcal{L}(\mathcal{E}))) = 1$$

which finally implies that $\sum_{x \in C} v(x) = 1$. □

We can characterise global valuations using tests, by inverting theorem 7.2.7.

Theorem 7.4.1. *Let $\mathcal{E}$ be a confusion-free event structure. Let v be a function $\mathcal{L}_{fin}(\mathcal{E}) \to [0,1]$. Then v is a global valuation if and only if for every finitary test C, $v[C] = 1$.*

Proof: First of all $v(\emptyset) = 1$, because $\{\emptyset\}$ is a finitary test. Next we want to show that for every finite configuration x and every covering D_c at x, $v[D_c] = v(x)$. Take a test C containing x. It is not difficult to build an inductive such, by firing in sequence all the cells filled by x. Consider the test $C' = C \setminus \{x\} \cup D_c$. Notice that $C \xrightarrow{\{x\},c} C'$. Therefore C' is a test. So that $v[C'] = 1$. But $v[C'] = v[C] - v(x) + v[D_c]$. □

7.5 Confluence

The interleaving semantics introduces some nondeterminism in the choice of the cell to fire. Intuitively this choice in inessential, in that it only determines the order of concurrent events. The causal semantics allows us to formalise this intuition, in terms of a confluence property: every two runs are part of a longer run. There is no real nondeterministic branching.

7.5.1 Confluence of tests

Theorem 7.5.1. *Let E be a probabilistic event structure. For every C, C' tests of E there exists a test C'' with $C, C' \leq C''$.*

Proof: Take two tests C, C' and consider the set

$$C'' := \big\{x \cup y \mid x \in C, y \in C',\ x, y \text{ compatible}\big\} .$$

We claim that C'' is a test. First consider two different elements of C'': $x \cup y$ and $x' \cup y'$. We can assume that $x \neq x'$. Then, since C is a test, we have that x, x' are incompatible. This means that there is a cell c and two different events $e, e' \in c$ such that $e \in x$ and $e' \in x'$. This implies that $x \cup y$ and $x' \cup y'$ are also incompatible. As for maximality, consider any configuration z. Since C is a test there exists $x \in C$ such that z, x are compatible. Then $z \cup x$ is a configuration. Since C' is a test, there exists $y \in C'$ such that $x \cup z$ and y are compatible. This implies that y and x are compatible. Therefore $x \cup y \in C''$. Now, since both x and y are compatible with z, we have that $x \cup y$ is compatible with z.

To show that $C \leq C''$, take $x \in C$. Since C' is a test there exists $y \in C'$ such that x, y are compatible, so that $x \cup y \in C''$ and clearly $x \subseteq x \cup y$. The other direction of the definition is obvious. □

The theorem above is carries over to inductive tests as

Proposition 7.5.2. *If C, C' are inductive tests of E, then $C'' := \big\{x \cup y \mid x \in C, y \in C',\ x, y$ compatible$\big\}$ is an inductive test.*

By induction on the derivation of C, C'. When they both are the singletons of the empty configuration, the statement is true. Suppose $C_0 \xrightarrow{Z,(c_z)} C$. Let $C''_0 := \big\{z \cup y \mid z \in C_0, y \in C',\ z, y \text{ compatible}\big\}$. Cy induction hypothesis, it is an inductive test. For every $z \in Z$ consider the configurations of the form $z \cup y \in C''_0$. Let Y_z be the set of configurations $y \in C'$ compatible with z and not filling the cell c_z. Then $y \in Y_z$ if and only if c_z is accessible at $z \cup y$.

Let $W = \bigcup_{z \in Z, y \in Y_z} \{z \cup y\}$, by the previous observation we have that, for some D,

$$C''_0 \xrightarrow{W,(c_z)} D .$$

Claim: $D = C''$.

Take $w \in D$. If $w \in C''_0$, then $w = z \cup y$ for some $z \in C_0, y \in C'$, and such that, if $z \in Z$, then $y \notin Y_z$. If $z \notin Z$, then $z \in C$ and so $z \in C''$. If $z \in Z$ and $y \notin Y_z$, then y fills c_z, so that for exactly one event $e \in c_z$, $z \cup \{e\}$ is compatible with y. In that case $z \cup \{e\} \cup y = z \cup y = w$, so that again $z \in C''$.

If $w \notin C''_0$, then there exist $z \in Z, y \in Y_z, e \in c_z$ such that $w = z \cup y \cup \{e\}$. Cut then $z \cup \{e\} \in C$ so that again $w \in C''$.

Conversely, suppose $w \in C''$, then $w = x \cup y$ for some $x \in C, y \in C'$. If $x \in C_0$, this means that $x \in Z$, so that $x \cup y \in D$. (Notice that if $z \neq z'$, then for all $y' \in C'$, $z \cup y \neq z' \cup y'$.) If $x \notin C_0$, this means that there is $z \in Z$, $e \in c_z$, such that $x = z \cup \{e\}$. If $y \in Y_z$, then $x \cup y = z \cup \{e\} \cup y$ and $x \cup y \in D$. If $y \notin Y_z$, then y fills c_z, and since x, y are compatible, it must be that $e \in y$. Therefore $x \cup y = z \cup y$, and again $x \cup y \in D$. □

7.5.2 Confluence of words

To interpret the confluence property in the context of probabilistic words we need an extension of the notion of Mazurkiewicz equivalence. We are now going to extend the notion of Mazurkiewicz equivalence to probabilistic words. Let $\bowtie$ be an irreflexive and symmetric relation on S. The pair $(S, \bowtie)$ is called a *concurrent alphabet.*

Definition 7.5.3. [Maz86] Let $(S, \bowtie)$ be a concurrent alphabet. We define the *Mazurkiewicz equivalence* $\equiv$ to be the least congruence on the monoid of strings S^* such that

$$a \bowtie b \Longrightarrow ab \equiv ba\,.$$

If $\langle E, \leq, \# \rangle$ is a confusion free event structure, we define an irreflexive and symmetric relation on E, by

$$eIe' \Longleftrightarrow e \not\leq e' \;\&\; e' \not\leq e \;\&\; \neg e \# e'\,.$$

One of the fundamental results we need is the following, which is an adaptation of a theorem in [Maz86].

Proposition 7.5.4. *Let σ, σ' be two strings of $\mathcal{E}$, then $\sigma \equiv \sigma'$ if and only if $Conf(\sigma) = Conf(\sigma')$.*

We propose two possible ways of extending Mazurkiewicz equivalence to probabilistic words. The first definition is a standard extension of a relation to a probabilistic framework. However, we also need a stronger notion of equivalence to carry out the proof of our main result.

Definition 7.5.5. Let $\beta = (\beta(\sigma))_W$ and $\gamma = (\gamma(\tau))_Z$ be two words of the same length over a concurrent alphabet $(S, \bowtie)$. We say that they are *Mazurkiewicz equivalent* and write $\beta \equiv_p \gamma$, if for every $\equiv$-equivalence class C, $\beta[C] = \gamma[C]$. We say that they are *strongly Mazurkiewicz equivalent* and write $\beta \equiv_s \gamma$, if there exists a bijection $\phi : W \to Z$ such that for every $\sigma \in W$,

- $\sigma \equiv \phi(\sigma)$;
- $\beta(\sigma) = \gamma(\phi(\sigma))$.

We call such ϕ a *witness* of the equivalence. Clearly $\equiv_s \subseteq \equiv_p$, but not vice versa.

Theorem 7.5.6. *For all schedulers f, g there exist two schedulers f', g' such that $b(f) \preceq b(f')$, $b(g) \preceq b(g')$ and $b(f') \equiv_p b(g')$.*

We observe that proposition 7.3.3 can be inverted. In the sequel, when f is a scheduler, we will write $Conf(f)$ to denote $Conf(Supp(b(f)))$.

Lemma 7.5.7. *Let $b(f) \in \mathcal{B}(\mathcal{E})$, let C be an inductive test such that $Conf(f) \leq C$. Then there exists a partial scheduler f' such that $Conf(f; f') = C$.*

Corollary 7.5.8. *For every inductive test C there exist a probabilistic word $b(f) \in \mathcal{B}(\mathcal{E})$ such that $Conf(f) = C$.*

Proof: By induction on the derivation of C.
Finally:

Proposition 7.5.9. *If f, g are schedulers of the same length such that $Conf(f) = Conf(g)$ then $b(f) \equiv_s b(g)$.*

Now we can prove Theorem 7.5.6.

Take f, g consider $Conf(f), Conf(g)$. By theorem 7.5.1 and Proposition 7.5.2 there exists an inductive test C such that $Conf(f), Conf(g) \leq C$, by lemma 7.5.7 there exist f', g' such that $Conf(f : f') = Conf(g; g') = C$. By the proposition 7.5.9 $f; f' \equiv_s g; g'$.

7.6 More on non-leaking valuations

We are now going to study in more detail the continuous valuations arising from global valuations.

Definition 7.6.1. A normalised continuous valuation ν on a DCPO D is *non-leaking* if for every open set O such that O contains all maximal elements of D, we have that $\nu(O) = 1$

Theorem 7.6.2. *There is a bijection between global non-leaking valuations on an event structure $\mathcal{E}$ and non-leaking continuous valuations on $\mathcal{L}(\mathcal{E})$.*

Proof: We have seen that a continuous valuation generated by a global valuation is non-leaking, because it is supported on the set of maximal configurations. Vice versa, suppose we have a non-leaking continuous valuation ν on $\mathcal{L}(\mathcal{E})$. Define a pre-valuation on $\mathcal{E}$ by $v(x) = \nu(\uparrow x)$. If C is a finitary test, we have that $v[C] = \nu(\uparrow (C)) = 1$. By theorem 7.4.1, v is a global valuation on $\mathcal{E}$. □

For the domain of configurations of an event structure, we are able to characterise non-leaking valuations as maximal valuations. We divide this into two steps.

Proposition 7.6.3. *Let $\mathcal{E}$ be a confusion free event structure. A non-leaking valuation on $\mathcal{L}(\mathcal{E})$ is maximal in $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$.*

Proof: Take a non-leaking valuation ν on $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$. It corresponds to a global valuation v on $\mathcal{E}$. Suppose ν is not maximal and consider a normalised valuation ξ such that $\nu < \xi$. Then this must be witnessed on a principal open set (if they coincided on principal open sets they would coincide everywhere). Note that $\nu < \xi$ implies that ξ is also non-leaking, and thus generated by a global valuation w. Let x be a minimal finite configuration for which $\nu(\uparrow x) < \xi(\uparrow x)$, that is $v(x) < w(x)$. Consider a maximal element e of x. Let $x' := x \setminus e$. By minimality $v(x') = w(x')$. If c is the cell of e we have $\sum_{e' \in c} v(x' \cup e') = v(x') = w(x') = \sum_{e' \in c} w(x' \cup e')$. Since $v(x' \cup e) < w(x' \cup e)$ there must be also e'' such that S $v(x' \cup e'') > w(x' \cup e'')$. Call $x'' := x' \cup e''$. From $v(x'') > w(x'')$ we get $\nu(\uparrow x'') > \xi(\uparrow x'')$ which contradicts $\nu < \xi$. □

Theorem 7.6.4. *Let $\mathcal{E}$ be a confusion free event structure. A maximal continuous valuation in $\mathcal{V}^1(\mathcal{L}(\mathcal{E}))$ is non-leaking.*

Proof: We will show that a leaking continuous valuation is not maximal. For any leaking continuous valuation ν we will build another continuous valuation ν' with $\nu < \nu'$. Consider the pre-valuation $v : \mathcal{L}_{fin}(\mathcal{E}) \to [0, 1]$ defined as $v(x) = \nu(\uparrow x)$. Consider its unique extension $v_\partial : \mathcal{L}_{fin}(\mathcal{E}_\partial) \to [0,1]$ as in theorem 6.5.3. Since v is not a global valuation, there exists a cell $\bar{c}$ and a configuration x such that $v_\partial(x^{\partial_{\bar{c}}}) > 0$. By contravariance we have in particular that $v_\partial([\bar{c})^{\partial_{\bar{c}}}) > 0$. Choose an event $\bar{e} \in \bar{c}$. For every event $a \in \uparrow \bar{e}$ we consider a distinct "copy" a'. For $a = \bar{e}$ we put $\bar{e}' = \partial_{\bar{c}}$.

We build a new event structure $\mathcal{E}'_\partial$ as follows (we drop in the sequel the subscript ∂). We essentially copy the structure of $\uparrow \bar{e}$ above $\bar{e}' = \partial_{\bar{c}}$.

- $E' = E \cup \{a' \mid a \in \uparrow \bar{e}\}$;
- $\leq'$ is $\leq$ extended with the following clauses
  - if $a, b \in \uparrow \bar{e}$ then $a' \leq' b'$ if and only if $a \leq b$;
  - if $a \notin \uparrow \bar{e}, b \in \uparrow \bar{e}$, then $a \leq' b'$ if and only if $a \leq b$;
- $\#'$ is $\#$ extended with the following clause: if $a, b \in \uparrow \bar{e}$ then $a' \#'_\mu b'$ if and only if $a \#_\mu b$.

We first observe that $\mathcal{E}'$ is confusion free: clearly $\#' \cup 1_{E'}$ is an equivalence. Suppose $a' \#'_\mu b'$ and take $d \in \uparrow \bar{e}$. Note that by definition $a \#_\mu b$. Then $d' \leq' a'$ if and only if $d \leq a$ if and only if $d \leq b$ if and only if $d' \leq' b'$. Take now $d \notin \uparrow e$. Then $d \leq' a'$ if and only if $d \leq a$ if and only if $d \leq b$ if and only if $d \leq' b'$.

For every cell c such that $c > \bar{e}$, we have a cell $c' = \{a' \mid a \in c\}$.

In general, for every subset $z \subseteq E$ we define z' to be $z \setminus \uparrow \bar{e} \cup \{a' \mid a \in \uparrow \bar{e} \cap x\}$. We have that x is a configuration if and only if x' is a configuration. Consider $a' \in x'$. Pick $b \in \uparrow \bar{e}$. If $b' \leq' a'$ then $b \leq a$ so $b \in x$ and therefore $b' \in x$. Pick $b \notin \uparrow \bar{e}$. If $b \leq' a'$ then $b \leq a$ then $b \in x$ and therefore $b \in x'$. Therefore x' is downward closed if x is. Similarly one shows that x is downward closed if x' is.

If $a, b \in \uparrow \bar{e}$, we have that $a' \#' b'$ if and only if $a \# b$, because they inherit the conflict from events above e. Now take $a \notin \uparrow \bar{e}, b \in \uparrow \bar{e}$, and suppose $a \#' b'$. Then there exist $a_0 \leq' a$ and $b_0 \leq' b'$ such that $a_0 \#'_\mu b_0$. Since $a \notin \uparrow \bar{e}$ then $a_0 \notin \uparrow \bar{e}$. There are two cases. The first case is $b_0 = \bar{e}'$. Note that $a_0 \neq \bar{e}$ so that $a_0 \#_\mu \bar{e}$ and then $a \# b$. The second case is $b_0 \notin \uparrow \bar{e}$ so that $a_0 \#_\mu b_0$. We also have that $b_0 \leq b$ so that $a \# b$. Similarly if $a \# b$ then $a \#' b'$. Therefore x is conflict free if and only if x' is conflict free.

Now we pick a "local valuation" defined on the new events, that is a function $p : \{a' \mid a > \bar{e}\} \to]0,1]$ such that for every new cell c', $\sum_{a' \in c'} p(a') = 1$. This allows us to define a global valuation v' on $\mathcal{E}'$. If x' is a finite configuration of $\mathcal{E}'$,

$$v'(x') = v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a')\,.$$

We have to argue that this defines a global valuation on $\mathcal{E}'$. Consider a finite configuration x' and a c-covering C at x'. If $\bar{e'} \notin x'$ and $c \neq \bar{c}$ then $x = x'$ and

$$v'(x') = v(x') = \sum_{y' \in C} v(y') = \sum_{y' \in C} v'(y')\,.$$

If $c = \bar{c}$, then again $x = x'$

$$v'(x') = v(x') = \sum_{y' \in C} v(y') = \sum_{y' \in C} v'(y')\,.$$

If $\bar{e'} \not\in x'$ and $c \not\geq \bar{e'}$ then

$$v'(x') = v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a')$$

$$= \sum_{e \in c} v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\} \cup \{e\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a')$$

$$= \sum_{e \in c} v(x' \cup \{e\} \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a') = \sum_{e \in c} v'(x' \cup \{e\}) \, .$$

If $\bar{e'} \not\in x'$ and $c \geq \bar{e'}$, then

$$v'(x') = v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a')$$

$$= v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a') \cdot \sum_{e' \in c} p(e')$$

$$= \sum_{e' \in c} v(x' \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in x' \cap \uparrow \bar{e}'} p(a') \cdot p(e')$$

$$= \sum_{e' \in c} v((x' \cup e') \setminus \uparrow \bar{e}' \cup \{\bar{e}'\}) \cdot \prod_{a' \in (x' \cup e') \cap \uparrow \bar{e}'} p(a')$$

$$= \sum_{e' \in c} v'(x' \cup e') \, .$$

Now we go back to $\mathcal{E}_\partial$. We want to transfer the weight of $\partial_{\bar{c}}$ onto $\bar{e}$. We define a new valuation v'' as follows.

- If $\partial_{\bar{c}} \in x$ then $v''(x) = 0$.
- If $\bar{e} \not\in x$ then $v''(x) = v(x)$.
- If $\bar{e} \in x$ then $v''(x) = v(x) + v'(x')$

By a case analysis similar to the one above one checks that v'' is a global valuation on $\mathcal{E}_\partial$. Note that $v''([\bar{c})^{\partial_{\bar{c}}}) = 0 < v([\bar{c})^{\partial_{\bar{c}}})$, thus $v'' \neq v$. The global valuation v'' generates a continuous valuation ν'' on $\mathcal{L}(\mathcal{E})$. We finally argue that $\nu < \nu''$. Clearly $\nu \neq \nu''$. To show $\nu \leq \nu''$ i t is enough to check it on the elements of Bs. Consider n configurations of $x_1, \ldots x_n \in \mathcal{L}_{fin}(\mathcal{E})$. We want to show that

$$\sum_{\emptyset \neq I \subseteq I_n} 1^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) \leq \sum_{\emptyset \neq I \subseteq I_n} 1^{|I|-1} v''\left(\bigvee_{i \in I} x_i\right) \, .$$

Consider the second member. Let J be the set of $i \in I_n$ such that $\bar{e} \not\in x_i$. Then

$$\sum_{\emptyset \neq I \subseteq I_n} 1^{|I|-1} v''\left(\bigvee_{i \in I} x_i\right)$$

$$= \sum_{\emptyset \neq I \subseteq J} 1^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v''\left(\bigvee_{i \in I} x_i\right)$$

$$= \sum_{\emptyset \neq I \subseteq J} 1^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right)$$

$$= \sum_{\emptyset \neq I \subseteq I_n} 1^{|I|-1} v\left(\bigvee_{i \in I} x_i\right) + \sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right).$$

It is enough to show that

$$\sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right) \geq 0\,.$$

Call ν' the continuous valuation corresponding to v'. Then $\nu'(\bigcup_{i \in J} \uparrow x_i) \geq \nu'(\bigcup_{i \in I_n} \uparrow x_i)$. That is:

$$\sum_{\emptyset \neq I \subseteq J} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right) \geq \sum_{\emptyset \neq I \subseteq I_n} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right).$$

The difference between the two is exactly

$$\sum_{\emptyset \neq I \setminus J \subseteq I_n} 1^{|I|-1} v'\left(\bigvee_{i \in I} x'_i\right)$$

which is positive. □

Abbas Edalat [Eda95b] studied non-leaking continuous valuations in is domain theoretic presentation of integration. His Proposition 5.18 is similar to our Proposition 7.6.3. He also conjectures an analogous of Theorem 7.6.4 but he is not able to prove it.

7.7 Discussion and Future Work

We have presented a way to add probabilities to confusion free event structures, by means of global valuations. We have shown the relation between global valuations and continuous valuations on the domain of configurations. We have completely characterised the normalised continuous valuations on such domain as global valuations on event structures with "invisible" events. We have characterised the maximal normalised valuation, partially answering a question by Edalat. We have defined various notions of a run for probabilistic event structures and proved a confluence theorem.

The main drawback of the work presented here is that it applies to a restricted class of event structures. Which kind of distributed systems can be modelled with them? If we were to model languages with communication regimes like CCS, confusion free event structures would not be sufficient.

A first informal inspection seems to suggest that a language featuring the restricted (linear) communication regime of Kahn-MacQueen networks could be modelled by confusion-free event structures. We have recently designed such language, using a linear typing system for CCS analogous to the type system for π-calculus presented in [KPT99]. Interestingly [KPT99] contains a confluence result very similar to our Theorem 7.5.6.

As observed in the introduction of this thesis, general communication regimes require the use of nondeterminism, especially when different processes may compete to communicate on the same "channel". In the interleaving approach to probabilistic concurrent systems, this form of nondeterminism is not distinguishable from the nondeterminism arising from the interleaving. What the model we have presented does, however, is to factor out this second form of nondeterminism. The confluence result is saying exactly that when the only nondeterminism arises from interleaving, then this nondeterminism is illusory. As intuitive as this statement may be, ours is the first formal probabilistic models in which one can prove it.

The confluence result may also be interesting from the point of view of model checking. All the model checking tools for probabilistic concurrency (see for example [KNP02]) use the interleaving models. Modelling concurrency by interleaving has the effect of blowing up the size of the state space on which the checking is performed. The confluence theorem tells us that, in systems modelled by confusion free event structures, there is essentially only one possible probabilistic run. This may allow us to reduce the size of the state space. With this applications in mind, a study of our model in the presence of fairness assumptions may also provide useful insights.

A future line of work would also consist in generalising our definition, following the line of Katoen's work, and trying to generalise the result connecting event structures and domains. The presence of nondeterminism suggests that the introduction of powerdomains may be necessary. Such work would thus be a perfect conclusion for this thesis, but life is short and we are old and lazy. May another lonely wanderer pick up the map we have drawn, and continue the exploration.

7 Appendix Proof of Proposition 7.3.4

We recall the statement of the proposition. If C, C' are inductive tests, then

$$C \leq C' \iff C \to^* C'$$

Proof: $\Longleftarrow$) by induction on the derivation $C \to^* C'$.

$\Longrightarrow$) by induction on the derivation $\{\emptyset\} \to^* C$. Suppose $\{\emptyset\} \to^* C \xrightarrow{X,(c_x)} \tilde{C}$. And suppose $\tilde{C} \leq C'$. First of all we then have that $C \leq C'$. By induction hypothesis $C \to^* C'$, say

$$C = C_0 \xrightarrow{l_0} C_1 \xrightarrow{l_1} C_2 \ldots \xrightarrow{l_{k-1}} C_k = C'$$

Where $l_i = X_i, (c^i_{x_i})$. We are going to build a derivation

$$\tilde{C} = \tilde{C}_0 \xrightarrow{\tilde{l}_0} \tilde{C}_1 \xrightarrow{\tilde{l}_1} \tilde{C}_2 \ldots \xrightarrow{\tilde{l}_{k-1}} \tilde{C}_k = C'$$

Notice that $X \subseteq C$. We can define the sequence

$$X = Z_0 \xrightarrow{X_0 \cap Z_0, (c^0_{x_0})} Z_1 \xrightarrow{X_1 \cap Z_1, (c^1_{x_1})} Z_2 \ldots \xrightarrow{X_{k-1} \cap Z_{k-1}, (c^{k-1}_{x_{k-1}})} Z_k$$

For every i we have $Z_i \subseteq C_i$. Notice that for all i, $X \leq Z_i$ which implies that for every $x_i \in Z_i$ there exists a (unique) $x \in X$ with $x \leq x_i$, which we call $past(x_i)$.

The Z_i's represent the evolution of X.

Now, we partition every Z_i in three sets

- the set Y_i of the configurations x_i which do not fill $c_{past(x_i)}$ and such that if $x_i \in X_i$, then $c^i_{x_i} \neq c_{past(x_i)}$.
- the set V_i of the configurations x_i which do not fill $c_{past(x_i)}$ and such that $x_i \in X_i$ & $c^i_{x_i} = c_{past(x_i)}$.
- the set W_i of the configurations x_i which fill $c_{past(x_i)}$.

We also define $U_i := Y_i \cup V_i$.

We argue that $W_k = Z_k$. Recall that $\tilde{C} \leq C_k$. Take a configuration x_k of Z_k and consider the unique $z \in \tilde{C}$ such that $z \leq x_k$. Then z and $past(x_k)$ are compatible. For all $e \in c_{past(x_k)}$, we have that $past(x_k) \cup e \in \tilde{C}$. Reasoning like in the proof of Proposition 6.2.5, we argue that z must be compatible with $past(x_k) \cup e$ for one of the $e \in c_{past(x_k)}$. Since $\tilde{C}$ is a test, then $past(x_k) \cup e = z$, which implies that x_k fills $c_{past(x_k)}$.

So every configuration $x_k \in Z_k$ fills $past(x_k)$. This filling must happen sometime along the derivation $C \to^* C'$.

- Y_i is the set of configurations such that the filling has not happened yet and it is not going to happen at step i;
- V_i is the set of configurations such that the filling has not happened yet and it is going to happen at step i;
- W_i is the set of configurations such that the filling has already happened.

To build the derivation $\tilde{C} \to^* C'$ we follow $C \to^* C'$ step by step, keeping in mind that all the filling has already happened before we even start.

Therefore: the configurations in W_i are OK, and they are activated as expected; the configurations in V_i do not need to be activated; the configurations in Y_i are not in $\tilde{C}_i$, so they have to be modified (by filling the suitable cell), and after that they can be activated.

For every x_i in U_i, $c_{past(x_i)}$ is accessible at x_i. Let Y' be such that

$$X_i \cap Y_i \xrightarrow{X_i \cap Y_i, (c_{past(x_i)})} Y'_i\,.$$

Define $\tilde{X}_i$ as $(X_i \setminus U_i) \cup Y'_i$, and define $\tilde{l}_i$ as $\tilde{X}_i, (c^i_{k(\tilde{x}_i)})$, where $k(\tilde{x}_i) = \tilde{x}_i$ if $\tilde{x}_i \notin Y'_i$ while $k(\tilde{x}_i) = \tilde{x}_i \setminus c_{past(\tilde{x}_i)}$ if $\tilde{x}_i \in Y'_i$.

Now, for $i > 0$, define $\tilde{C}_i$ by

$$C_i \xrightarrow{U_i, (c_{past(x_i)})} \tilde{C}_i\,.$$

Then for every $0 \leq i \leq k$

$$\tilde{C}_i \xrightarrow{\tilde{l}_i} \tilde{C}_{i+1}$$

Once we have showed this, we are done. Indeed, since $U_k = \emptyset$, we have that $\tilde{C}_k = C_k = C'$. We have to prove that the following diagram "commutes":

$$\begin{tikzcd}
C_i \arrow[r, "{X_i,(c^i_{x_i})}"] \arrow[d, "{U_i,(c_{past(x_i)})}"'] & C_{i+1} \arrow[d, "{U_{i+1},(c_{past(x_{i+1})})}"] \\
\tilde{C}_i \arrow[r, "{\tilde{X}_i,(c^i_{k(\tilde{x}_i)})}"] & \tilde{C}_{i+1} .
\end{tikzcd}$$

Let C^* be such that

$$\tilde{C}_i \xrightarrow{\tilde{X}_i,(c_{k(\tilde{x}_i)})} C^* .$$

We show that $C^* = \tilde{C}_{i+1}$ by arguing that they are both equal to

$$C_i \setminus X_i \setminus Y_i \cup \bigcup_{x \in Y_i \cap X_i} x + c^i_x + c_{past(x)} \cup \bigcup_{x \in X_i \setminus Y_i} x + c^i_x \cup \bigcup_{x \in Y_i \setminus X_i} x + c_{past(x)}$$

where $x + c_1 + c_2$ denotes $\{x \cup \{e_1, e_2\} \mid e_1 \in c_1, e_2 \in c_2\}$. Let's start from $\tilde{C}_{i+1}$. We first observe that $V_i \subseteq X_i$. Also if $x_{i+1} \in U_{i+1}$, then

- either $x_{i+1} \in Y_i \setminus X_i$;
- or $x_{i+1} \in x_i + c^i_{x_i}$ for some $x_i \in Y_i \cap X_i$.

This is because if $x_{i+1} \in U_{i+1}$ then x_{i+1} does not fill $c_{past(x_{i+1})}$. The rest comes by observing that $U_{i+1} \subseteq Z_{i+1}$ and

$$Z_i \xrightarrow{X_i \cap Z_i,(c^i_{x_i})} Z_{i+1} \quad .$$

Note also that if $x_{i+1} \in x_i + c^i_{x_i}$, then $past(x_{i+1}) = past(x_i)$. Therefore

$$\begin{aligned}
\tilde{C}_{i+1} = \quad & C_{i+1} \setminus U_{i+1} && \cup \bigcup_{x_{i+1} \in U_{i+1}} x_{i+1} + c_{past(x_{i+1})} \\
= \quad & C_i \setminus X_i && \cup \bigcup_{x_i \in X_i} x_i + c^i_{x_i} \setminus U_{i+1} \\
& && \cup \bigcup_{x_{i+1} \in U_{i+1}} x_{i+1} + c_{past(x_{i+1})} \\
= \quad & C_i \setminus X_i \setminus Y_i && \cup \bigcup_{x_i \in X_i \setminus Y_i} x_i + c^i_{x_i} \\
& && \cup \bigcup_{x_i \in X_i \cap Y_i} x_i + c^i_{x_i} + c_{past(x_i)} \\
& && \cup \bigcup_{x_i \in Y_i \setminus X_i} x_i + c_{past(x_i)} .
\end{aligned}$$

Now for C^*. Recall that $\tilde{X}_i$ is defined as $X_i \setminus U_i \cup Y_i'$, where

$$X_i \cap Y_i \xrightarrow{X_i \cap Y_i, (c_{past(x_i)})} Y_i' \, .$$

We have that if $\tilde{x}_i \in \tilde{X}_i$ then

- either $\tilde{x}_i \in X_i \setminus U_i$;
- or $\tilde{x}_i \in x_i + c_{past(x_i)}$ for some $x_i \in Y_i \cap X_i$.

In the first case we have $k(\tilde{x}_i) = \tilde{x}_i$, in the second case we have $k(\tilde{x}_i) = x_i$ Recall also that if $x_i \in V_i$ then $c^i_{x_i} = c_past(x_i)$. Therefore

$$\begin{array}{rcl}
C^* = & \tilde{C}_i \setminus \tilde{X}_i & \cup \displaystyle\bigcup_{\tilde{x}_i \in \tilde{X}_i} \tilde{x}_i + c^i_{k(\tilde{x}_i)} \\
= & C_i \setminus U_i & \cup \displaystyle\bigcup_{x_i \in U_i} x_i + c_{past(x_i)} \setminus \tilde{X}_i \\
 & & \cup \displaystyle\bigcup_{\tilde{x}_i \in \tilde{X}_i} \tilde{x}_i + c^i_{k(\tilde{x}_i)} \\
= & C_i \setminus U_i \setminus X_i & \cup \displaystyle\bigcup_{x_i \in X_i \cap Y_i} x_i + c_{past(x_i)} + c^i_{x_i} \\
 & & \cup \left[\displaystyle\bigcup_{x_i \in X_i \setminus U_i} \tilde{x}_i + c^i_{\tilde{x}_i} \cup \bigcup_{x_i \in V_i} x_i + c^i_{x_i} \right] \\
 & & \cup \displaystyle\bigcup_{x_i \in Y_i \setminus X_i} c_{past(x_i)} \\
= & C_i \setminus X_i \setminus Y_i & \cup \displaystyle\bigcup_{x_i \in X_i \cap Y_i} x_i + c^i_{x_i} + c_{past(x_i)} \\
 & & \cup \displaystyle\bigcup_{x_i \in X_i \setminus Y_i} x_i + c^i_{x_i} \\
 & & \cup \displaystyle\bigcup_{x_i \in Y_i \setminus X_i} x_i + c_{past(x_i)} \, .
\end{array}$$

□

Addendum

Few weeks before the date of the defense of this thesis, we discovered a paper by Hagen Völzer [V01], which considerably overlaps with our work on event structures. In particular Theorem 6.4.1 could be proved more easily as consequence of his Theorem 1. It is not here the place to unravel the details of this observation. This is the subject of future work.

Index

Bibliography

[Abr87] Samson Abramsky. *Domain Theory and the Logic of Observable Properties.* PhD thesis, University of London - Queen Mary College, 1987.

[AJ94] Samson Abramsky and Achim Jung. Domain theory. In *Handbook of Logic in Computer Science*, volume 3. Oxford University Press, 1994.

[AM00] Mauricio Alvarez-Manilla. *Measure Theoretic Results for Continuous Valuations on Partially Ordered Spaces*. PhD thesis, University of London - Imperial College of Science, Technology and Medicine, September 2000.

[AMESD00] Mauricio Alvarez-Manilla, Abbas Edalat, and Nasser Saheb-Djaromi. An extension result for continuous valuations. *Journal of the London Mathematical Society*, 61(2):629–640, 2000.

[BD95] Dany Breslauer and Devdatt P. Dubhashi. Combinatorics for computer scientists. Technical Report LS-95-4, BRICS, 1995.

[BDEP97] Richard Blute, Josée Desharnais, Abbas Edalat, and Prakash Panangaden. Bisimulation for labelled markov processes. In *Proc. 12th LICS*, pages 149–158, 1997.

[Bec69] Jon Beck. Distributive laws. In *Seminar on Triples and Categorical Homology Theory*, pages 119–140, 1969.

[Bir67] Garrett Birkhoff. *Lattice theory.* American Mathematical Society, 1967.

[BK00] Christel Baier and Marta Kwiatkowska. Domain equations for probabilistic processes. *Mathematical Structures in Computer Science*, 10(6):665–717, 2000.

[Bra] Gilles Brassard. Quantum information processing for computer scientists. MIT press - to appear.

[BS01] Emanuele Bandini and Roberto Segala. Axiomatizations for probabilistic bisimulation. In *Proceedings of 28th ICALP*, volume 2076 of *LNCS*, pages 370–381, 2001.

[BSdV03] Falk Bartels, Ana Sokolova, and Erik de Vink. A hierarchy of probabilistic system types. In H. Peter Gumm, editor, *Electronic Notes in Theoretical Computer Science*, volume 82. Elsevier, 2003.

[C0́3] Mario J. Cáccamo. *A Calculus for Categories.* PhD thesis, BRICS - Aarhus, 2003.

[CHW02] Mario J. Cáccamo, J. Martin E. Hyland, and Glynn Winskel. Lecture notes in category theory, 2002. Available at http://www.brics.dk/∼mcaccamo.

[Coh81] Paul M. Cohn. *Universal Algebra.* Reidel, 1981.

[CW01] Federico Crazzolara and Glynn Winskel. Events in security protocols. In *Proc. of the 8th ACM Conference on Computer and Communication Security, Philadelphia*, 2001. Available as BRICS report RS-01-13.

[dA97] Luca de Alfaro. *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, December 1997.

[dAHJ01] Luca de Alfaro, Thomas A. Henzinger, and Ranjit Jhala. Compositional methods for probabilistic systems. In *Proc. 12th CONCUR*, volume 2154 of *LNCS*, pages 351–365, 2001.

[DEP98] Josée Desharnais, Abbas Edalat, and Prakash Panangaden. A logical characterization of bisimulation for labeled markov processes. In *Proc. 13th LICS*, pages 478–487, 1998.

[DGJP99] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labeled markov systems. In *Proceeding of 10th CONCUR*, volume 1664 of *LNCS*, pages 258–273, 1999.

[Eda95a] Abbas Edalat. Domain theory and integration. *Theoretical Computer Science*, 151(1):163–193, 13 November 1995.

[Eda95b] Abbas Edalat. Dynamical systems, measures and fractals via domain theory. *Information and Computation*, 120(1):32–48, 1995.

[Esc03] Martín Escardó. Topology of data types and computability concepts. Handwritten notes - availible at the author's web page, 2003.

[G$^+$03] Gerhard Gierz et al. *Continuous Lattices and Domains.* Cambridge University Press, 2003.

[Gau57] N.D. Gautam. The validity of equations of vomplex algebras. *Archiv für Mathematische Logik und Grundlagenforschung*, 3:117–124, 1957.

[GJS90] A. Giacalone, C.-C. Jou, and S.A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In IFIP TC 2, editor, *Proc. Working Conference on Programming Concepts and Methods*, Israel, April 1990.

[GM99] Shafi Goldwasser and Bellare Mihir. Lecture notes in cryptography. Available at http://www-cse.ucsd.edu/∼mihir/papers/gb.ps, 1999.

[Hal50] Paul Halmos. *Measure Theory.* van Nostrand, 1950. New edition by Springer in 1974.

[Han91] Hans Hansson. *Time and Probability in Formal Design of Distributed systems.* PhD thesis, Uppsala University, 1991.

[HJ89] Hans Hansson and Bengt Jonsson. A calculus for communicating systems with time and probabilities. In *Proc. 10th IEEE Real-Time Systems Symposium*, pages 102–111, 1989.

[HO00] J. Martin E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, December 2000.

[HP00] Mihaela Herescu and Catuscia Palamidessi. Probabilistic asynchronous π-calculus. In *Proc. 3rd FoSSaCS*, volume 1784 of *LNCS*, pages 146–160. Springer, 2000.

[HPP02] J. Martin E. Hyland, Gordon D. Plotkin, and John Power. Combining computational effects: Commutativity and sum. In *Proc. of IFIP TCS*, pages 474–484. Kluwer, 2002.

[HV99] Jerry I. den Hartog and Erik P. de Vink. Mixing up nondeterminism and probability: A preliminary report. *ENTCS 22*, 1999.

[JLY01] Bengt Jonsson, Kim G. Larsen, and Wang Yi. Probabilistic extensions of process algebras. In *Handbook of Process Algebras*. Elsevier, 2001.

[Jon90] Claire Jones. *Probabilistic Non-determinism.* PhD thesis, University of Edinburgh, 1990.

[JT98] Achim Jung and Regina Tix. The troublesome probabilistic powerdomain. In *Third Workshop on Computation and Approximation*, volume 13 of *Electronic Notes in Theoretical Computer Science*, 1998.

[Kah74] Gilles Kahn. The semantics of simple language for parallel programming. In *Proceedings of the IFIP congress 74*, pages 471–475. North-Holland, 1974.

[Kat96] Joost-Pieter Katoen. *Quantitative and Qualitative Extensions of Event Structures.* PhD thesis, University of Twente, 1996.

[Kir93] Olaf Kirch. Bereiche und Bewertungen. Master's thesis, Technische Hochschule Darmstadt, 1993.

[KM77] Gilles Kahn and David B. MacQueen. Coroutines and networks of parallel processes. In *Proceedings of the IFIP congress 77*, pages 993–998. North-Holland, 1977.

[KNP02] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Prism: Probabilistic symbolic model checker. In *Proceedings of 12th TOOLS*, volume 2324 of *LNCS*, pages 200–204. Springer, 2002. `http://www.cs.bham.uk/∼dxp/prism/`.

[Koz81] Dexter Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22:328–250, 1981.

[KP93] Gilles Kahn and Gordon D. Plotkin. Concrete domains. *Theoretical Computer Science*, 121(1-2):187–277, 1993.

[KPT99] Naoki Kobayashi, Benjamin C. Pierce, and David N. Turner. Linearity and the Pi-Calculus. *ACM Transactions on Programming Languages and Systems*, 21(5):914–947, 1999.

[Low93] Gavin Lowe. *Probabilities and Priorities in Timed CSP.* PhD thesis, Universit of Oxford, 1993.

[LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[Lyn96] Nancy Lynch. *Distributed Algorithms.* Morgan Kaufmann, 1996.

[M^+94] Carrol Morgan et al. Refinement-oriented probability for CSP. Technical Report PRG-TR-12-94, Oxford University Computing Laboratory, August 1994.

[M^+95] Carroll Morgan et al. Argument duplication in probabilistic CSP. Technical Report TR-11-95, University of Oxford, October 1995.

[Mac71] Saunders MacLane. Categories for the working mathematician. Berlin, 1971.

[Maz86] Antoni Mazurkiewicz. Trace theory. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *LNCS*, pages 279–324. Springer, 1986.

[Mil77] Robin Milner. Fully abstract models of typed lambda-calculus. *Theoretical Computer Science*, 4:1–22, 1977.

[Mil89] Robin Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[Mis00] Michael Mislove. Nondeterminism and probabilistic choice: Obeying the law. In *Proc. 11th CONCUR*, volume 1877 of *LNCS*, pages 350–364. Springer, 2000.

[Mog91] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, July 1991.

[NC00] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge, 2000.

[NPW81] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part i. *Theoretical Computer Science*, 13(1):85–108, 1981.

[NW03] Mikkel Nygaard and Glynn Winskel. Domain theory for concurrency. *Theoretical Computer Science*, 2003. To appear.

[Ong95] C.-H. Luke Ong. Correspondence between operational and denotational semantics. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Vol 4*, pages 269–356. Oxford University Press, 1995.

[Plo76] Gordon D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5(3):452–487, 1976.

[Plo77] Gordon D. Plotkin. LCF considerd as a programming language. *Theoretical Computer Science*, 5:223–225, 1977.

[Plo83] Gordon D. Plotkin. Domains. University of Edinburgh, 1983.

[PP02] Gordon D. Plotkin and John Power. Notions of computation determine monads. In *Proc of 5th FOSSACS*, volume 2303 of *LNCS*, pages 342–256. Springer, 2002.

[Pro70] Giovanni Prodi. *Analisi Matematica*. Bollati Boringhieri, 1970.

[Put94] Martin L. Puterman. *Markov decision processes : discrete stochastic dynamic programming*. Wiley, New York, 1994.

[PZ93] Amir Pnueli and Lenore Zuck. Probabilistic verification. *Information and Computation*, 103:1–29, 1993.

[Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[RT86] Grzegorz Rozenberg and P.S. Thiagarajan. Petri nets: Basic notions, structure, behaviour. In *Current Trends in Concurrency*, volume 224 of *LNCS*, pages 585–668. Springer, 1986.

[RT94] Jan J.M.M. Rutten and Daniele Turi. Initial algebra and final coalgebra semantics for concurrency. *Lecture Notes in Computer Science*, 803:530–582, 1994.

[Sco72] Dana S. Scott. Continuous lattices. In F. William Lawvere, editor, *Toposes, Algebraic Geometry, and Logic*, volume 274 of *Lecture Notes in Computer Science*, pages 97–136. Springer-Verlag, Berlin, Heidelberg, and New York, 1972.

[SD80] Nasser Saheb-Djaromi. Cpo's of measures for nondeterminism. *Theoretical Computer Science*, 12:19–37, 1980.

[Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, M.I.T., 1995.

[Sei95] Karen Seidel. Probabilistic communicating processes. *Theoretical Computer Science*, 152:219–249, 1995.

[SL95] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995. An extended abstract appears in *Proc. 5th CONCUR*, Uppsala, Sweden, LNCS 836, pages 481–496, August 1994.

[Smy78] Michael B. Smyth. Powerdomains. *Journal of Computer and Systems Sciences*, 16(1):23–36, February 1978.

[Smy83] Micheal B. Smyth. Powerdomains and predicate transformers: a topological view. In *Proceedings of 10th ICALP*, volume 154 of *LNCS*. Springer, 1983.

[Sri95] Aravind Srinivasan. The role of randomness in computation. Technical Report NS-95-6, BRICS, 1995.

[SS71] Dana S. Scott and Christopher Strachey. Towards a mathematical semantics for computer languages. In *Proceedings, 21st Symposium on Computers and Automata*, pages 19–46. Polytechnic Institute of Brooklyn, 1971. Also, Programming Research Group Technical Monograph PRG–6, Oxford University.

[Sto02] Mariëlle Stoelinga. An introduction to probabilistic automata. *Bulletin of the European Association for Theoretical Computer Science*, 78:176–198, October 2002.

[Tix99] Regina Tix. *Continuous D-Cones: Convexity and Powerdomain Constructions.* PhD thesis, Technische Universität Darmstadt, 1999.

[Tix00] Regina Tix. Convex power constructions for continuous d-cones. In Dieter Spreen, editor, *Electronic Notes in Theoretical Computer Science*, volume 35. Elsevier Science Publishers, 2000.

[V0̈1] Hagen Völzer. Randomized non-sequential processes. In *Proceedings of 12th CONCUR*, volume 2154 of *LNCS*, pages 184–201, 2001.

[Var85] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 26th FOCS*, pages 327–338, 1985.

[Var02] Daniele Varacca. The powerdomain of indexed valuations. In *Proc. of 17th IEEE Symposium on Logic in Computer Science*, 2002.

[vG$^+$90] Rob van Glabbeek et al. Reactive, generative, and stratified models of probabilistic processes. In *Proc. 5th LICS*, pages 130–141, 1990.

[Vic96] Steve Vickers. *Topology via Logic.* Cambridge University Press, 1996.

[VN03] Daniele Varacca and Mogens Nielsen. Probabilistic petri nets and Mazurkiewicz equivalence. Draft, February 2003.

[Win80] Glynn Winskel. *Events in Computation.* Ph.D. thesis, Dept. of Computer Science, University of Edinburgh, 1980.

[Win82] Glynn Winskel. Event structure semantics for CCS and related languages. In *Proceedings of 9th ICALP*, volume 140 of *LNCS*, pages 561–576. Springer, 1982.

[Win83] Glynn Winskel. A note on powerdomains and modality. In *FCT*, volume 158 of *LNCS*, pages 505–514, 1983.

[Win87] Glynn Winskel. Event structures. In *Advances in Petri Nets 1986, Part II; Proceedings of an Advanced Course,* Bad Honnef, September 1986, volume 255 of *LNCS*, pages 325–392. Springer, 1987.

[Win93] Glynn Winskel. *The Formal Semantics of Programming Languages.* MIT Press, 1993.

[WN95] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of logic in Computer Science*, volume 4. Clarendon Press, 1995.

Recent BRICS Dissertation Series Publications

DS-03-14 Daniele Varacca. *Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation*. November 2003. PhD thesis. xii+163 pp.

DS-03-13 Mikkel Nygaard. *Domain Theory for Concurrency*. November 2003. PhD thesis. xiii+161 pp.

DS-03-12 Paulo B. Oliva. *Proof Mining in Subsystems of Analysis*. September 2003. PhD thesis. xii+198 pp.

DS-03-11 Maciej Koprowski. *Cryptographic Protocols Based on Root Extracting*. August 2003. PhD thesis. xii+138 pp.

DS-03-10 Serge Fehr. *Secure Multi-Player Protocols: Fundamentals, Generality, and Efficiency*. August 2003. PhD thesis. xii+125 pp.

DS-03-9 Mads J. Jurik. *Extensions to the Paillier Cryptosystem with Applications to Cryptological Protocols*. August 2003. PhD thesis. xii+117 pp.

DS-03-8 Jesper Buus Nielsen. *On Protocol Security in the Cryptographic Model*. August 2003. PhD thesis. xii+247 pp.

DS-03-7 Mario José Cáccamo. *A Formal Calculus for Categories*. June 2003. PhD thesis. xiv+151.

DS-03-6 Rasmus K. Ursem. *Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization*. June 2003. PhD thesis. xiv+183 pp.

DS-03-5 Giuseppe Milicia. *Applying Formal Methods to Programming Language Design and Implementation*. June 2003. PhD thesis. xvi+211.

DS-03-4 Federico Crazzolara. *Language, Semantics, and Methods for Security Protocols*. May 2003. PhD thesis. xii+159.

DS-03-3 Jiří Srba. *Decidability and Complexity Issues for Infinite-State Processes*. 2003. PhD thesis. xii+172 pp.

DS-03-2 Frank D. Valencia. *Temporal Concurrent Constraint Programming*. February 2003. PhD thesis. xvii+174.

DS-03-1 Claus Brabrand. *Domain Specific Languages for Interactive Web Services*. January 2003. PhD thesis. xiv+214 pp.

DS-02-5 Rasmus Pagh. *Hashing, Randomness and Dictionaries*. October 2002. PhD thesis. x+167 pp.