EPTCS 126

Proceedings of the

Workshop on
Fixed Points in Computer Science

Turino, Italy, September 1st, 2013

Edited by: David Baelde and Arnaud Carayol

Published: 28th August 2013
DOI: 10.4204/EPTCS.126
ISSN: 2075-2180
Open Publishing Association

Preface

This volume contains the proceedings of the Ninth Workshop on *Fixed Points in Computer Science* (FICS 2013) which took place on September 1st 2013 in Torino, Italy as a satellite event of the conference *Computer Science Logic* (CSL 2013).

Fixed points play a fundamental role in several areas of computer science. They are used to justify (co)recursive definitions and associated reasoning techniques. The construction and properties of fixed points have been investigated in many different settings such as: design and implementation of programming languages, logics, verification, databases. The FICS workshop aims to provide a forum for researchers of the computer science and logic communities who study or apply the theory of fixed points.

The editors thank all authors who submitted papers to FICS 2013, and program committee members Andreas Abel, Lars Birkedal, Javier Esparza, Neil Ghani, Dexter Kozen, Ralph Matthes, Paul-André Melliès, Matteo Mio, Luke Ong, Pawel Parys, Luigi Santocanale, Makoto Tatsuta and Wolfgang Thomas for their work in selecting papers.

Apart from presentations of accepted papers, we are delighted that FICS 2013 featured three invited talks: Anuj Dawar on *Fixed point approximations of graph isomorphisms*, Nicola Gambino on *Cartesian closed bicategories* and Alexandra Silva on *Rational fixed points in programming languages*. Many thanks to them for having accepted the invitation.

Finally, we would like to express our deep gratitude to CSL 2013 for local organization and to EACSL, INRIA and Université Marne-la-Vallée for funding FICS 2013.

David Baelde and Arnaud Carayol

Table of Contents

Non-monotonic Pre-fixed Points and Learning

Stefano Berardi   Ugo de'Liguoro

Università di Torino

stefano.berardi@unito.it   ugo.deliguoro@unito.it

We consider the problem of finding pre-fixed points of interactive realizers over arbitrary knowledge spaces, obtaining a relative recursive procedure. Knowledge spaces and interactive realizers are an abstract setting to represent learning processes, that can interpret non-constructive proofs. Atomic pieces of information of a knowledge space are stratified into levels, and evaluated into truth values depending on knowledge states. Realizers are then used to define operators that extend a given state by adding and possibly removing atoms: in a learning process states of knowledge change non-monotonically. Existence of a pre-fixed point of a realizer is equivalent to the termination of the learning process with some state of knowledge which is free of patent contradictions and such that there is nothing to add. In this paper we generalize our previous results in the case of level 2 knowledge spaces and deterministic operators to the case of ω-level knowledge spaces and of non-deterministic operators.

1 Introduction

A fundamental aspect of constructive interpretations of classical arithmetic is how information is gathered and handled while looking for a witness of the proved formulas. This has been understood by several authors as a problem of control and side effects, although intended in different ways. Building over Coquand's semantics of evidence of classical arithmetic [7] and its representation as limiting interaction sequences [3], we have developed the concept of *interactive realizability* in [2, 4], which consists of interpreting non constructive proofs as effective strategies that "learn" the witness.

According to [5], learning the truth of an arithmetical statement can be abstractly presented as a process going through steps, which we call *states of knowledge*, such that a (candidate) witness can be relatively computed out of them. These are certain subsets of a countable set $\mathbb{A}$ whose elements are pieces of evidence that we dub *answers*. On the other hand $\mathbb{A}$ is equipped with an equivalence relation $\sim$ whose equivalence classes $[a]_\sim$ are *questions*; since we allow that in a state of knowledge each question has at most one answer, we say that X is a state if for all $a \in \mathbb{A}$, the set $X \cap [a]_\sim$ is either a singleton or empty. We also denote by $\mathbb{S}$ the set of states.

Over states we can define a "query map" $\mathsf{q}([a]_\sim, X) \in \mathscr{P}_{fin}(\mathbb{A})$, taking a question $[a]_\sim$, a state $X \in \mathbb{S}$, and returning the set $X \cap [a]_\sim$, that is a singleton $\{b\}$ if $b \in X$ is the only answer to $[a]_\sim$; the empty set otherwise. We call *state topology* the smallest topology making the query map continuous. Equivalently the state topology is generated by the canonical sub-basics $A_a = \{X \in \mathbb{S} \mid a \in X\}$ and $B_a = \{X \in \mathbb{S} \mid X \cap [a]_\sim = \emptyset\}$ for $a \in \mathbb{A}$.

Knowledge is improved by means of "realizers" $r : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$ that are functions guessing a finite set of new information $r(X)$ with respect to the current state of knowledge X. We assume that $r(X) \subseteq \mathbb{A}$ is always a finite set, so that a step of an "algorithm" to compute with r consists of proceeding from some X to $X' \cup Y$, that we treat here as a reduction relation $X \twoheadrightarrow^r_1 X' \cup Y$, where $X' \subseteq X$ and $\emptyset \neq Y \subseteq r(X) \setminus X$

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 1–10, doi:10.4204/EPTCS.126.1

have to satisfy certain requirements. Under this respect if $r(X) \subseteq X$, namely X is a pre-fixed point of r, then the computation terminates in the state X.

In [2, 4] we have studied the case where $\mathbb{A}$ is essentially made of decidable arithmetical statements which are known to be true, and considered the case where $r(X)$ is either a singleton or it is empty. In this case $X \twoheadrightarrow^r_1 X \cup r(X)$ if $r(X) \neq \emptyset$, and the sequence of reductions $X_0 \twoheadrightarrow^r_1 X_1 \twoheadrightarrow^r_1 \cdots$ out of some X_0 is uniquely determined by r and the sequence $X_0 \subseteq X_1 \subseteq \cdots$ is monotonic. Hence we have proved termination by applying Knaster-Tarski theorem.

We call deterministic the case in which $r(X)$ is at most a singleton. A first generalization of the picture is when $r(X)$ may include more than one answer, which is the non-deterministic case. Then $r(X)$ is not required to be a state, and the next state is $X \cup Y$, for some non-deterministic choice of a subset $Y \subseteq r(X)$ of pairwise unrelated answers w.r.t. $\sim$. A further extension is when X' is a proper subset of X in the reduction step $X \twoheadrightarrow^r_1 X' \cup Y$, then loosing the monotonicity of the sequence $X_0, X_1, \ldots$. This is the case when the truth values of answers are logically related, and adding some new answer may turn to false the truth values of some previously true answers. In this case whenever we add some answer we have also to remove some, and the fixed point result becomes difficult to prove.

To model logical dependencies of answers we assume that $\mathbb{A}$ is "stratified" by a map $\mathsf{lev} : \mathbb{A} \to \mathbb{N}$, splitting the answers into ω levels, in decreasing order of "reliability". As we explained in [2, 4], we need ω-levels of answers to describe the constructive content of classical proofs of arithmetic. Logical dependence means that an answer of level n (e.g. a universal statement) that has been considered as true so far, might be falsified by discovering that an answer of level $< n$ (a counterexample) should be true. Hence we relativize the truth value of answers to a state (to which they do not necessarily belong) using a function $\mathsf{tr}(a, X)$ that only depends on the answers in X having a smaller level, that is $\mathsf{tr}(a, X) = \mathsf{tr}(a, \{x \in X \mid \mathsf{lev}(x) < \mathsf{lev}(a)\})$. Further we require that $\mathsf{tr}(a, X)$ depends continuously on the state parameter w.r.t. the state topology. This is how we abstractly capture the idea that this should be a relative computable function, which will be recursive in case of a finite set X of answers. Instead, we add *no* level restriction on a realizer r: if $X \in \mathbb{S}$, then the answers of level n in $r(X)$ may depend on the answers of *any* level in X, including the answers of level $\geq n$ in X. Finally we also say that $X \in \mathbb{S}$ is *sound* if $\mathsf{tr}(a, X) = \mathsf{T}$ for all $a \in X$. Only sound pre-fixed points are of interest.

The fact that the truth value of an answer w.r.t. a state X only depends on truth values of lower level answers in X suggests the following non-deterministic algorithm to find a sound pre-fixed point of the function r: we pick one or more answers with the same level n from $r(X)$ and dropping all answers of level $> n$ from X. We express the algorithm through the relation $X \twoheadrightarrow^r_1 X' \cup Y$ whenever $X' = \{x \in X \mid \mathsf{lev}(x) \leq n\}$ and $Y \subseteq r(X)$ is a finite *homogeneous* state made of answers of the same level, say n, which is considered as the level of the state. Then we establish the main result of the paper, namely that if r is a *realizer* (see Definition 2.1 below) then any reduction $\twoheadrightarrow^r$ out of some sound X_0 terminates, within a finite number of steps, by a sound pre-fixed point of r, which is finite if X_0 is such.

We have a final warning about the proof in this paper. It is possible to show that our termination result implies the 1-consistency of First Order Arithmetic, and therefore it is not provable in it. Thus, no elementary proof of our result is possible, although we have found several non-elementary proofs. The proof included here is classical and it uses set theory, choice axiom and uncountable reduction sequences: none of them is strictly required, but we trade off logical complexity for readability. We could remove ordinals, choice axiom and even Excluded Middle from the proof, at the price of a harder (and longer) argument.

The plan of the paper is as follows. In §2 we define a reduction relation on states depending on a realizer r, which is the non-deterministic algorithm to search a pre-fixed point of r. In §3 we prove that the set of states from which this algorithm always terminates is an open set in the state topology. In §4 we use this fact to prove that if there is some reduction sequence of length ω out of some state, then there is a reduction sequence of length ω_1 out of the same state. Eventually, in §5, we prove that reduction sequences of length ω_1 do not exist, so that we conclude that all reduction sequences of our algorithm are of finite length. Then in §6 we discuss some related works and we conclude.

2 A non-deterministic parallel algorithm for finding pre-fixed points

For convenience we recall the basic definitions from [5] and the introduction above. We are given a countable set $\mathbb{A}$ and an equivalence relation $\sim$ over $\mathbb{A}$; the map $\mathsf{lev} : \mathbb{A} \to \mathbb{N}$ respects $\sim$ that is $\mathsf{lev}(x) = \mathsf{lev}(y)$ if $x \sim y$; $X \subseteq \mathbb{A}$ is a state if for all $x, y \in X$, $x \neq y$ implies $x \not\sim y$; the set $\mathbb{S}$ of states is taken with the state topology, generated by the sub-basics $A_a = \{X \in \mathbb{S} \mid a \in X\}$ and $B_a = \{X \in \mathbb{S} \mid X \cap [a]_\sim = \emptyset\}$; we take $\mathbb{A}$ and 2 with the discrete topology and $\mathbb{A} \times \mathbb{S}$ with the product topology.

Definition 2.1 (Layered Valuation, Sound State and Realizer) *A* layered valuation *over* $(\mathbb{A}, \sim, \mathsf{lev})$, *shortly a* valuation, *is a continuous mapping* $\mathsf{tr} : \mathbb{A} \times \mathbb{S} \to 2$ *such that*

$$\mathsf{tr}(a, X) = \mathsf{tr}(a, \{x \in X \mid \mathsf{lev}(x) < \mathsf{lev}(a)\}).$$

A state $X \in \mathbb{S}$ *is* sound *if* $\mathsf{tr}(x, X) = \mathsf{T}$ *for all* $x \in X$.

A realizer *w.r.t. the valuation* tr *is a continuous map* $r : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$*, where* $\mathscr{P}_{fin}(\mathbb{A})$ *is taken with the discrete topology, which is such that:*

$$\forall X \in \mathbb{S}\ \forall a \in r(X).\ X \cap [a]_\sim = \emptyset\ \ \&\ \ \mathsf{tr}(a, X) = \mathsf{T}.$$

Given $n \in \mathbb{N}$ and a state X we define the subsets of X:

$$X \upharpoonright_{<n} = \{x \in X \mid \mathsf{lev}(x) < n\}, \quad X \upharpoonright_{>n} = \{x \in X \mid \mathsf{lev}(x) > n\}, \quad X \upharpoonright_{=n} = \{x \in X \mid \mathsf{lev}(x) = n\}.$$

We also write $X \upharpoonright_{\leq n} = X \upharpoonright_{<n} \cup X \upharpoonright_{=n}$. We denote by $\mathbb{S}_{\text{fin}}$ the set of finite states; let $s, s', t, t', \ldots$ range over $\mathbb{S}_{\text{fin}}$.

Definition 2.2 (Reduction) *We say that a state* $s \in \mathbb{S}_{fin}$ *is* homogeneous *if* $s \neq \emptyset$ *and for some* $n \in \mathbb{N}$, $\mathsf{lev}(x) = n$ *for all* $x \in s$*; then we write* $\mathsf{lev}(s) = n$*. For any homogeneous* s *of level* n *we define a map* $\mathtt{R}_s : \mathbb{S} \to \mathbb{S}$ *by:*

$$\mathtt{R}_s(X) \upharpoonright_{<n} = X \upharpoonright_{<n}, \quad \mathtt{R}_s(X) \upharpoonright_{=n} = X \upharpoonright_{=n} \cup s, \quad \mathtt{R}_s(X) \upharpoonright_{>n} = \emptyset.$$

Then, given a realizer r *and an homogeneous* s *we define the binary* reduction relation *over* $\mathbb{S}$ *by:*

$$X \twoheadrightarrow^{s,r} Y \Leftrightarrow s \subseteq r(X)\ \ \&\ \ \mathtt{R}_s(X) = Y.$$

We say that X *reduces to* Y *in one step* and we write $X \twoheadrightarrow^r_1 Y$ if $X \twoheadrightarrow^{s,r} Y$ for some homogeneous s. As immediate consequence of the definitions of $\twoheadrightarrow^r_1$, tr and r we establish:

Lemma 2.3

1. $X \twoheadrightarrow^r_1 Y\ \ \&\ \ X \in \mathbb{S}_{fin} \Rightarrow Y \in \mathbb{S}_{fin}$.
2. $X \twoheadrightarrow^r_1 Y\ \ \&\ \ X$ *is sound* $\Rightarrow Y$ *is sound.*

3. $\neg\exists Y.\, X \twoheadrightarrow^r_1 Y \Leftrightarrow r(X) \subseteq X$.

A reduction sequence of length n from X to Y is a sequence $X_0,\ldots,X_n$ such that $X = X_0 \twoheadrightarrow^r_1 X_1 \twoheadrightarrow^r_1 \ldots \twoheadrightarrow^r_1 X_n = Y$. An infinite reduction sequence out of X is an endless sequence $X = X_0 \twoheadrightarrow^r_1 X_1 \twoheadrightarrow^r_1 \ldots \twoheadrightarrow^r_1 X_n \ldots$ of reductions. For any integer $n \in \mathbb{N}$ we say that X reduces to Y in n steps and we write $X \twoheadrightarrow^r_n Y$ if there is a length n reduction sequence from X to Y. We write $X \twoheadrightarrow^r Y$ if $X \twoheadrightarrow^r_n Y$ for some $n \in \mathbb{N}$.

We observe that X is a pre-fixed point of r, that is $r(X) \subseteq X$, if and only if there is no homogeneous set $s \subseteq r(X)$ such that $X \cap s = \emptyset$, that is if and only if for all $Y \in \mathbb{S}$ we have $X \not\twoheadrightarrow^r_1 Y$. If $\sim$ is decidable and both r and $\mathtt{tr}$ are relative recursive then we can see $Y \twoheadrightarrow^r_1 Z$ as the one step relation of a non-deterministic algorithm computing a pre-fixed point X of r starting with some $X_0 \in \mathbb{S}$; then such an X, if any, can be seen as a result of the computation starting with X_0. By lemma 2.3 we know that if we move from some finite sound state s_0, e.g. $\emptyset$, the reduction relation $\twoheadrightarrow^r_1$ generates a tree with finite and sound states as nodes, which is finitary because $r(X)$ is finite even for infinite X so that there can be only finitely many homogeneous $s \subseteq r(X)$. In particular the relation $X \twoheadrightarrow^r_1 Y$ is decidable for finite X and Y, and relative recursive in general.

We say that $X \in \mathbb{S}$ is strongly normalizing if all reduction sequences out of X are finite. We denote by $\mathtt{SN}$ the set of all strongly normalizing states. Our thesis is that $\mathtt{SN} = \mathbb{S}$, namely that the reduction tree out of any X is finite. This implies that if $s \in \mathbb{S}_{\text{fin}}$ and s is sound we can effectively find a finite and sound pre-fixed point t of r by reducing s.

3 The set of strongly normalizing states is open

The first step toward establishing $\mathtt{SN} = \mathbb{S}$ is to prove that $\mathtt{SN}$ is open in the state topology. To prove this we first characterize the reduction relation.

Lemma 3.1 (Reduction) *Let $s \in \mathbb{S}_{fin}$ be any homogeneous state of level n. Assume $X, Y \in \mathbb{S}$ and $X \twoheadrightarrow^{s,r} Y$. Let $m \in \mathbb{N}$.*

1. $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$
2. $X \not\twoheadrightarrow^{s,r} X$.
3. *If $m \leq n$, then* $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$
4. *If* $X \upharpoonright_{<m+1} \not\subseteq Y \upharpoonright_{<m+1}$, *then* $Y \upharpoonright_{=m} = \emptyset$.
5. *If* $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ *then* $m \leq n$.
6. *If* $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ *then* $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$

Proof

1. By definition of $X \twoheadrightarrow^{s,r} Y$ we have $s \neq \emptyset$, $X \cap s = \emptyset$ and $Y \upharpoonright_{=n} = X \upharpoonright_{=n} \cup s$. We conclude $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$.
2. By point 1, if $X \twoheadrightarrow^{s,r} Y$ then $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$, hence $Y \neq X$.
3. Assume $m \leq n$ in order to prove $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$. We reason by cases.
   (a) Let $m < n$. Then $m+1 \leq n$. By definition of $X \twoheadrightarrow^{s,r} Y$ we have $X \upharpoonright_{<n} = Y \upharpoonright_{<n}$, and from $m+1 \leq n$ we conclude $X \upharpoonright_{<m+1} = (X \upharpoonright_{<n}) \upharpoonright_{<m+1} = (Y \upharpoonright_{<n}) \upharpoonright_{<m+1} = Y \upharpoonright_{<m+1}$.
   (b) Let $m = n$. Then by point 1 above and $X \upharpoonright_{<n} = Y \upharpoonright_{<n}$ we have $X \upharpoonright_{<m+1} = X \upharpoonright_{<n+1} \subset Y \upharpoonright_{<n+1} = Y \upharpoonright_{<m+1}$

4. By point 3, if $X \upharpoonright_{<m+1} \not\subseteq Y \upharpoonright_{<m+1}$, then $m > n$. We deduce $Y \upharpoonright_{=m} \subseteq Y \upharpoonright_{>n} = \emptyset$.
5. Assume $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ in order to prove that $m \leq n$. If it were $m > n$, we would deduce $X \upharpoonright_{=n} = (X \upharpoonright_{<m}) \upharpoonright_{=n} = (Y \upharpoonright_{<m}) \upharpoonright_{=n} = Y \upharpoonright_{=n}$, contradicting point 1. Thus, $m \leq n$.
6. We apply points 5 and 3 in this order.

The next step is to prove that $\mathtt{SN}$ is open in the state topology. For all $n \in \mathbb{N}$, $n > 0$ we define $\mathtt{SN}_n = \{X \in \mathbb{S} | \forall Y \in \mathbb{S}. X \not\twoheadrightarrow^r_n Y\}$ the set of states from which there is no reduction sequences of length n from X. The reduction tree $\mathtt{T}(X) = \{Y \in \mathbb{S} | X \twoheadrightarrow^r Y\}$ from $X \in \mathbb{S}$ is finitely branching: from any node Y, the number of children of Y has upper bound the number of subsets of $r(Y)$, which is finite. By König's Lemma, $\mathtt{T}(X)$ is finite if and only if all branches of tree (all reduction sequences from X) are finite. Thus, $\mathtt{T}(X)$ is finite if and only if there is some upper bound $n \in \mathbb{N}$ to the reduction sequences from X. This implies $\mathtt{SN} = \bigcup_{n \in \mathbb{N}} \mathtt{SN}_n$. Therefore in order to prove that $\mathtt{SN}$ is open it is enough to prove that all $\mathtt{SN}_n$ are open.

Lemma 3.2 ($\mathtt{SN}$ is open) *Assume $s \in \mathbb{S}_{\mathit{fin}}$ is any homogeneous state. Let $I, I_0, I_1 \in \mathscr{P}_{\mathit{fin}}(\mathbb{A})$ be finite sets of answers. Assume $X, Y, X', Y' \in \mathbb{S}$.*

1. *For all $a \in \mathbb{A}$, $\{X \in \mathbb{S} | a \notin X\}$ is open.*
2. *If (I_0, I_1) is a partition of I, then $\{X \in \mathbb{S} | (I \cap X = I_0) \wedge (I \setminus X = I_1)\}$ is open.*
3. *$\mathtt{R}_s : \mathbb{S} \to \mathbb{S}$ is a continuous map.*
4. *$\mathtt{SN}_1$ is open.*
5. *For all $n \in \mathbb{N}$, $\mathtt{SN}_n$ is open.*
6. *$\mathtt{SN}$ is open*

Proof

1. Assume $a \in \mathbb{A}$ and $O = \{X \in \mathbb{S} | a \notin X\}$. The set O consists of all states including some element of $[a]_\sim$ different from a, or having empty intersection with $[a]_\sim$. Thus O is the union of all sets $\{X \in \mathbb{S} | b \notin X\}$ for $b \in [a]_\sim$ and $b \not a$, and of the set $\{X \in \mathbb{S} | X \cap [a]_\sim = \emptyset\}$. All these sets are basic open of the state topology, therefore O is an open set of the state topology.
2. Assume that (I_0, I_1) is a partition of I and $O = \{X \in \mathbb{S} | (I \cap X = I_0) \wedge (I \setminus X = I_1)\}$. Since both $(I \cap X, I \setminus X)$ and (I_0, I_1) are partitions of I, the condition $(I \cap X = I_0) \wedge (I \setminus X = I_1)$ is equivalent to $(I \cap X \supseteq I_0) \wedge (I \setminus X \supseteq I_1)$. Thus, O is equal to the intersection of all sets $\{X \in \mathbb{S} | a \in X\}$, for any $a \in I_0$, and of all sets $\{X \in \mathbb{S} | a \notin X\}$, for $a \in I_1$. These sets are finitely many because I is finite, and are either sub-basic open, or are open by point 1 above. Thus, O, being a finite intersection of open sets, is open.
3. Assume $s \in \mathbb{S}_{\mathrm{fin}}$ is an homogeneous state of level n. Assume $a \in \mathbb{A}$ and $A_a\{Z \in \mathbb{S} | a \in Z\}$, $B_a = \{Z \in \mathbb{S} | Z \cap [a]_\sim\}$ are sub-basic open. We have to prove that if $Y \in A_a$ then $\mathtt{R}_s^{-1}(A_a)$, $\mathtt{R}_s^{-1}(B_a)$ are open sets. We prove this statement by case analysis.
   (a) If $\mathsf{lev}(a) > n$ then $\mathtt{R}_s^{-1}(A_a) = \emptyset$.
   (b) If $a \in s$ then $\mathtt{R}_s^{-1}(A_a) = \mathbb{S}$.
   (c) If $\mathsf{lev}(a) \leq n$ and $a \notin s$ then $\mathtt{R}_s^{-1}(A_a) = A_a$.
   (d) If $\mathsf{lev}(a) > n$ then $\mathtt{R}_s^{-1}(B_a) = \mathbb{S}$.
   (e) If $s \cap [a]_\sim \neq \emptyset$ then $\mathtt{R}_s^{-1}(B_a) = \emptyset$.

(f) If $\mathsf{lev}(a) \leq n$ and $s \cap [a]_{\sim} = \emptyset$ then $\mathtt{R}_s^{-1}(B_a) = B_a$.

4. $\mathtt{SN}_1$ is the set of states reducing to no state, equivalently, the set of states $X \in \mathbb{S}$ which are pre-fixed points of r. Thus, we have to prove that if X is a pre-fixed point of r, then there is some open set $X \in O$ such that all $Y \in O$ are pre-fixed points of r. Let $O' = r^{-1}(\{r(X)\})$, $O'' = \{Y \in \mathbb{S} | (r(X) \cap Y = r(X)) \wedge (r(X) \setminus Y = \emptyset)\}$, and $O = O' \cap O''$. O' is open because $r : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$ is continuous and $\mathscr{P}_{fin}(\mathbb{A})$ has the discrete topology. O'' is open by point 2, with $I_0 = r(X)$ and $I_1 = \emptyset$. Thus, O is open. By definition, $X \in O' = r^{-1}(\{r(X)\})$ and $X \in O'' = \{Y \in \mathbb{S} | Y \cap r(X) = r(X) \wedge Y \setminus r(X) = \emptyset\}$, because $r(X) \subseteq X$. Thus, $X \in O$. For any $Y \in O$ we have by definition of O: $r(Y) = r(X)$ and $r(Y) = r(X) \subseteq Y$, as we wished to show.

5. We prove that $\mathtt{SN}_n$ is open by induction over $n \in \mathbb{N}, n > 0$. The case $n = 1$ is the previous point. Assume $\mathtt{SN}_n$ is open in order to prove that $\mathtt{SN}_{n+1}$ is open. Let $X \in \mathtt{SN}_{n+1}$: we have to prove that there is some open set $X \in O \subseteq \mathtt{SN}_{n+1}$. $r(X) \setminus X$ is finite, therefore there are finitely many homogeneous states $s_1, \ldots, s_k \subseteq r(X) \setminus X$. These states define exactly all reductions from X: $X \twoheadrightarrow_1^{s_i,r} X_i$, for $i = 1, \ldots, k$. From $X \in \mathtt{SN}_{n+1}$ we deduce $X_i \in \mathtt{SN}_n$ for all $i = 1, \ldots, k$. Let $O_i = \mathtt{R}_{s_i}^{-1}(\mathtt{SN}_n)$: O_i is open by point 3 above, and $X \in O_i$ because $\mathtt{R}_{s_i}(X) \in \mathtt{SN}_n$ by the assumption $X \in \mathtt{SN}_{n+1}$. Let $O' = r^{-1}(\{r(X)\})$, $O'' = \{Y \in \mathbb{S} | (r(X) \cap Y = r(X) \cap X) \wedge (r(X) \setminus Y = r(X) \setminus X)\}$. By definition we have $X \in O'$, $X \in O''$. O' is open because r is continuous, and O'' is open by point 2. Let $O = O' \cap O'' \cap O_1 \cap \ldots \cap O_n$: then $X \in O$ and O is open. For all $Y \in O$ we have $r(Y) = r(X)$, and $r(Y) \setminus Y = r(X) \setminus Y = r(X) \setminus X$. Therefore the reductions from Y are exactly in number of k: $Y \twoheadrightarrow_1^{s_i,r} Y_i$ for all $i = 1, \ldots, k$. We have $Y_i \in \mathtt{SN}_n$ by $O \subseteq O_i = \mathtt{R}_{s_i}^{-1}(\mathtt{SN}_n)$. We conclude that $Y \in \mathtt{SN}_{n+1}$, as wished.

6. $\mathtt{SN}$ is the union of all $\mathtt{SN}_n$, therefore is a union of open sets and it is open.

4 Reduction sequences of transfinite length

The next step is to prove that if there are states in $\mathbb{S} \setminus \mathtt{SN}$, then there are reduction sequences of any transfinite length. From this fact we will derive a contradiction.

We denote the class of ordinals with $\mathtt{ON}$, and ordinals with Greek letters $\alpha, \beta, \gamma, \lambda, \mu, \ldots$. We recall that a limit ordinal is any ordinal λ such that for all $\alpha < \lambda$ we have $\alpha + 1 < \lambda$. ω, the first infinite ordinal, and ω_1, the first uncountable ordinal, are limit. ω_1 has the additional property that any l.u.b. of some countable set I of ordinals all $< \omega_1$ is some $\xi < \omega_1$.

A sequence of length α on $\mathbb{S}$ is any map $\sigma : [0, \alpha[\to \mathbb{S}$. We represent sequences of length α with indexed sets $\sigma = \{X_\beta | \beta < \alpha\}$. When $\alpha =$ some limit ordinal λ, the limit of a sequence $\{X_\beta | \beta < \lambda\}$ is defined as $\lim_{\beta \to \lambda} X_\beta = \cup_{\beta < \lambda} \cap_{\beta \leq \gamma < \lambda} X_\gamma$. To put otherwise, $\lim_{\beta \to \lambda} X_\beta$ consists of all answers which belong to the states of $\{X_\beta | \beta < \lambda\}$ from some β on. A *limit sequence of length* α is any sequence $\{X_\beta | \beta < \alpha\}$ of length α such that for all limit ordinal $\lambda < \alpha$ we have $X_\lambda = \lim_{\beta \to \lambda} X_\beta$. A *limit reduction sequence of length* α is any limit sequence of length α such that for all $\beta + 1 < \alpha$ we have $X_\beta \twoheadrightarrow_1^r X_{\beta+1}$. We will prove that if $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is some limit reduction sequence of length ω_1 over $\mathbb{S} \setminus \mathtt{SN}$. Then we will prove that limit reduction sequence of length ω_1 over $\mathbb{S}$ (and with more reason, over $\mathbb{S} \setminus \mathtt{SN}$) cannot exists. The conclusion will be that $\mathbb{S} \setminus \mathtt{SN} = \emptyset$, as wished.

If $X \in \mathbb{S} \setminus \mathtt{SN}$, then there is some infinite reduction sequence

$$X = X_0 \twoheadrightarrow_1^r X_1 \twoheadrightarrow_1^r \ldots \twoheadrightarrow_1^r X_n \ldots$$

from X. Thus, there is some X_1 such that $X {\twoheadrightarrow}^{r}_{1} X_1$ and there is some infinite reduction sequence from X_1, hence $X {\twoheadrightarrow}^{r}_{1} X_1$ for some $X_1 \in \mathbb{S} \setminus \mathtt{SN}$. By choice axiom, there is some choice map

$$\mathtt{next} : (\mathbb{S} \setminus \mathtt{SN}) \to (\mathbb{S} \setminus \mathtt{SN})$$

such that $X {\twoheadrightarrow}^{r}_{1} \mathtt{next}(X)$ for all $X \in \mathbb{S}$. `next` is the empty map if $\mathtt{SN} = \mathbb{S}$. From now on, we assume to be fixed a choice map `next` as above.

Using `next`, from any $X \in \mathbb{S} \setminus \mathtt{SN}$ we may easily define an infinite reduction sequence $\mathtt{next}^n(X)$ all in $\mathbb{S} \setminus \mathtt{SN}$. We will prove that we may extend it to a limit reduction sequence on $\mathbb{S} \setminus \mathtt{SN}$ of length ω_1. This is because closed sets in the State Topology are closed by limit, and $\mathbb{S} \setminus \mathtt{SN}$ is a closed set.

In this part of the proof we need the notion of "definitively true".

Definition 4.1 (Definitively true) *Assume $\lambda \in \mathtt{ON}$ is limit and $\sigma = \{X_\beta | \beta < \lambda\}$ is any sequence of length λ.*

1. *σ satisfies $X_\gamma \subseteq X_{\gamma+1}$ definitively if $\exists \beta < \alpha . \forall \gamma \in [\beta, \lambda[. X_\gamma \subseteq X_{\gamma+1}$.*
2. *σ is definitively weakly increasing if $\exists \beta < \alpha . \forall \gamma, \delta \in [\beta, \lambda[. (\gamma \le \delta) \implies X_\gamma \subseteq X_\delta$.*
3. *σ is definitively constant if $\exists \beta < \alpha . \forall \gamma \in [\beta, \lambda[. X_\beta = X_\gamma$.*

The next step is to prove some easy properties of limit reduction sequences.

Lemma 4.2 (Limit Reduction sequences) *Assume $\lambda \in \mathtt{ON}$ is a limit ordinal and $\sigma = \{X_\alpha | \alpha < \lambda\}$ is any limit sequence on $\mathbb{S}$ of length λ. Let $L = \lim_{\gamma \to \omega_1} X_\gamma$.*

1. *If for some $\alpha < \lambda$ and all $\alpha \le \beta < \lambda$ we have $X_\alpha \subseteq X_\beta$, then $X_\alpha \subseteq L$.*
2. *If for some $\alpha < \lambda$ and all $\alpha \le \beta < \lambda$ we have $X_\beta \subseteq X_{\beta+1}$, then σ is weakly increasing from the same α.*
3. *If σ is definitively increasing and $\lambda = \omega_1$, then σ is definitively stationary.*
4. *For any $n \in \mathbb{N}$, $\sigma \restriction_{<n} = \{X_\alpha \restriction_{<n} | \alpha < \lambda\}$ is a limit sequence.*

Proof

1. Assume $X_\alpha \subseteq X_\gamma$ for all $\alpha \le \gamma < \lambda$. Then $X_\alpha \subseteq \bigcap_{\alpha \le \gamma < \lambda} X_\gamma \subseteq \lim X_{\gamma \to \lambda} X_\gamma = L$.
2. Assume $\alpha \le \alpha' < \lambda$. We prove $X_{\alpha'} \subseteq X_\beta$ by induction on $\alpha' \le \beta < \lambda$. Assume $\beta = \alpha'$. Then $X_{\alpha'} \subseteq X_{\alpha'}$. Assume $\beta = \gamma + 1 > \gamma \ge \alpha' \ge \alpha$. Then $X_{\alpha'} \subseteq X_\gamma$ by induction hypothesis and $X_\gamma \subseteq X_{\gamma+1}$ by hypothesis, hence $X_{\alpha'} \subseteq X_\beta$. Assume β is limit: then $X_{\alpha'} \subseteq X_\gamma$ for all $\alpha' \le \gamma < \beta$ by induction hypothesis, therefore $X_{\alpha'} \subseteq X_\beta$ by point 1 applied to the sequence $\{X_\gamma | \gamma < \beta\}$.
3. Assume that σ is definitively increasing from some α and $\lambda = \omega_1$, in order to prove that σ is definitively stationary. For all $a \in L$ we have $a \in X_\gamma$ definitively, therefore there is a first $\xi_a < \omega_1$ such that $a \in X_{\xi_a} \subseteq X_\gamma$ for all $\gamma \ge \xi_a$. Let ξ be l.u.b. of $\{\xi_a | a \in L\} \cup \{\alpha\}$. L is at most countable because $L \subseteq \mathbb{A}$, which is at most countable, and α and all ξ_a are $< \omega_1$, therefore $\xi < \omega_1$. We proved that there is some $\alpha \le \xi < \omega_1$ such that for all $\alpha \le \xi \le \gamma < \omega_1$ we have $L \subseteq X_\gamma$. From point 1 and $X_\gamma \subseteq X_\delta$ for all $\gamma \le \delta < \omega_1$ we have $X_\gamma \subseteq L$. We conclude $L = X_\gamma$ for all $\xi \le \gamma < \omega_1$.
4. Assume $\mu < \lambda$ is limit. Then $X_\mu = \bigcup_{\alpha < \mu} \bigcap_{\alpha \le \beta < \mu} X_\beta$, hence $X_\mu \restriction_{<n} = \bigcup_{\alpha < \mu} \bigcap_{\alpha \le \beta < \mu} X_\beta \restriction_{<n}$. Thus, $\sigma \restriction_{<n}$ is a limit sequence.

We explain now how to define a length ω_1 limit reduction sequence in $\mathbb{S} \setminus \mathtt{SN}$. The crucial remark is the following: for any answer in any element of a limit reduction sequence, either the answer belongs to the limit of the sequence together with all answers of level less or equal, or in some future step the is erased together with all answers of the same level (see the first point of the next Lemma).

Lemma 4.3 *Assume $\lambda \in \mathtt{ON}$ is a limit ordinal and $\sigma = \{X_\beta | \beta < \lambda\}$ is any limit reduction sequence of length λ. Let $L = \lim_{\beta \to \lambda} X_\beta \in \mathbb{S}$, and $n \in \mathbb{N}$*

1. *For all $\alpha < \lambda$ and all $n \in \mathbb{N}$, either $X_\alpha \restriction_{<n+1} \subseteq L$, or there is some $\alpha < \gamma < \lambda$ such that $X_\gamma \restriction_{=m} = \emptyset$*
2. *L is topologically adherent to $\{X_\beta | \beta < \lambda\}$ (that is, any open set including L intersects $\{X_\beta | \beta < \lambda\}$.*
3. *If $C \subseteq \mathbb{S}$ is closed and $\{X_\beta | \beta < \lambda\} \subseteq C$ then $L \in C$*
4. *$\mathbb{S} \setminus \mathtt{SN}$ is closed*
5. *If $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is some length ω_1 limit reduction sequence in $\mathbb{S} \setminus \mathtt{SN}$.*

Proof

1. Consider the sequence $\tau = \{X_\beta \restriction_{<n+1} | \beta < \lambda\}$: this is a limit sequence by Lemma 4.2.3. If $X_\beta \restriction_{<n+1} \subseteq X_{\beta+1} \restriction_{<n+1}$ for all $\alpha \leq \beta < \lambda$, then τ is weakly increasing from α by Lemma 4.2.1. In this case $X_\alpha \restriction_{<n+1} \subseteq X_\gamma \restriction_{<n+1} \subseteq X_\gamma$ for all $\alpha \leq \gamma < \lambda$, therefore $X_\alpha \restriction_{<n+1} \subseteq L$ by definition of L. Assume instead that $X_\beta \restriction_{<n+1} \not\subseteq X_{\beta+1} \restriction_{<n+1}$ for some $\alpha \leq \beta < \lambda$. Then by Lemma 3.1.4 we have $X_{\beta+1} \restriction_{=n} = \emptyset$.
2. Fix $\alpha < \lambda$, and assume O is any sub-basic open and $L \in O$, in order to prove that $X_\beta \in O$ for some $\alpha \leq \beta < \lambda$. For some $a \in \mathbb{A}$, either $O = A_a = \{X \in \mathbb{S} | a \in X\}$, or $O = B_a = \{X \in \mathbb{S} | X \cap [a]_\sim = \emptyset\}$. We reason by cases.
   (a) If $O = A_a$ we have $a \in L$. By definition of L, for some $\alpha < \lambda$ and all $\alpha \leq \beta < \lambda$ we have $a \in X_\beta$. In particular, $a \in X_\alpha$, hence $X_\alpha \in O$.
   (b) If $O = B_a$ we have $L \cap [a]_\sim = \emptyset$. Assume $n = \mathsf{lev}(a)$: by point 1 above there is some $\alpha \leq \beta < \lambda$ such that either $X_\beta \restriction_{<n+1} \subseteq L$ or $X_\beta \restriction_{=n} = \emptyset$. In both cases we have $X_\beta \restriction_{=n} \subseteq L \restriction_{=n}$, either because $X_\beta \restriction_{=n} = (X_\beta \restriction_{<n+1}) \restriction_{=n} \subseteq L \restriction_{=n}$, or because $X_\beta \restriction_{=n} = \emptyset \subseteq L \restriction_{=n}$. We deduce $X_\beta \cap [a]_\sim = (X_\beta \restriction_{=n}) \cap [a]_\sim \subseteq (L \restriction_{=n}) \cap [a]_\sim \subseteq L \cap [a]_\sim = \emptyset$. Thus, $X_\beta \in B_a$.
3. Assume $C \subseteq \mathbb{S}$ is closed and $\{X_\beta | \beta < \lambda\} \subseteq C$ in order to prove that $L \in C$. Assume for contradiction that $L \notin C$. Then $L \in \mathbb{S} \setminus C$, which is open. By the previous point we have $X_\alpha \in \mathbb{S} \setminus C$ for some $\alpha < \lambda$, contradicting $X_\alpha \in C$.
4. $\mathbb{S} \setminus \mathtt{SN}$ is closed because $\mathtt{SN}$ is open.
5. From any $X \in \mathbb{S} \setminus \mathtt{SN}$ we may define a limit reduction sequence of length ω_1 (and in fact of *any* length). We set $X_0 = X$, $X_{\alpha+1} = \mathtt{next}(X_\alpha)$ for all $\alpha < \omega_1$ and $X_\lambda = \lim_{\beta \to \lambda} X_\beta$ for all limit $\lambda < \omega_1$. We check that the definition is correct. By assumption $X_0 = X \in \mathbb{S} \setminus \mathtt{SN}$. Assume $\alpha < \omega_1$ and $X_\alpha \in \mathbb{S} \setminus \mathtt{SN}$, then $X_{\alpha+1} = \mathtt{next}(X_\alpha) \in \mathbb{S} \setminus \mathtt{SN}$. Assume $\lambda < \omega_1$ is limit and $\{X_\beta | \beta < \lambda\} \subseteq \mathbb{S} \setminus \mathtt{SN}$. Since $\mathbb{S} \setminus \mathtt{SN}$ is closed by point 3, then by point 2 above we have $X_\lambda = \lim_{\beta \to \lambda} X_\beta \in \mathbb{S} \setminus \mathtt{SN}$.

5 A termination result from an algorithm searching fixed points

In the previous section we proved that if $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is a limit reduction sequence of length ω. In this section we will prove that no limit reduction sequence of length ω_1 may exists, and we will conclude that $\mathbb{S} \setminus \mathtt{SN} = \emptyset$, as wished. We first prove that limit reduction sequences of length ω_1 are definitively stationary.

Lemma 5.1 (Stationarity) *Assume $\sigma = \{X_\alpha | \alpha < \omega_1\}$ is any sequence on $\mathbb{S}$. Let $n \in \mathbb{N}$.*

1. *For any limit reduction sequence $\{X_\beta | \beta < \omega_1\}$ of length ω_1 on $\mathbb{S}$, the sequence $\{X_\beta \upharpoonright_{<n} | \beta < \omega_1\}$ is definitively stationary.*
2. *Any limit reduction sequence $\{X_\beta | \beta < \omega_1\}$ of length ω_1 on $\mathbb{S}$ is definitively stationary.*

Proof

1. We argue by induction on $n \in \mathbb{N}$. Assume $n = 0$: then $X_\beta \upharpoonright_{<n} = \emptyset$ is definitively stationary. Assume $X_\beta \upharpoonright_{<n}$ is definitively stationary, in order to prove $X_\beta \upharpoonright_{<n+1}$ is definitively stationary. If $X_\beta \upharpoonright_{<n} = X_{\beta+1} \upharpoonright_{<n}$ then $X_\beta \upharpoonright_{<n+1} \subseteq X_{\beta+1} \upharpoonright_{<n}$ by Lemma 3.1.6, hence we definitively have $X_\beta \upharpoonright_{<n+1} \subseteq X_{\beta+1} \upharpoonright_{<n}$. By Lemma 4.2.4 $X_\beta \upharpoonright_{<n+1}$ is a limit sequence, and by Lemma 4.2.2 it is weakly increasing. It has length ω_1, therefore by Lemma 4.2.3 it is definitively stationary.
2. By the previous point, for all $n \in \mathbb{N}$ there is a first $\alpha_n < \omega_1$ such that $X_\beta \upharpoonright_{<n}$ is stationary from α_n. Let $\alpha < \omega_1$ by the l.u.b. of $\{\alpha_n | n \in \mathbb{N}\}$: then for all $n \in \mathbb{N}$, $X_\beta \upharpoonright_{<n}$ is stationary from α. Thus, X_β is stationary from α.

The strong termination result for the reduction relation $\twoheadrightarrow^r_1$ easily follows.

Theorem 5.2 (Pre-fixed point Theorem) *For all states $X \in \mathbb{S}$, for all realizers $r : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$, all reduction sequences $X \twoheadrightarrow^r_1 X_1 \twoheadrightarrow^r_1 \ldots \ldots \twoheadrightarrow^r_1 X_n \twoheadrightarrow^r_1 \ldots$ from X are finite.*

Proof Assume there is some $X \in \mathbb{S} \setminus \mathtt{SN}$. By Lemma 5.1.5 there is some limit reduction sequence $\{X_\beta | \beta < \omega_1\} \subseteq (\mathbb{S} \setminus \mathtt{SN})$ from X of length ω_1. By Lemma 5.1.2, $\{X_\beta | \beta < \omega_1\}$ is definitively stationary, therefore for some $\alpha < \omega_1$ we have $X_{\alpha+1} = X_\alpha$, hence $X_\alpha \twoheadrightarrow^r_1 X_{\alpha+1} = X_\alpha$, against Lemma 3.1.2.

6 Related works and conclusions

In this section we stress the most relevant differences of the present work w.r.t. the ones by the authors themselves and by others. The essential difference w.r.t. [2] and [4] is non-monotonicity. In[1] also the case of non-monotonic learning is considered, though only deterministic learning processes are treated. In [6], which is the full version of [5], we propose a deterministic algorithm to compute a (finite) sound pre-fixed point of any effective realizer; however we have been able to treat the case in which the maximum level of answers is 2, while here we have a termination proof of a non-deterministic algorithm working on states with answers of arbitrary level $< \omega$.

We stress that non-determinism is no minor trick. First, if the output $r(X)$ of a realizer may include more than one answer, then our convergence result also holds for any $r' : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$, *even if not continuous*, provided there is some continuous $r : \mathbb{S} \to \mathscr{P}_{fin}(\mathbb{A})$ such that $r'(X) \subseteq r(X)$ for all $X \in \mathbb{S}$. This simple remark shows that the result for the non-deterministic case is much stronger than the result for the deterministic one.

In [8] Mints considered the ω-level version of the problem. In our terminology, he introduced a non-deterministic reduction relation adding one answer at the time, and proved a *weak* normalization result: there is a reduction sequence from the empty state to some sound irreducible state. However, in [8] there is no normalizing reduction strategy, and we suspect that the strong normalization result would fail in that setting.

In conclusion we have presented a new result that we consider as a step toward a realistic use of non constructive proofs as algorithms. Improvements are certainly possible, such as for example a more sophisticated way of representing logical dependencies than level. The aim is to find an algorithm removing the minimum amount of answers from a state when adding new ones, hence resulting into a faster computation.

References

[1] F. Aschieri. *Learning, Realizability and Games in Classical Arithmetic*, Ph. d. thesis, University of Turin and Queen Mary University of London, 2010. `http://www.di.unito.it/~stefano/Aschieri-PhDThesis.pdf`

[2] F. Aschieri, S. Berardi, *Interactive Learning-Based Realizability Interpretation for Heyting Arithmetic with* $\mathtt{EM}_1$, Proceedings of TLCA 2009, Springer Lecture Notes in Computer Science, vol. 5608, 2009. doi:10.1007/978-3-642-02273-9_4

[3] S. Berardi, U. de'Liguoro, *Toward the interpretation of non-constructive reasoning as non-monotonic learning*, Information and Computation, vol. 207, 1, pag. 63-81, (2009). doi:10.1016/j.ic.2008.10.003

[4] S. Berardi, U. de'Liguoro, *Interactive realizers. A new approach to program extraction from non constructive proofs*, ACM Transactions on Computational Logic, vol. 13 n. 2, 2012. doi:10.1145/2159531.2159533

[5] S. Berardi, U. de'Liguoro, *Knowledge Spaces and Interactive Realizers*. Proc. of CSL 2012, LIPICs vol. 16, pag. 77-91, 2012. doi:10.4230/LIPIcs.CSL.2012.77

[6] S. Berardi, U. de'Liguoro, *Knowledge Spaces and the Completeness of Learning Strategies*. Submitted for publication, 2013. Draft version vailable from `http://www.di.unito.it/~deligu/papers/BdL_Knowledge.pdf`

[7] T. Coquand. *A semantics of evidence for classical arithmetic.* J. Symb. Log., 60:325–337, 1995.doi:10.2307/2275524.

[8] G. Mints. *Non-Deterministic Epsilon Substitution Method for PA and* ID_1, Logic, Construction, Computation, Ontos-Verlag Series in Mathematical Logic, Berger et al. editors, 2012.

From Branching to Linear Time, Coalgebraically

Corina Cîrstea
University of Southampton
cc2@ecs.soton.ac.uk

We consider state-based systems modelled as coalgebras whose type incorporates branching, and show that by suitably adapting the definition of coalgebraic bisimulation, one obtains a general and uniform account of the linear-time behaviour of a state in such a coalgebra. By moving away from a boolean universe of truth values, our approach can measure the extent to which a state in a system with branching is able to exhibit a particular linear-time behaviour. This instantiates to measuring the probability of a specific behaviour occurring in a probabilistic system, or measuring the minimal cost of exhibiting a specific behaviour in the case of weighted computations.

1 Introduction

When analysing process behaviour, one of the early choices one has to make is between a linear and a branching view of time. In branching-time semantics, the choices a process has for proceeding from a particular state are taken into account when defining a notion of process equivalence (with bisimulation being the typical such equivalence), whereas in linear-time semantics such choices are abstracted away and the emphasis is on the individual executions that a process is able to exhibit. From a system verification perspective, one often chooses the linear-time view, as this not only leads to simpler specification logics and associated verification techniques, but also meets the practical need to verify all possible system executions.

While the theory of coalgebras has, from the outset, been able to provide a uniform account of various bisimulation-like observational equivalences (and later, of various simulation-like behavioural preorders), it has so far not been equally successful in giving a generic account of the linear-time behaviour of a state in a system whose type incorporates a notion of branching. For example, the generic trace theory of [9] only applies to systems modelled as coalgebras of type $\mathsf{T} \circ F$, with the monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ specifying a branching type (e.g. non-deterministic or probabilistic), and the endofunctor $F : \mathsf{Set} \to \mathsf{Set}$ defining the structure of individual transitions (e.g. labelled transitions or successful termination). The approach in loc. cit. is complemented by that of [12], where traces are derived using a determinisation procedure similar to the one for non-deterministic automata. The latter approach applies to systems modelled as coalgebras of type $G \circ \mathsf{T}$, where again a monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ is used to model branching behaviour, and an endofunctor G specifies the transition structure. Neither of these approaches is able to account for potentially infinite traces, as typically employed in model-based formal verification. This limitation is partly addressed in [1], but again, this only applies to coalgebras of type $\mathsf{T} \circ F$, albeit with more flexibility in the underlying category (which in particular allows a measure-theoretic account of infinite traces in probabilistic systems). Finally, none of the above-mentioned approaches exploits the compositionality that is intrinsic to the coalgebraic approach. In particular, coalgebras of type $G \circ \mathsf{T} \circ F$ (of which systems with both inputs and outputs are an example, see Example 5.7) can not be accounted for by any of the existing approaches. This paper presents an attempt to address the above limitations concerning the types of coalgebras and the nature of traces that can be accounted for, by providing a *uniform* and *compositional* treatment of (possibly infinite) linear-time behaviour in systems with branching.

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 11–27, doi:10.4204/EPTCS.126.2

In our view, one of the reasons for only a partial success in developing a fully general coalgebraic theory of traces is the long-term aspiration within the coalgebra community to obtain a uniform characterisation of trace equivalence via a finality argument, in much the same way as is done for bisimulation (in the presence of a final coalgebra). This encountered difficulties, as a suitable category for carrying out such an argument proved difficult to find in the general case. In this paper, we tackle the problem of getting a handle on the linear-time behaviour of a state in a coalgebra with branching from a different angle: we do not attempt to directly define a notion of trace equivalence between two states (e.g. via finality in some category), but focus on *testing* whether a state is able to exhibit a particular trace, and on measuring the extent of this ability. This ”measuring” relates to the type of branching present in the system, and instantiates to familiar concepts such as the probability of exhibiting a given trace in probabilistic systems, the minimal cost of exhibiting a given trace in weighted computations, and simply the ability to exhibit a trace in non-deterministic systems.

The technical tool for achieving this goal is a generalisation of the notions of relation and relation lifting [10], which lie at the heart of the definition of coalgebraic bisimulation. Specifically, we employ relations valued in a partial semiring, and a corresponding generalised version of relation lifting. Our approach applies to coalgebras whose type is obtained as the composition of several endofunctors on Set: one of these is a monad T that accounts for the presence of branching in the system, while the remaining endofunctors, assumed here to be polynomial, jointly determine the notion of linear-time behaviour. This strictly subsumes the types of systems considered in earlier work on coalgebraic traces [9, 1, 12], while also providing compositionality in the system type.

Our main contribution, presented in Section 5, is a *uniform* and *compositional* account of linear-time behaviour in state-based systems with branching. A by-product of our work is an extension of the study of additive monads carried out in [14, 3] to what we call *partially additive monads* (Section 3). Our approach can be summarised as follows:

- We move from two-valued to multi-valued relations, with the universe of truth values being induced by the choice of monad for modelling branching. This instantiates to relations valued in the interval $[0,1]$ in the case of probabilistic branching, the set $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ in the case of weighted computations, and simply $\{\bot, \top\}$ in the case of non-deterministic branching. This reflects our view that the notion of truth used to reason about the observable behaviour of a system should be dependent on the branching behaviour present in that system. Such a dependency is also expected to result in temporal logics that are more natural and more expressive, and at the same time have a conceptually simpler semantics. In deriving a suitable structure on the universe of truth values, we generalise results on additive monads [14, 3] to *partially additive monads*. This allows us to incorporate probabilistic branching under our approach. We show that for a commutative, partially additive monad T on Set, the set $\mathsf{T}1$ carries a partial semiring structure with an induced preorder, which in turn makes $\mathsf{T}1$ an appropriate choice of universe of truth values.

- We generalise and adapt the notion of relation lifting used in the definition of coalgebraic bisimulation, in order to (i) support multi-valued relations, and (ii) abstract away branching. Specifically, we make use of the partial semiring structure carried by the universe of truth values to generalise relation lifting of polynomial endofunctors to multi-valued relations, and employ a canonical *extension lifting* induced by the monad T to capture a move from branching to linear time. The use of this extension lifting allows us to make formal the idea of testing whether, and to what extent, a state in a coalgebra with branching can exhibit a particular *linear-time* behaviour. Our approach resembles the idea employed by partition refinement algorithms for computing bisimulation on labelled transition systems with finite state spaces [13]. There, one starts from a single partition of the state space, with all states related to each other, and repeatedly refines it through stepwise unfolding of the transition structure, until a fixpoint is reached. Similarly, we start by assuming that a state in a system with branching can exhibit any linear-time behaviour, and moreover, assign the maximum possible value to each pair consisting of a state and a linear-time behaviour. We then repeatedly refine the values associated to such pairs, through stepwise unfolding of the coalgebraic structure.

The present work is closely related to our earlier work on maximal traces and path-based logics [1], which described a game-theoretic approach to testing if a system with non-deterministic branching is able to exhibit a particular trace. Here we consider arbitrary branching types, and while we do not emphasise the game-theoretic aspect, our use of greatest fixpoints has a very similar thrust.

Acknowledgements Several fruitful discussions with participants at the 2012 Dagstuhl Seminar on Coalgebraic Logics helped refine the ideas presented here. Our use of relation lifting was inspired by the recent work on coinductive predicates [8], itself based on the seminal work in [10] on the use of predicate and relation lifting in the formalisation of induction and coinduction principles. Last but not least, the comments received from the anonymous reviewers contributed to improving the presentation of this work and to identifying new directions for future work.

2 Preliminaries

2.1 Relation Lifting

The concepts of *predicate lifting* and *relation lifting*, to our knowledge first introduced in [10], are by now standard tools in the study of coalgebraic models, used e.g. to provide an alternative definition of the notion of bisimulation (see e.g. in [11]), or to describe the semantics of coalgebraic modal logics [17, 16]. While these concepts are very general, their use so far usually restricts this generality by viewing both predicates and relations as sub-objects in some category (possibly carrying additional structure). In this paper, we make use of the full generality of these concepts, and move from the standard view of relations as subsets to a setting where relations are valuations into a universe of truth values. This section recalls the definition of relation lifting in the standard setting where relations are given by monomorphic spans.

Throughout this section (only), Rel denotes the category whose objects are binary relations $(R, \langle r_1, r_2\rangle)$ with $\langle r_1, r_2\rangle : R \to X \times Y$ a monomorphic span, and whose arrows from $(R, \langle r_1, r_2\rangle)$ to $(R', \langle r'_1, r'_2\rangle)$ are given by pairs of functions $(f : X \to X', g : Y \to Y')$ s.t. $(f \times g) \circ \langle r_1, r_2\rangle$ factors through $\langle r'_1, r'_2\rangle$:

$$\begin{tikzcd}
R \arrow[r, tail, "\langle r_1,r_2\rangle"] \arrow[d, dashed] & X \times Y \arrow[d, "f\times g"] \\
R' \arrow[r, tail, "\langle r'_1,r'_2\rangle"] & X' \times Y'
\end{tikzcd}$$

In this setting, the *relation lifting of a functor* $F : \mathsf{Set} \to \mathsf{Set}$ is defined as a functor $\mathsf{Rel}(F) : \mathsf{Rel} \to \mathsf{Rel}$ taking a relation $\langle r_1, r_2\rangle : R \to X \times Y$ to the relation defined by the span $\langle F(r_1), F(r_2)\rangle : F(R) \to F(X) \times F(Y)$, obtained via the unique epi-mono factorisation of $\langle F(r_1), F(r_2)\rangle$:

$$\begin{tikzcd}
R \arrow[d, tail, "\langle r_1,r_2\rangle"'] & F(R) \arrow[r, two heads] \arrow[d, "\langle F(r_1),F(r_2)\rangle"'] & \mathsf{Rel}(F)(R) \arrow[dl, tail] \\
X \times Y & F(X) \times F(Y) &
\end{tikzcd}$$

It follows easily that this construction is functorial, and in particular preserves the order $\leq$ between relations on the same objects given by $(R,\langle r_1,r_2\rangle) \leq (S,\langle s_1,s_2\rangle)$ if and only if $\langle r_1,r_2\rangle$ factors through $\langle s_1,s_2\rangle$:

$$\begin{tikzcd}
R \arrow[r, tail, dashed] \arrow[rr, tail, bend right, "\langle r_1,r_2\rangle"'] & S \arrow[r, tail, "\langle s_1,s_2\rangle"] & X \times Y
\end{tikzcd}$$

An alternative definition of $\mathsf{Rel}(F)$ for F a *polynomial functor* (i.e. constructed from the identity and constant functors using *finite* products and set-indexed coproducts) can be given by induction on the structure of F. We refer the reader to [11, Section 3.1] for details of this definition. An extension of this definition to a more general notion of relation will be given in Section 4.

2.2 Coalgebras

We model state-based, dynamical systems as coalgebras over the category of sets. Given a functor $F : \mathbb{C} \to \mathbb{C}$ on an arbitrary category, an *F-coalgebra* is given by a pair (C,γ) with C an object of $\mathbb{C}$, used to model the state space, and $\gamma : C \to FC$ a morphism in $\mathbb{C}$, describing the one-step evolution of the system states. Then, a canonical notion of observational equivalence between the states of two F-coalgebras is provided by the notion of bisimulation. Of the many, and under the assumption that F preserves weak pullbacks, equivalent definitions of bisimulation (see [11] for a detailed account), we recall the one based on relation lifting. This applies to coalgebras over the category of sets (as described below), but also more generally to categories with logical factorisation systems (as described in [11]). According to this definition, an *F-bisimulation* between coalgebras (C,γ) and (D,δ) over Set is a $\mathsf{Rel}(F)$-coalgebra:

$$\begin{tikzcd}
R \arrow[r, dashed] \arrow[d, tail] & \mathsf{Rel}(F)(R) \arrow[d, tail] \\
X \times Y \arrow[r, "\gamma\times\delta"'] & F(X) \times F(Y)
\end{tikzcd}$$

In the remainder of this section we sketch a coalgebraic generalisation of a well-known partition refinement algorithm for computing *bisimilarity* (i.e. the largest bisimulation) on finite-state labelled transition systems [13]. For an arbitrary endofunctor $F : \mathsf{Set} \to \mathsf{Set}$ and two finite-state F-coalgebras (C,γ) and (D,δ), the generalised algorithm iteratively computes relations $\simeq_i \subseteq C \times D$ with $i = 0,1,\ldots$ as follows:

- $\sim_0 = C \times D$
- $\sim_{i+1} = (\gamma \times \delta)^*(\mathsf{Rel}(F)(\simeq_i))$ for $i = 0,1,\ldots$

where $(\gamma \times \delta)^*$ takes a relation $R \subseteq FC \times FD$ to the relation $\{(c,d) \in C \times D \mid (\gamma(c),\delta(d)) \in R\}$. Thus, in the initial approximation $\simeq_0$ of the bisimilarity relation, all states are related, whereas at step $i+1$ two states are related if and only if their one-step observations are suitably related using the relation $\simeq_i$. Bisimilarity between the coalgebras (C,γ) and (D,δ) thus arises as the greatest fixpoint of a monotone operator on the complete lattice of relations between C and D, which takes a relation $R \subseteq C \times D$ to the relation $(\gamma \times \delta)^*(\mathsf{Rel}(F)(R))$. A similar characterisation of bisimilarity exists for coalgebras with infinite state spaces, but in this case the fixpoint can not, in general, be reached in a finite number of steps.

The above greatest fixpoint characterisation of bisimilarity is generalised and adapted in Section 5, in order to characterise the extent to which a state in a coalgebra with branching can exhibit a linear-time behaviour. There, the two coalgebras in question have different types: the former has branching behaviour and is used to model the system of interest, whereas the latter has linear behaviour only and describes the domain of possible traces.

2.3 Monads

In what follows, we use monads (T,η,μ) on Set (where $\eta : \mathsf{Id} \Rightarrow \mathsf{T}$ and $\mu : \mathsf{T}\circ\mathsf{T} \Rightarrow \mathsf{T}$ are the *unit* and *multiplication* of T) to capture branching in coalgebraic types. Moreover, we assume that these monads are *strong* and *commutative*, i.e. they come equipped with a *strength map* $\mathsf{st}_{X,Y} : X \times \mathsf{T}Y \to \mathsf{T}(X\times Y)$ as well as a *double strength map* $\mathsf{dst}_{X,Y} : \mathsf{T}X \times \mathsf{T}Y \to \mathsf{T}(X\times Y)$ for each choice of sets X,Y; these maps are natural in X and Y, and satisfy coherence conditions w.r.t. the unit and multiplication of T. We also make direct use of the *swapped strength map* $\mathsf{st}'_{X,Y} : \mathsf{T}X \times Y \to \mathsf{T}(X\times Y)$, obtained from the strength via the *twist map* $\mathsf{tw}_{X,Y} : X\times Y \to Y \times X$:

$$\begin{tikzcd}
\mathsf{T}X \times Y \arrow[r, "\mathsf{tw}_{\mathsf{T}X,Y}"] & Y \times \mathsf{T}X \arrow[r, "\mathsf{st}_{Y,X}"] & \mathsf{T}(Y\times X) \arrow[r, "\mathsf{T}\mathsf{tw}_{Y,X}"] & \mathsf{T}(X\times Y)
\end{tikzcd}$$

Example 2.1. As examples of monads, we consider:

1. the *powerset monad* $\mathscr{P} : \mathsf{Set} \to \mathsf{Set}$, modelling nondeterministic computations, with unit given by singletons and multiplication given by unions. Its strength and double strength are given by

$$\mathsf{st}_{X,Y}(x,V) = \{x\}\times V \qquad\qquad \mathsf{dst}_{X,Y}(U,V) = U\times V$$

   for $x\in X$, $U\in\mathscr{P}X$ and $V\in\mathscr{P}Y$,

2. the *semiring monad* $\mathsf{T}_S : \mathsf{Set} \to \mathsf{Set}$ with $(S,+,0,\bullet,1)$ a semiring, given by

$$\mathsf{T}_S(X) = \{f : X \to S \mid \mathsf{sup}(f) \text{ is finite}\}$$

   with $\mathsf{sup}(f) = \{x\in X \mid f(x)\neq 0\}$ the *support* of f. Its unit and multiplication are given by

$$\eta_X(x)(y) = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise}\end{cases} \qquad\qquad \mu_X(f\in S^{(S^X)}) = \sum_{g\in\mathsf{sup}(f)}\sum_{x\in\mathsf{sup}(g)} f(g)\bullet g(x)$$

   while its strength and double strength are given by

$$\mathsf{st}_{X,Y}(x,g)(z,y) = \begin{cases} g(y) & \text{if } z = x \\ 0 & \text{otherwise}\end{cases} \qquad\qquad \mathsf{dst}_{X,Y}(f,g)(z,y) = f(z)\bullet g(y)$$

   for $x\in X$, $f\in\mathsf{T}_S(X)$, $g\in\mathsf{T}_S(Y)$, $z\in X$ and $y\in Y$. As a concrete example, we will consider the semiring $W = (\mathbb{N}^\infty,\min,\infty,+,0)$, and use T_W to model weighted computations.

3. the *sub-probability distribution monad* $\mathscr{S} : \mathsf{Set} \to \mathsf{Set}$, modelling probabilistic computations, with unit given by the Dirac distributions (i.e. $\eta_X(x) = (x\mapsto 1)$), and multiplication given by $\mu_X(\Phi) = \sum\limits_{\varphi\in\mathsf{sup}(\Phi)}\sum\limits_{x\in\mathsf{sup}(\varphi)}\Phi(\varphi)*\varphi(x)$, with $*$ denoting multiplication on $[0,1]$. Its strength and double strength are given by

$$\mathsf{st}_{X,Y}(x,\psi)(z,y) = \begin{cases} \psi(y) & \text{if } z = x \\ 0 & \text{otherwise}\end{cases} \qquad\qquad \mathsf{dst}_{X,Y}(\varphi,\psi)(z,y) = \varphi(z)*\psi(y)$$

   for $x\in X$, $\varphi\in\mathscr{S}(X)$, $\psi\in\mathscr{S}(Y)$, $z\in X$ and $y\in Y$.

3 From Partially Additive, Commutative Monads to Partial Commutative Semirings with Order

Later in this paper we will consider coalgebras whose type is given by the composition of several endofunctors on Set, one of which is a commutative monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ accounting for the presence of branching in the systems of interest. This section extends results in [14, 3] to show how to derive a universe of truth values from such a monad. The assumption of loc. cit. concerning the *additivity* of the monad under consideration is here weakened to *partial additivity* (see Definition 3.1); this allows us to incorporate the sub-probability distribution monad (which is not additive) into our framework. Specifically, we show that any commutative, partially additive monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ induces a partial commutative semiring structure on the set $\mathsf{T}1$, with $1 = \{*\}$ a final object in Set. We recall that a *commutative semiring* consists of a set S carrying two commutative monoid structures $(+,0)$ and $(\bullet,1)$, with the latter distributing over the former: $s \bullet 0 = 0$ and $s \bullet (t+u) = s \bullet t + s \bullet u$ for all $s,t,u \in S$. A *partial commutative semiring* is defined similarly, except that $+$ is a partial operation subject to the condition that whenever $t+u$ is defined, so is $s \bullet t + s \bullet u$, and moreover $s \bullet (t+u) = s \bullet t + s \bullet u$. The relevance of a partial commutative semiring structure on the set of truth values will become clear in Sections 4 and 5.

It follows from results in [3] that any commutative monad (T, η, μ) on Set induces a commutative monoid $(\mathsf{T}(1), \bullet, \eta_1(*))$, with multiplication $\bullet : \mathsf{T}(1) \times \mathsf{T}(1) \to \mathsf{T}(1)$ given by the composition

$$\mathsf{T}(1) \times \mathsf{T}(1) \xrightarrow{\mathsf{dst}_{1,1}} \mathsf{T}(1 \times 1) \xrightarrow{\mathsf{T}\pi_2} \mathsf{T}(1)$$

Alternatively, this multiplication can be defined as the composition

$$\mathsf{T}(1) \times \mathsf{T}(1) \xrightarrow{\mathsf{st}'_{1,1}} \mathsf{T}(1 \times \mathsf{T}(1)) \xrightarrow{T\pi_2} \mathsf{T}^2(1) \xrightarrow{\mu_1} \mathsf{T}(1)$$

or as

$$\mathsf{T}(1) \times \mathsf{T}(1) \xrightarrow{\mathsf{st}_{1,1}} \mathsf{T}(\mathsf{T}(1) \times 1) \xrightarrow{T\pi_1} \mathsf{T}^2(1) \xrightarrow{\mu_1} \mathsf{T}(1)$$

(While the previous two definitions coincide for commutative monads, this is not the case in general.)

Remark 3.1. The following maps define left and right actions of $(\mathsf{T}(1), \bullet)$ on $\mathsf{T}(X)$:

$$\mathsf{T}(1) \times \mathsf{T}(X) \xrightarrow{\mathsf{dst}_{1,X}} \mathsf{T}(1 \times X) \xrightarrow{\mathsf{T}\pi_2} \mathsf{T}(X) \qquad \mathsf{T}(X) \times \mathsf{T}(1) \xrightarrow{\mathsf{dst}_{X,1}} \mathsf{T}(X \times 1) \xrightarrow{\mathsf{T}\pi_1} \mathsf{T}(X)$$

On the other hand, any monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ with $\mathsf{T}\emptyset = 1$ is such that, for any X, $\mathsf{T}X$ has a *zero element* $0 \in \mathsf{T}X$, obtained as $(\mathsf{T}!_X)(*)$. This yields a *zero map* $0 : Y \to \mathsf{T}X$ for any X, Y, obtained as the composition

$$Y \xrightarrow{!_Y} \mathsf{T}\emptyset \xrightarrow{T!_X} \mathsf{T}X$$

with the maps $!_Y : Y \to \mathsf{T}\emptyset$ and $!_X : \emptyset \to X$ arising by finality and initiality, respectively. Now consider the following map:

$$T(X+Y) \xrightarrow{\langle \mu_X \circ \mathsf{T}p_1, \mu_Y \circ \mathsf{T}p_2 \rangle} \mathsf{T}X \times \mathsf{T}Y \tag{1}$$

where $p_1 = [\eta_X, 0] : X + Y \to \mathsf{T}X$ and $p_2 = [0, \eta_Y] : X + Y \to \mathsf{T}Y$.

Definition 3.1. *A monad* $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ *is called* additive[1] *(*partially additive*) if* $\mathsf{T}\emptyset = 1$ *and the map in (1) is an isomorphism (respectively monomorphism).*

[1] Additive monads were studied in [14, 3].

The (partial) inverse of the map $\langle \mu_X \circ \mathsf{T}p_1, \mu_Y \circ \mathsf{T}p_2 \rangle$ can be used to define a (partial) addition on the set $\mathsf{T}X$, given by $\mathsf{T}[1_X, 1_X] \circ q_{X,X}$, where $q_{X,X} : \mathsf{T}X \times \mathsf{T}X \to \mathsf{T}(X+X)$ is the (partial) left inverse of $\langle \mu_X \circ \mathsf{T}p_1, \mu_Y \circ \mathsf{T}p_2 \rangle$:

$$\begin{tikzcd}
\mathsf{T}X & \mathsf{T}(X+X) \arrow[l, "{\mathsf{T}[1_X,1_X]}"'] \arrow[r, shift left, "{\langle \mu_X\circ\mathsf{T}p_1, \mu_Y\circ\mathsf{T}p_2\rangle}"] & \mathsf{T}X\times\mathsf{T}X \arrow[l, shift left, dashed, "q_{X,X}"] \arrow[ll, bend left, "+"]
\end{tikzcd}$$

That is, $a+b$ is defined if and only if $(a,b) \in \mathsf{Im}(\langle \mu_X \circ \mathsf{T}p_1, \mu_Y \circ \mathsf{T}p_2 \rangle)$ [2].

[3, Section 5.2] explores the connection between additive, commutative monads and commutative semirings. The next result provides a generalisation to partially additive, commutative monads and partial commutative semirings.

The proof of Proposition 3.2 is a slight adaptation of the corresponding proofs in [3, Section 5.2].

Proposition 3.2. *Let T be a commutative, (partially) additive monad. Then:*

1. $(\mathsf{T}1, \bullet, \eta_1(*))$ *is a commutative monoid.*
2. $(\mathsf{T}X, 0, +)$ *is a (partial) commutative monoid, for each set X.*
3. $(\mathsf{T}1, 0, +, \bullet, \eta_1(*))$ *is a (partial) commutative semiring.*

Proof (Sketch). The commutativity of the following diagram lies at the heart of the proof of item 3:

$$\begin{tikzcd}
\mathsf{T}1\times\mathsf{T}1 \arrow[dd, "\bullet"'] & \mathsf{T}(1+1)\times\mathsf{T}1 \arrow[l, "{\mathsf{T}[1_X,1_X]\times 1_{\mathsf{T}1}}"'] \arrow[r, shift left, "\delta\times 1_{\mathsf{T}1}"] \arrow[dd, "a_{\mathsf{T}(1+1)}"'] & (\mathsf{T}1\times\mathsf{T}1)\times\mathsf{T}1 \arrow[l, shift left, dashed, "q_{1,1}\times 1_{\mathsf{T}1}"] \arrow[d, "{\langle \pi_1\times\pi_2, \pi_2\times\pi_2\rangle}"] \\
 & & (\mathsf{T}1\times\mathsf{T}1)\times(\mathsf{T}1\times\mathsf{T}1) \arrow[d, "\bullet\times\bullet"] \\
\mathsf{T}1 & \mathsf{T}(1+1) \arrow[l, "{\mathsf{T}[1_X,1_X]}"] \arrow[r, shift left, "\delta"] & \mathsf{T}1\times\mathsf{T}1 \arrow[l, shift left, dashed, "q_{1,1}"]
\end{tikzcd}$$

where $a_{\mathsf{T}X} : \mathsf{T}X \times \mathsf{T}1 \to \mathsf{T}X$ is the right action from Remark 3.1, and δ is the map $\langle \mu_1 \circ \mathsf{T}p_1, \mu_1 \circ \mathsf{T}p_2 \rangle$ used in the definition of $+$ on $\mathsf{T}1$. The composition $\bullet \circ (\mathsf{T}[1_X, 1_X] \times 1_{\mathsf{T}1}) \circ (q_{1,1} \times 1_{\mathsf{T}1})$ captures the computation of $(a+b) \bullet c$, whereas the composition $\mathsf{T}[1_X, 1_X] \circ q_{1,1} \circ (\bullet \times \bullet) \circ \langle \pi_1 \times \pi_2, \pi_2 \times \pi_2 \rangle$ captures the computation $a \bullet c + b \bullet c$, with $a, b, c \in \mathsf{T}1$. The fact that δ commutes with the strength map (by (iv) of [3, Lemma 15]), together with $a_{\mathsf{T}(1+1)}$ and $\bullet$ being essentially given by the double strength maps $\mathsf{dst}_{1+1,1}$ and $\mathsf{dst}_{1,1}$, yields $(\bullet \times \bullet) \circ \langle \pi_1 \times \pi_2, \pi_2 \times \pi_2 \rangle \circ (\delta \times 1_{\mathsf{T}1}) = \delta \circ a_{\mathsf{T}(1+1)}$, that is, commutativity (via the plain arrows) of the right side of the above diagram. This immediately results in $a \bullet c + b \bullet c$ being defined whenever $a+b$ is defined, and hence in the commutativity of the right side of the diagram also via the dashed arrows. This, combined with the commutativity of the left side of the diagram (which is simply naturality of the right action a), gives $(a+b) \bullet c = a \bullet c + b \bullet c$ whenever $a+b$ is defined. □

Example 3.2. For the monads in Example 2.1, one obtains the commutative semirings $(\{\bot, \top\}, \vee, \bot, \wedge, \top)$ when $\mathsf{T} = \mathscr{P}$, $(\mathbb{N}^\infty, \min, \infty, +, 0)$ when $\mathsf{T} = \mathsf{T}_W$ [3], and the partial commutative semiring $([0,1], +, 0, *, 1)$ when $\mathsf{T} = \mathscr{S}$ (where in the latter case $a+b$ is defined if and only if $a+b \leq 1$).

[2] A similar, but *total*, addition operation is defined in [14, 3] for additive monads.

[3] This is sometimes called the *tropical semiring*.

4 Generalised Relations and Relation Lifting

This section introduces generalised relations valued in a partial commutative semiring, and shows how to lift polynomial endofunctors on Set to the category of generalised relations. We begin by fixing a partial commutative semiring $(S,+,0,\bullet,1)$, and noting that the partial monoid $(S,+,0)$ can be used to define a preorder relation on S as follows:

$$x \sqsubseteq y \quad \text{if and only if} \quad \text{there exists } z \in S \text{ such that } x+z=y$$

for $x,y \in S$. It is then straightforward to show (using the definition of a partial commutative semiring) that the preorder $\sqsubseteq$ has $0 \in S$ as bottom element, and is preserved by $\bullet$ in each argument. Proper (i.e. not partial) semirings where the preorder $\sqsubseteq$ is a partial order are called *naturally ordered* [5]. We here extend this terminology to partial semirings.

Example 4.1. For the monads in Example 2.1, the preorders associated to the induced partial semirings (see Example 3.2) are all partial orders: $\leq$ on $\{\bot,\top\}$ for $\mathsf{T}=\mathscr{P}$, $\leq$ on $[0,1]$ for $\mathsf{T}=\mathscr{S}$, and $\geq$ on $\mathbb{N}^\infty$ for $\mathsf{T}=\mathsf{T}_W$.

We let Rel denote the category[4] with objects given by triples (X,Y,R), where $R : X\times Y \to S$ is a function defining a *multi-valued relation* (or *S-relation*), and with arrows from (X,Y,R) to (X',Y',R') given by pairs of functions (f,g) as below, such that $R \sqsubseteq R' \circ (f\times g)$:

$$\begin{tikzcd}
X\times Y \arrow[r, "f\times g"] \arrow[d, "R"'] \arrow[dr, phantom, "\sqsubseteq"] & X'\times Y' \arrow[d, "R'"] \\
S \arrow[r, equal] & S
\end{tikzcd}$$

Here, the order $\sqsubseteq$ on S has been extended pointwise to S-relations with the same carrier.

We write $\mathsf{Rel}_{X,Y}$ for the *fibre over* (X,Y), that is, the full subcategory of Rel whose objects are S-relations over $X\times Y$ and whose arrows are given by $(1_X,1_Y)$. It is straightforward to check that the functor $q : \mathsf{Rel} \to \mathsf{Set}\times\mathsf{Set}$ taking (X,Y,R) to (X,Y) defines a fibration: the reindexing functor $(f,g)^* : \mathsf{Rel}_{X',Y'} \to \mathsf{Rel}_{X,Y}$ takes $R' : X'\times Y' \to S$ to $R'\circ(f\times g) : X\times Y \to S$.

We now proceed to generalising relation lifting to S-relations.

Definition 4.1. *Let $F : \mathsf{Set}\to\mathsf{Set}$. A* relation lifting of F *is a functor*[5] $\Gamma : \mathsf{Rel}\to\mathsf{Rel}$ *such that $q\circ\Gamma = (F\times F)\circ q$:*

$$\begin{tikzcd}
\mathsf{Rel} \arrow[r, "\Gamma"] \arrow[d, "q"'] & \mathsf{Rel} \arrow[d, "q"] \\
\mathsf{Set}\times\mathsf{Set} \arrow[r, "F\times F"'] & \mathsf{Set}\times\mathsf{Set}
\end{tikzcd}$$

We immediately note a fundamental difference compared to standard relation lifting as defined in Section 2.1. While in the case of standard relations each functor admits exactly one lifting, Definition 4.1 implies neither the existence nor the uniqueness of a lifting. We defer the study of a canonical lifting (similar to $\mathsf{Rel}(F)$ in the case of standard relations) to future work, and show how to define a relation lifting of F in the case when F is a polynomial functor. To this end, we make the additional assumption that the unit 1 of the semiring multiplication is a top element (which we also write as $\top$) for the preorder

[4] To keep notation simple, the dependency on S is left implicit.

[5] Given the definition of the fibration q, such a functor is automatically a morphism of fibrations.

$\sqsubseteq$. Recall that $\sqsubseteq$ also has a bottom element (which we will sometimes denote by $\perp$), given by the unit 0 of the (partial) semiring addition. The definition of the relation lifting of a polynomial functor F is by structural induction on F and makes use of the semiring structure on S:

- If $F = \mathsf{Id}$, $\mathsf{Rel}(F)$ takes an S-relation to itself.
- If $F = C$, $\mathsf{Rel}(F)$ takes an S-relation to the equality relation $\mathsf{Eq}(C) : C \times C \to S$ given by

$$\mathsf{Eq}_C(c, c') \;=\; \begin{cases} \top \text{ if } c = c' \\ \perp \text{ otherwise} \end{cases}$$

- If $F = F_1 \times F_2$, $\mathsf{Rel}(F)$ takes an S-relation $R : X \times Y \to S$ to:

$$(F_1X \times F_2X) \times (F_1Y \times F_2Y) \xrightarrow{\langle \pi_1 \times \pi_1, \pi_2 \times \pi_2 \rangle} (F_1X \times F_1Y) \times (F_2X \times F_2Y) \xrightarrow{\mathsf{Rel}(F_1)(R) \times \mathsf{Rel}(F_2)(R)} S \times S \xrightarrow{\bullet} S$$

  The functoriality of this definition follows from the preservation of $\sqsubseteq$ by $\bullet$ (see Section 3).
- if $F = F_1 + F_2$, $\mathsf{Rel}(F)(R) : (F_1X + F_2X) \times (F_1Y + F_2Y) \to S$ is defined by case analysis:

$$\mathsf{Rel}(F)(R)(\iota_i(u), \iota_j(v)) \;=\; \begin{cases} \mathsf{Rel}(F_i)(R)(u, v) & \text{if } i = j \\ \perp & \text{otherwise} \end{cases}$$

  for $i, j \in \{1, 2\}$, $u \in F_iX$ and $v \in F_jY$. This definition generalises straightforwardly from binary to set-indexed coproducts.

Remark 4.2. A more general definition of relation lifting, which applies to arbitrary functors on Set, is outside the scope of this paper. We note in passing that such a relation lifting could be defined by starting from a *generalised predicate lifting* $\delta : F \circ \mathsf{P}_0 \Rightarrow \mathsf{P}_0 \circ F$ for the functor F, similar to the predicate liftings used in the work on coalgebraic modal logic [17]. Here, the contravariant functor $\mathsf{P}_0 : \mathsf{Set} \to \mathsf{Set}^{\mathsf{op}}$ takes a set X to the hom-set $\mathsf{Set}(X, S)$. Future work will also investigate the relevance of the results in [6, 7] to a general definition of relation lifting in our setting. Specifically, the work in loc. cit. shows how to construct truth-preserving predicate liftings and equality-preserving relation liftings for arbitrary functors on the base category of a *Lawvere fibration*, to the total category of that fibration.

For the remainder of this paper, we take $(S, +, 0, \bullet, 1)$ to be the partial semiring derived in Section 3 from a commutative, partially additive monad T, and we view S as the set of truth values. In the case of the powerset monad, this corresponds to the standard view of relations as subsets, whereas in the case of the sub-probability distribution monad, this results in relations given by valuations in the interval $[0, 1]$.

Example 4.3. Let $F : \mathsf{Set} \to \mathsf{Set}$ be given by $FX = 1 + A \times X$, with A a set (of labels), and let $(S, +, 0, \bullet, 1)$ be the partial semiring with carrier $\mathsf{T}1$ defined in Section 3.

- For $\mathsf{T} = \mathscr{P}$, $\mathsf{Rel}(F)$ takes a (standard) relation $R \subseteq X \times Y$ to the relation

$$\{(\iota_1(*), \iota_1(*)\} \cup \{((a, x), (a, y)) \mid a \in A, (x, y) \in R\}$$

- For $\mathsf{T} = \mathscr{S}$, $\mathsf{Rel}(F)$ takes $R : X \times Y \to [0, 1]$ to the relation $R' : FX \times FY \to [0, 1]$ given by

$$R'(\iota_1(*), \iota_1(*)) = 1 \qquad R'((a, x), (a, y)) = R(x, y) \qquad R'(u, v) = 0 \;\text{ in all other cases}$$

- For $\mathsf{T} = \mathsf{T}_W$, $\mathsf{Rel}(F)$ takes $R : X \times Y \to \mathbb{N}^\infty$ to the relation $R' : FX \times FY \to \mathbb{N}^\infty$ given by

$$R'(\iota_1(*), \iota_1(*)) = 0 \qquad R'((a, x), (a, y)) = R(x, y) \qquad R'(u, v) = \infty \;\text{ in all other cases}$$

5 From Bisimulation to Traces

Throughout this section we fix a commutative, partially additive monad $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$ and assume, as in the previous section, that the natural preorder $\sqsubseteq$ induced by the partial commutative semiring obtained in Section 3 has the multiplication unit $\eta_1(*) \in \mathsf{T}1$ as top element. Furthermore, we assume that this preorder is an ω^{op}*-chain complete* partial order, where ω^{op}-chain completeness amounts to any decreasing chain $x_1 \sqsupseteq x_2 \sqsupseteq \ldots$ having a greatest lower bound $\sqcap_{i\in\omega} x_i$. These assumptions are clearly satisfied by the orders in Example 4.1.

We now show how combining the liftings of polynomial functors to the category of generalised relations valued in the partial semiring $\mathsf{T}1$ (as defined in Section 4) with so-called *extension liftings* which arise canonically from the monad T, can be used to give an account of the linear-time behaviour of a state in a coalgebra with branching. The type of such a coalgebra can be any composition involving polynomial endofunctors and the branching monad T, although compositions of type $\mathsf{T} \circ F$, $G \circ \mathsf{T}$ and $G \circ \mathsf{T} \circ F$ with F and G polynomial endofunctors are particularly emphasised in what follows.

We begin with some informal motivation. When Rel is the standard category of binary relations, recall from Section 2.2 that an F-bisimulation is simply a $\mathsf{Rel}(F)$-coalgebra, and that the largest F-bisimulation between two F-coalgebras (C,γ) and (D,δ) can be obtained as the greatest fixpoint of the monotone operator on $\mathsf{Rel}_{C\times D}$ which takes a relation R to the relation $(\gamma\times\delta)^*(\mathsf{Rel}(F)(R))$. Generalising the notion of F-bisimulation from standard relations to $\mathsf{T}1$-relations makes little sense when the systems of interest are F-coalgebras. However, when considering say, coalgebras of type $\mathsf{T}\circ F$, it turns out that liftings of F to the category of $\mathsf{T}1$-relations (as defined in Section 4) can be used to describe the *linear-time behaviour* of states in such a coalgebra, when combined with suitable liftings of T to the same category of relations. To see why, let us consider labelled transition systems viewed as coalgebras of type $\mathscr{P}(1+A\times\mathsf{Id})$. In such a coalgebra $\gamma : C \to \mathscr{P}(1+A\times C)$, explicit termination is modelled via transitions $c \to \iota_1(*)$, whereas deadlock (absence of a transition) is modelled as $\gamma(c) = \emptyset$. In this case, $\mathsf{Rel}(\mathscr{P})\circ\mathsf{Rel}(1+A\times\mathsf{Id})$ is naturally isomorphic to $\mathsf{Rel}(\mathscr{P}(1+A\times\mathsf{Id}))$ [6], and takes a relation $R\subseteq X\times Y$ to the relation $R' \subseteq \mathscr{P}(1+A\times X)\times\mathscr{P}(1+A\times Y)$ given by

$$(U,V)\in R' \quad \text{if and only if} \quad \begin{cases} \text{if } \iota_1(*)\in U \text{ then } \iota_1(*)\in V, \text{ and conversely} \\ \text{if } (a,x)\in U \text{ then there exists } (a,y)\in V \text{ with } (x,y)\in R, \text{ and conversely} \end{cases}$$

Thus, the largest $\mathscr{P}(1+A\times\mathsf{Id})$-bisimulation between two coalgebras (C,γ) and (D,δ) can be computed as the greatest fixpoint of the operator on $\mathsf{Rel}_{C,D}$ obtained as the composition

$$R\subseteq C\times D \xmapsto{\mathsf{Rel}(F)} R_1 \subseteq FC\times FD \xmapsto{\mathsf{Rel}(\mathscr{P})} R_2 \subseteq \mathscr{P}(FC)\times\mathscr{P}(FD) \xmapsto{(\gamma\times\delta)^*} R' \subseteq C\times D \qquad (2)$$

where $F = 1+A\times\mathsf{Id}$. Note first that $\mathsf{Rel}(\mathscr{P})$ (defined in Section 2.1 for an arbitrary endofunctor on Set) takes a relation $R\subseteq X\times Y$ to the relation $R'\subseteq\mathscr{P}(X)\times\mathscr{P}(Y)$ given by

$$(U,V)\in R' \;\text{ if and only if }\; \text{for all } x\in U \text{ there exists } y\in V \text{ with } (x,y)\in R, \text{ and conversely}$$

Now consider the effect of replacing $\mathsf{Rel}(\mathscr{P})$ in (2) with the lifting $L : \mathsf{Rel}\to\mathsf{Rel}$ that takes a relation $R\subseteq X\times Y$ to the relation $R'\subseteq\mathscr{P}(X)\times Y$ given by

$$(U,y)\in R' \;\text{ if and only if }\; \text{there exists } x\in U \text{ with } (x,y)\in R$$

[6] A similar observation holds more generally for $\mathscr{P}\circ F$ with F a polynomial endofunctor. In general, only a natural transformation $\mathsf{Rel}(F\circ G)\Rightarrow\mathsf{Rel}(F)\circ\mathsf{Rel}(G)$ exists, see [11, Exercise 4.4.6].

To do so, we must change the type of the coalgebra (D,δ) from $\mathscr{P}\circ F$ to just F. A closer look at the resulting operator on $\mathsf{Rel}_{C,D}$ reveals that it can be used to test for the existence of a matching trace: each state of the F-coalgebra (D,δ) can be associated a *maximal trace*, i.e. an element of the final F-coalgebra, by finality. In particular, when $F = 1 + A \times \mathsf{Id}$, maximal traces are either finite or infinite sequences of elements of A. Thus, the greatest fixpoint of the newly defined operator on $\mathsf{Rel}_{C\times D}$ corresponds to the relation on $C \times D$ given by

$c \ni_{\mathsf{tr}} d$ if and only if there exists a sequence of choices of transitions starting from $c \in C$ that leads to exactly the same maximal trace (element of $A^* \cup A^\omega$) as the single trace of $d \in D$

This relation models the ability of the state c to exhibit the same trace as that of d.

The remainder of this section formalises the above intuitions, and generalises them to arbitrary monads T and polynomial endofunctors F, as well as to arbitrary compositions involving the monad T and polynomial endofunctors. We begin by restricting attention to coalgebras of type $\mathsf{T}\circ F$, with the monad T capturing branching and the endofunctor F describing the structure of individual transitions. In this case it is natural to view the elements of the final F-coalgebra as possible *linear-time* observable behaviours of states in $\mathsf{T}\circ F$-coalgebras. Similarly to the above discussion, we let (C,γ) and (D,δ) denote a $\mathsf{T}\circ F$-coalgebra and respectively an F-coalgebra. The lifting of F to $\mathsf{T}1$-relations will be used as part of an operator on $\mathsf{Rel}_{C,D}$. In order to generalise the lifting L above to arbitrary monads T, we recall the following result from [15], which assumes a strong monad T on a cartesian closed category.

Proposition 5.1 ([15, Proposition 4.1])**.** *Let (B,β) be a T-algebra. For any $f : X \times Y \to B$, there exists a unique* 1*-linear $\overline{f} : \mathsf{T}X \times Y \to B$ making the following triangle commute:*

$$\begin{tikzcd}
\mathsf{T}X \times Y \arrow[r, "\overline{f}"] & B \\
X \times Y \arrow[u, "\eta_X \times 1_Y"] \arrow[ur, "f"'] &
\end{tikzcd}$$

In the above, 1-*linearity* is linearity in the first variable. More precisely, for T-algebras (A,α) and (B,β), a map $f : A \times Y \to B$ is called 1-*linear* if the following diagram commutes:

$$\begin{tikzcd}
\mathsf{T}(A) \times Y \arrow[r, "\mathsf{st}'_{A,Y}"] \arrow[d, "\alpha \times 1_Y"'] & \mathsf{T}(A \times Y) \arrow[r, "\mathsf{T}(f)"] & \mathsf{T}(B) \arrow[d, "\beta"] \\
A \times Y \arrow[rr, "f"'] & & B
\end{tikzcd}$$

Clearly 1-linearity should be expected of the lifting $L(R) : \mathsf{T}X \times Y \to \mathsf{T}1$ of a relation $R : X \times Y \to \mathsf{T}1$, as this amounts to $L(R)$ commuting with the T-algebra structures $(\mathsf{T}X, \mu_X)$ and $(\mathsf{T}1, \mu_1)$. Given this, the diagram of Proposition 5.1 forces the definition of the generalised lifting.

Definition 5.2. *The* extension lifting $L_{\mathsf{T}} : \mathsf{Rel} \to \mathsf{Rel}$ *is the functor taking a relation $R : X \times Y \to \mathsf{T}1$ to its unique* 1*-linear extension $\overline{R} : \mathsf{T}X \times Y \to \mathsf{T}1$.*

Remark 5.1. It follows from [15] that a direct definition of the relation $\overline{R} : \mathsf{T}X \times Y \to \mathsf{T}1$ is as the composition

$$\mathsf{T}X \times Y \xrightarrow{\mathsf{st}'_{X,Y}} \mathsf{T}(X \times Y) \xrightarrow{\mathsf{T}(R)} \mathsf{T}^2 1 \xrightarrow{\mu_1} \mathsf{T}1$$

This also yields functoriality of L_{T}, which follows from the functoriality of its restriction to each fibre category $\mathsf{Rel}_{X,Y}$, as proved next.

Proposition 5.3. *The mapping* $R \in \mathsf{Rel}_{X,Y} \mapsto \overline{R} \in \mathsf{Rel}_{\mathsf{T}X,Y}$ *is functorial.*

Proof (Sketch). Let $R, R' \in \mathsf{Rel}_{X,Y}$ be such that $R \sqsubseteq R'$. Hence, there exists $S \in \mathsf{Rel}_{X,Y}$ such that $R + S = R'$ (pointwise). To show that $\overline{R} \sqsubseteq \overline{R'}$, it suffices to show that $\mu_1 \circ \mathsf{T}(R) \sqsubseteq \mu_1 \circ \mathsf{T}(R')$ (pointwise). To this end, we note that commutativity of the map δ with the monad multiplication, proved in [3, Lemma 15 (iii)] and captured by the commutativity of the lower diagram below (via the plain arrows)

$$\begin{tikzcd}
\mathsf{T}^2 1 \arrow[r, "\mu_1"] & \mathsf{T}1 \\
\mathsf{T}^2(1+1) \arrow[u, "\mathsf{T}^2!"] \arrow[r, "\mu_{1+1}"] \arrow[d, shift left, "\mathsf{T}\delta"] & \mathsf{T}(1+1) \arrow[u, "\mathsf{T}!"] \arrow[dd, shift right, "\delta"'] \\
\mathsf{T}(\mathsf{T}1 \times \mathsf{T}1) \arrow[u, dashed, shift left, "\mathsf{T}q_{1,1}"] \arrow[d, "\langle \mathsf{T}\pi_1, \mathsf{T}\pi_2 \rangle"] & \\
\mathsf{T}^2 1 \times \mathsf{T}^2 1 \arrow[r, "\mu_1 \times \mu_1"'] & \mathsf{T}1 \times \mathsf{T}1 \arrow[uu, dashed, shift right, "q_{1,1}"']
\end{tikzcd}$$

also yields commutativity of the whole diagram (via the dashed arrows). This formalises the commutativity of $+$ (defined as $\mathsf{T}! \circ q_{1,1}$) with the monad multiplication. Now pre-composing this commutative diagram (dashed arrows) with the map

$$\mathsf{T}(X \times Y) \longrightarrow \mathsf{T}(\mathsf{T}1 \times \mathsf{T}1)$$

given by the image under T of the map $(x,y) \mapsto \langle R(x,y), S(x,y) \rangle$ yields

$$(\mu_1 \circ \mathsf{T}(R)) + (\mu_1 \circ \mathsf{T}(S)) = \mu_1 \circ \mathsf{T}(R+S) = \mu_1 \circ \mathsf{T}R'$$

and therefore, using the definition of $\sqsubseteq$, $\mu_1 \circ \mathsf{T}(R) \sqsubseteq \mu_1 \circ \mathsf{T}(R')$. This concludes the proof. □

Thus, L_{T} is a functor making the following diagram commute:

$$\begin{tikzcd}
\mathsf{Rel} \arrow[r, "L_{\mathsf{T}}"] \arrow[d, "q"'] & \mathsf{Rel} \arrow[d, "q"] \\
\mathsf{Set} \times \mathsf{Set} \arrow[r, "\mathsf{T} \times \mathsf{Id}"'] & \mathsf{Set} \times \mathsf{Set}
\end{tikzcd}$$

We are finally ready to give an alternative account of maximal traces of $\mathsf{T} \circ F$-coalgebras.

Definition 5.4. *Let* (C, γ) *denote a* $\mathsf{T} \circ F$*-coalgebra, and let* (Z, ζ) *denote the final* F*-coalgebra. The* maximal trace map $\mathsf{tr}_\gamma : C \to (\mathsf{T}1)^Z$ of γ *is the exponential transpose of the greatest fixpoint* $R : C \times Z \to \mathsf{T}1$ *of the operator* $\mathscr{O} : \mathsf{Rel}_{C,Z} \to \mathsf{Rel}_{C,Z}$ *given by the composition*

$$\mathsf{Rel}_{C,Z} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FC,FZ} \xrightarrow{L_{\mathsf{T}}} \mathsf{Rel}_{\mathsf{T}(FC),FZ} \xrightarrow{(\gamma \times \zeta)^*} \mathsf{Rel}_{C,Z}$$

The above definition appeals to the existence of least fixpoints in chain-complete partial orders, as formalised in the following fixpoint theorem from [4].

Theorem 5.5 ([4, 8.22])**.** *Let P be a complete partial order and let $\mathscr{O} : P \to P$ be order-preserving. Then $\mathscr{O}$ has a least fixpoint.*

Definition 5.4 makes use of this result applied to the *dual* of the order $\sqsubseteq$. Our assumption that $\sqsubseteq$ is ω^{op}-chain complete makes the dual order a complete partial order. Monotonicity of the operator in Definition 5.4 is an immediate consequence of the functoriality of $\mathsf{Rel}(F)$, L_{T} and $(\gamma \times \delta)^*$.

[4] also gives a construction for the least fixpoint of an order-preserving operator on a complete partial order, which involves taking a limit over an ordinal-indexed chain. Instantiating this construction to the dual of the order $\sqsubseteq$ yields an ordinal-indexed sequence of relations (R_α), where:

- $R_0 = \top$ (i.e. the relation on $C \times D$ given by $(c,d) \mapsto 1$),
- $R_{\alpha+1} = \mathscr{O}(R_\alpha)$,
- $R_\alpha = \sqcap_{\beta<\alpha} R_\beta$, if α is a limit ordinal.

Remark 5.2. While in the case $\mathsf{T} = \mathscr{P}$, restricting to finite-state coalgebras (C,γ) and (D,δ) results in the above sequence of relations stabilising in a finite number of steps, for $\mathsf{T} = \mathscr{S}$ or $T = \mathsf{T}_W$ this is not in general the case. However, for probabilistic or weighted computations, an approximation of the greatest fixpoint may be sufficient for verification purposes, since a threshold can be provided as part of a verification task.

Remark 5.3. By replacing the F-coalgebra (Z,ζ) by (I,α^{-1}) with (I,α) an *initial* F-algebra, one obtains an alternative account of *finite* traces of states in $\mathsf{T} \circ F$-coalgebras, with the *finite trace map* $\mathsf{ftr}_\gamma : C \to (\mathsf{T}1)^I$ of a $\mathsf{T} \circ F$-coalgebra (C,γ) being obtained via the greatest fixpoint of essentially the same operator $\mathscr{O}$, but this time on $\mathsf{Rel}_{C,I}$. In fact, one can use any F-coalgebra in place of (Z,ζ), and for a specific verification task, a coalgebra with a finite state space, encoding a given linear-time behaviour, might be all that is required.

Remark 5.4. The choice of functor F directly impacts on the notion of linear-time behaviour. For example, by regarding labelled transition systems as coalgebras of type $\mathscr{P}(A \times \mathsf{Id})$ instead of $\mathscr{P}(1 + A \times \mathsf{Id})$ (i.e. not modelling successful termination explicitly), finite traces are not anymore accounted for – the elements of the final F-coalgebra are given by infinite sequences of elements of A. This should not be regarded as a drawback, in fact it illustrates the flexibility of our approach.

Example 5.5. Let F denote an arbitrary polynomial functor (e.g. $1 + A \times \mathsf{Id}$).

- For $T = \mathscr{P}$, the extension lifting $L_{\mathscr{P}} : \mathsf{Rel} \to \mathsf{Rel}$ takes a (standard) relation $R \subseteq X \times Y$ to the relation $L_{\mathscr{P}}(R) \subseteq \mathscr{P}(X) \times Y$ given by

  $$(U,y) \in L_{\mathscr{P}}(R) \text{ if and only if there exists } x \in U \text{ with } (x,y) \in R$$

  As a result, the greatest fixpoint of $\mathscr{O}$ relates a state c in a $\mathscr{P} \circ F$-coalgebra (C,γ) with a state z of the final F-coalgebra if and only if there exists a sequence of choices in the unfolding of γ starting from c, that results in an F-behaviour bisimilar to z. This was made more precise in [1], where infinite two-player games were developed for verifying whether a state of a $\mathscr{P} \circ F$-coalgebra has a certain maximal trace (element of the final F-coalgebra).

- For $T = T_{\mathscr{S}}$, the extension lifting $L_{\mathscr{S}} : \mathsf{Rel} \to \mathsf{Rel}$ takes a valuation $R : X \times Y \to [0,1]$ to the valuation $L_{\mathscr{S}}(R) : \mathscr{S}(X) \times Y \to [0,1]$ given by

  $$L_{\mathscr{S}}(R)(\varphi,y) = \sum_{x \in \mathsf{sup}(\varphi)} \varphi(x) * R(x,y)$$

Thus, the greatest fixpoint of $\mathscr{O}$ yields, for each state in a $\mathscr{S} \circ F$-coalgebra and each potential maximal trace z, the probability of this trace being exhibited. As computing these probabilities amounts to multiplying infinitely-many probability values, the probability of an infinite trace will often turn out to be 0 (unless from some point in the unfolding of a particular state, probability values of 1 are associated to the individual transitions that match a particular infinite trace). This may appear as a deficiency of our framework, and one could argue that a measure-theoretic approach, whereby a probability measure is derived from the probabilities of finite prefixes of infinite traces, would be more appropriate. Future work will investigate the need for a measure-theoretic approach. At this point, we simply point out that in a future extension of the present approach to linear-time logics (where individual maximal traces are to be replaced by linear-time temporal logic formulas), this deficiency is expected to disappear.

- For $\mathsf{T} = \mathsf{T}_W$, the extension lifting $L_W : \mathsf{Rel} \to \mathsf{Rel}$ takes a *weighted relation* $R : X \times Y \to W$ to the relation $L_W(R) : \mathsf{T}_W(X) \times Y \to W$ given by

$$L_W(R)(f,y) = \min_{x \in \mathsf{sup}(f)} (f(x) + R(x,y))$$

for $f : X \to W$ and $y \in Y$. Thus, the greatest fixpoint of $\mathscr{O}$ maps a pair (c,z), with c a state in a $\mathsf{T}_W \circ F$-coalgebra and z a maximal trace, to the *cost* (computed via the min function) of exhibiting that trace. The case of weighted computations is somewhat different from our other two examples of branching types, in that the computation of the fixpoint starts from a relation that maps each pair of states (c,z) to the value $0 \in \mathbb{N}^\infty$ (the top element for $\sqsubseteq$), and refines this down (w.r.t. the $\sqsubseteq$ order) through stepwise unfolding of the coalgebra structures γ and ζ.

The approach presented above also applies to coalgebras of type $G \circ \mathsf{T}$ with G a polynomial endofunctor, and more generally to coalgebras whose type is obtained as the composition of polynomial endofunctors and the monad T, with possibly several occurrences of T in this composition. In the case of $G \circ \mathsf{T}$-coalgebras, instantiating our approach yields different results to the extension semantics proposed in [12]. Specifically, the instantiation involves taking (Z,ζ) to be a final G-coalgebra and (C,γ) to be an arbitrary $G \circ \mathsf{T}$-coalgebra, and considering the monotone operator on $\mathsf{Rel}_{C,Z}$ given by the composition

$$\mathsf{Rel}_{C,Z} \xrightarrow{L_{\mathsf{T}}} \mathsf{Rel}_{\mathsf{T}C,Z} \xrightarrow{\mathsf{Rel}(G)} \mathsf{Rel}_{G(\mathsf{T}C),GZ} \xrightarrow{(\gamma\times\zeta)^*} \mathsf{Rel}_{C,Z} \tag{3}$$

The following example illustrates the difference between our approach and that of [12].

Example 5.6. For $G = 2 \times \mathsf{Id}^A$ with A a finite alphabet and $\mathsf{T} = \mathscr{P}$, $G \circ \mathsf{T}$-coalgebras are non-deterministic automata, whereas the elements of the final G-coalgebra are given by functions $z : A^* \to 2$ and correspond to languages over A. In this case, the greatest fixpoint of the operator in (3) maps a pair (c,z), with c a state of the automaton and z a language over A, to $\top$ if and only if there exists a sequence of choices in the unfolding of the automaton starting from c that results in a deterministic automaton which accepts the language denoted by z. Taking the union over all z such that (c,z) is mapped to $\top$ now gives the language accepted by the non-deterministic automaton with c as initial state, but only under the assumption that for each $a \in A$, an a-labelled transition exists from any state of the automaton. This example points to the need to further generalise our approach, so that in particular it can also be applied to pairs consisting of a $G \circ \mathsf{T}$-coalgebra and a G'-coalgebra, with G' different from G. This would involve considering relation liftings for pairs of (polynomial) endofunctors. We conjecture that taking G and T as above and $G' = 1 + A \times \mathsf{Id}$ would allow us to recover the notion of acceptance of a finite word over A by a non-deterministic automaton.

Finally, we sketch the general case of coalgebras whose type is obtained as the composition of several endofunctors on Set, one of which is a monad T that accounts for the presence of branching in the system, while the remaining endofunctors are polynomial and jointly determine the notion of linear-time behaviour. For simplicity of presentation, we only consider coalgebras of type $G \circ \mathsf{T} \circ F$, with the final $G \circ F$-coalgebra (Z, ζ) providing the domain of possible linear-time behaviours.

Definition 5.6. *The* linear-time behaviour *of a state in a coalgebra (C, γ) of type $G \circ \mathsf{T} \circ F$ is the greatest fixpoint of an operator $\mathscr{O}$ on $\mathsf{Rel}_{C,Z}$ defined by the composition:*

$$\mathsf{Rel}_{C,Z} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FC,FZ} \xrightarrow{L_{\mathsf{T}}} \mathsf{Rel}_{\mathsf{T}(FC),FZ} \xrightarrow{\mathsf{Rel}(G)} \mathsf{Rel}_{G(\mathsf{T}FC),GFZ} \xrightarrow{(\gamma \times \zeta)^*} \mathsf{Rel}_{C,Z} \tag{4}$$

The greatest fixpoint of $\mathscr{O}$ measures the extent with which a state in a $G \circ \mathsf{T} \circ F$-coalgebra can exhibit a given linear behaviour (element of the final $G \circ F$-coalgebra). Definition 5.6 generalises straightforwardly to coalgebraic types given by arbitrary compositions of polynomial endofunctors and the monad T, with the extension lifting L_{T} being used once for each occurrence of T in such a composition.

Example 5.7. Coalgebras of type $G \circ \mathsf{T} \circ F$, where $G = (1 + \mathsf{Id})^A$ and $F = \mathsf{Id} \times B$, model systems with branching, with both inputs (from a finite set A) and outputs (in a set B). In this case, the possible linear behaviours are given by special trees, with both finite and infinite branches, whose edges are labelled by elements of A (from each node, one outgoing edge for each $a \in A$), and whose nodes (with the exception of the root) are either labelled by $* \in 1$ (for leaves) or by an element of B (for non-leaves). The linear-time behaviour of a state in a $G \circ \mathsf{T} \circ F$-coalgebra is then given by:

- the set of trees that can be exhibited from that state, when $\mathsf{T} = \mathscr{P}$,
- the probability of exhibiting each tree (with the probabilities corresponding to different branches being *multiplied* when computing this probability), when $\mathsf{T} = \mathscr{S}$, and
- the minimum cost of exhibiting each tree (with the costs of different branches being *added* when computing this cost), when $\mathsf{T} = \mathsf{T}_W$.

The precise connection between our approach and earlier work in [9, 1, 12] is yet to be explored. In particular, our assumptions are different from those of loc. cit., for example in [9] the $\mathrm{DCPO}_\perp$-enrichedness of the Kleisli category of T is required.

Remark 5.8. Our approach does not *directly* apply to the probability distribution monad (defined similarly to the sub-probability distribution monad, but with probabilities adding up to exactly 1), as this monad does not satisfy the condition $\mathsf{T}\emptyset = 1$ of Definition 3.1. However, systems where branching is described using probability distributions can still be dealt with, by regarding all probability distributions as sub-probability distributions.

In the remainder of this section, we briefly explore the usefulness of an operator similar to $\mathscr{O}$, which employs a similar extension lifting arising from the *double strength* of the monad T. We begin by noting that a result similar to Proposition 5.1 is proved in [15] for a commutative monad on a cartesian closed category.

Proposition 5.7 ([15, Proposition 9.3])**.** *Let (B, β) be a T-algebra. Then any $f : X \times Y \to B$ extends uniquely along $\eta_X \times \eta_Y$ to a bilinear $\tilde{f} : \mathsf{T}X \times \mathsf{T}Y \to B$, making the following triangle commute:*

$$\begin{tikzcd}
\mathsf{T}X \times \mathsf{T}Y \arrow[r, "\tilde{f}"] & B \\
X \times Y \arrow[u, "\eta_X \times \eta_Y"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Here, bilinearity amounts to linearity in each argument.

Definition 5.8. *For a commutative monad* $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, *the* double extension lifting $L'_{\mathsf{T}} : \mathsf{Rel} \to \mathsf{Rel}$ *is the functor taking a relation* $R : X \times Y \to \mathsf{T}1$ *to its unique bilinear extension* $\tilde{R} : \mathsf{T}X \times \mathsf{T}Y \to \mathsf{T}1$.

Remark 5.9. An alternative definition of L'_{T} is as the composition of L_{T} with a dual lifting, which takes a relation $R : X \times Y \to \mathsf{T}1$ to its unique 2-linear extension $\overline{R} : X \times \mathsf{T}Y \to \mathsf{T}1$.

Remark 5.10. Again, it can be shown that a direct definition of the relation $\tilde{R} : \mathsf{T}X \times \mathsf{T}Y \to \mathsf{T}1$ is as the composition

$$\mathsf{T}X \times \mathsf{T}Y \xrightarrow{\mathsf{dst}_{X,Y}} \mathsf{T}(X \times Y) \xrightarrow{\mathsf{T}(R)} \mathsf{T}^2 1 \xrightarrow{\mu_1} \mathsf{T}1$$

Proposition 5.9. *The mapping* $R \in \mathsf{Rel}_{X,Y} \mapsto \overline{R} \in \mathsf{Rel}_{X,\mathsf{T}Y}$ *is functorial.*

We now fix *two* $\mathsf{T} \circ F$-coalgebras (C,γ) and (D,δ) and explore the greatest fixpoint of the operator $\mathscr{O}' : \mathsf{Rel}_{C,D} \to \mathsf{Rel}_{C,D}$ defined by the composition

$$\mathsf{Rel}_{C,D} \xrightarrow{\mathsf{Rel}(F)} \mathsf{Rel}_{FC,FD} \xrightarrow{L'_{\mathsf{T}}} \mathsf{Rel}_{\mathsf{T}(FC),\mathsf{T}(FD)} \xrightarrow{(\gamma\times\zeta)^*} \mathsf{Rel}_{C,D}$$

As before, the operator $\mathscr{O}'$ is monotone and therefore admits a greatest fixpoint. We argue that this fixpoint also yields useful information regarding the linear-time behaviour of states in $\mathsf{T} \circ F$-coalgebras. Moreover, this generalises to coalgebras whose types are arbitrary compositions of polynomial functors and the branching monad T. This is expected to be of relevance when extending the linear-time view presented here to linear-time logics and associated formal verification techniques. The connection to formal verification constitutes work in progress, but the following examples motivate our claim that the lifting L'_{T} is worth further exploration.

Example 5.11. Let $F : \mathsf{Set} \to \mathsf{Set}$ be a polynomial endofunctor, describing some linear-type behaviour.

1. For non-deterministic systems (i.e. $\mathscr{P} \circ F$-coalgebras), the greatest fixpoint of $\mathscr{O}'$ relates two states if and only if they admit a common maximal trace.
2. For probabilistic systems (i.e. $\mathscr{S} \circ F$-coalgebras), the greatest fixpoint of $\mathscr{O}'$ measures the probability of two states exhibiting the same maximal trace.
3. For weighted systems (i.e. $\mathsf{T}_W \circ F$-coalgebras), the greatest fixpoint of $\mathscr{O}'$ measures the *joint* minimal cost of two states exhibiting the same maximal trace. To see this, note that the lifting L'_W : $\mathsf{Rel} \to \mathsf{Rel}$ takes a weighted relation $R : X \times Y \to W$ to the relation $L'_W(R) : \mathsf{T}_W(X) \times \mathsf{T}_W(Y) \to W$ given by
$$L'_W(R)(f,g) = \min_{x \in \mathsf{sup}(f), y \in \mathsf{sup}(g)} (f(x) + g(y) + R(x,y))$$

6 Conclusions and Future Work

We have provided a general and uniform account of the linear-time behaviour of a state in a coalgebra whose type incorporates some notion of branching (captured by a monad on Set). Our approach is compositional, and so far applies to notions of linear behaviour specified by *polynomial* endofunctors on Set. The key ingredient of our approach is the notion of extension lifting, which allows the branching behaviour of a state to be abstracted away in a coinductive fashion.

Immediate future work will attempt to exploit the results of [6, 7] in order to define generalised relation liftings for *arbitrary* endofunctors on Set, and to extend our approach to other base categories.

The work in loc. cit. could also provide an alternative description for the greatest fixpoint used in Definition 5.6.

The present work constitutes a stepping stone towards a coalgebraic approach to the formal verification of linear-time properties. This will employ linear-time coalgebraic temporal logics for the specification of system properties, and automata-based techniques for the verification of these properties, as outlined in [2] for the case of non-deterministic systems.

References

[1] Corina Cîrstea (2011): *Maximal Traces and Path-Based Coalgebraic Temporal Logics*. *Theoretical Computer Science* 412(38), pp. 5025–5042, doi:10.1016/j.tcs.2011.04.025.

[2] Corina Cîrstea (2011): *Model Checking Linear Coalgebraic Temporal Logics: An Automata-Theoretic Approach*. In: *Proc. CALCO 2011*, *Lecture Notes in Computer Science* 6859, Springer, pp. 130–144, doi:10.1007/978-3-642-22944-2_10.

[3] Dion Coumans & Bart Jacobs (2013): *Scalars, Monads, and Categories*. In C. Heunen, M. Sadrzadeh & E. Grefenstette, editors: *Quantum Physics and Linguistics. A Compositional, Diagrammatic Discourse*, Oxford Univ. Press, pp. 184–216, doi:10.1093/acprof:oso/9780199646296.001.0001.

[4] Brian A. Davey & Hilary A. Priestley (2002): *Introduction to Lattices and Order (2. ed.)*. Cambridge University Press, doi:10.1017/CBO9780511809088.

[5] Zoltan Ésik & Werner Kuich (2007): *Modern Automata Theory*. http://dmg.tuwien.ac.at/kuich/.

[6] Clément Fumex, Neil Ghani & Patricia Johann (2011): *Indexed Induction and Coinduction, Fibrationally*. In: *Proc. CALCO 2011*, *Lecture Notes in Computer Science* 6859, Springer, pp. 176–191, doi:10.1007/978-3-642-22944-2_13.

[7] Neil Ghani, Patricia Johann & Clément Fumex (2012): *Generic Fibrational Induction*. *Logical Methods in Computer Science* 8(2), doi:10.2168/LMCS-8(2:12)2012.

[8] Ichiro Hasuo, Kenta Cho, Toshiki Kataoka & Bart Jacobs (2013): *Coinductive Predicates and Final Sequences in a Fibration*. In: *Proc. MFPS XXIX*, pp. 181–216.

[9] Ichiro Hasuo, Bart Jacobs & Ana Sokolova (2007): *Generic Trace Semantics via Coinduction*. *Logical Methods in Computer Science* 3(4), pp. 1–36, doi:10.2168/LMCS-3(4:11)2007.

[10] Claudio Hermida & Bart Jacobs (1998): *Structural Induction and Coinduction in a Fibrational Setting*. *Inf. Comput.* 145(2), pp. 107–152, doi:10.1006/inco.1998.2725.

[11] Bart Jacobs (2012): *Introduction to Coalgebra. Towards Mathematics of States and Observations (Version 2.0)*. Draft.

[12] Bart Jacobs, Alexandra Silva & Ana Sokolova (2012): *Trace Semantics via Determinization*. In: *Proc. CMCS 2012*, *Lecture Notes in Computer Science* 7399, Springer, pp. 109–129, doi:10.1007/978-3-642-32784-1.

[13] Paris C. Kanellakis & Scott A. Smolka (1990): *CCS Expressions, Finite State Processes, and Three Problems of Equivalence*. *Inf. Comput.* 86(1), pp. 43–68, doi:10.1016/0890-5401(90)90025-D.

[14] Anders Kock (2011): *Monads and extensive quantities*. ArXiv:1103.6009.

[15] Anders Kock (2012): *Commutative monads as a theory of distributions*. *Theory and Applications of Categories* 26(4), pp. 97–131.

[16] Lawrence S. Moss (1999): *Coalgebraic Logic*. *Ann. Pure Appl. Logic* 96(1-3), pp. 277–317, doi:10.1016/S0168-0072(98)00042-6.

[17] Dirk Pattinson (2003): *Coalgebraic modal logic: soundness, completeness and decidability of local consequence*. *Theor. Comput. Sci.* 309(1-3), pp. 177–193, doi:10.1016/S0304-3975(03)00201-9.

A Coinductive Approach to Proof Search

José Espírito Santo
Centro de Matemática
Universidade do Minho
Portugal

Ralph Matthes
Institut de Recherche en Informatique de Toulouse (IRIT)
C.N.R.S. and University of Toulouse
France

Luís Pinto
Centro de Matemática
Universidade do Minho
Portugal

We propose to study proof search from a coinductive point of view. In this paper, we consider intuitionistic logic and a focused system based on Herbelin's LJT for the implicational fragment. We introduce a variant of lambda calculus with potentially infinitely deep terms and a means of expressing alternatives for the description of the "solution spaces" (called Böhm forests), which are a representation of all (not necessarily well-founded but still locally well-formed) proofs of a given formula (more generally: of a given sequent).

As main result we obtain, for each given formula, the reduction of a coinductive definition of the solution space to a effective coinductive description in a finitary term calculus with a formal greatest fixed-point operator. This reduction works in a quite direct manner for the case of Horn formulas. For the general case, the naive extension would not even be true. We need to study "co-contraction" of contexts (contraction bottom-up) for dealing with the varying contexts needed beyond the Horn fragment, and we point out the appropriate finitary calculus, where fixed-point variables are typed with sequents. Co-contraction enters the interpretation of the formal greatest fixed points - curiously in the semantic interpretation of fixed-point variables and not of the fixed-point operator.

1 Introduction

Proof theory starts with the observation that a proof is more than just the truth value of a theorem. A valid theorem can have many proofs, and several of them can be interesting. In this paper, we somehow extend this to the limit and study all proofs of a given proposition. Of course, who studies proofs can also study any of them (or count them, if there are only finitely many possible proofs, or try to enumerate them in the countable case). But we do this study somehow simultaneously: we introduce a language to express the full "solution space" of proof search. And since we focus on the generative aspects of proof search, it would seem awkward to filter out failed proof attempts from the outset. This does not mean that we pursue impossible paths in the proof search (which would hardly make sense) but that we allow to follow infinite paths. An infinite path does not correspond to a successful proof, but it is a structure of locally correct proof steps. In other words, we use coinductive syntax to model *all* locally correct proof figures. This gives rise to a not necessarily wellfounded search tree. However, to keep the technical effort simpler, we have chosen a logic where this tree is finitely branching, namely the implicational fragment of intuitionistic propositional logic (with proof system given by the cut-free fragment of the system $\overline{\lambda}$ by Herbelin [3]).

Lambda terms or variants of them (expressions that may have bound variables) are a natural means to express proofs (an observation that is called *the* Curry-Howard isomorphism) in implicational logic. Proof alternatives (locally, there are only finitely many of them since our logic has no quantifier that ranges over infinitely many individuals) can be formally represented by a finite sum of such solution space expressions, and it is natural to consider those sums up to equivalence of the *set* of the alternatives. Since infinite lambda-terms are involved and since whole solution spaces are being modeled, we call these coinductive terms *Böhm forests*.

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 28–43, doi:10.4204/EPTCS.126.3

By their coinductive nature, Böhm forests are no proper syntactic objects: they can be defined by all mathematical (meta-theoretic) means and are thus not “concrete”, as would be expected from syntactic elements. This freedom of definition will be demonstrated and exploited in the canonical definition (Definition 6) of Böhm forests as solutions to the task of proving a sequent (a formula A in a given context Γ). In a certain sense, nothing is gained by this representation: although one can calculate on a case-by-case basis the Böhm forest for a formula of interest and see that it is described as fixed point of a system of equations (involving auxiliary Böhm forests as solutions for the other meta-variables that appear in those equations), an arbitrary Böhm forest can only be observed to any finite depth, without ever knowing whether it is the expansion of a regular cyclic graph structure (the latter being a finite structure).

Our main result is that the Böhm forests that appear as solution spaces of sequents have such a finitary nature: more precisely, they can be interpreted as semantics of a finite term in a variant of lambda calculus with alternatives and formal greatest fixed-points. For the Horn fragment (where nesting of implications to the left is disallowed), this works very smoothly without surprises (Theorem 15). The full implicational case, however, needs some subtleties concerning the fixed-point variables over which the greatest fixed points are formed and about capturing redundancy that comes from the introduction of several hypotheses that suppose the same formula. The interpretation of the finite expressions in terms of Böhm forests needs a special operation that we call *co-contraction* (contraction bottom-up). However, this operation is already definable in terms of Böhm forests. Without this operation, certain repetitive patterns in the solution spaces due to the presence of negative occurrences of implications could not be identified. With it, we obtain the finitary representation (Theorem 24).

In the next section, we quickly recapitulate syntax and typing rules of the cut-free fragment of system $\overline{\lambda}$ and also carefully describe its restriction to Horn formulas.

Section 3 has the definition of the not necessarily well-founded proofs, corresponding to a coinductive reading of $\overline{\lambda}$ (including its typing system). This is system $\overline{\lambda}^{co}$. Elimination alternatives are then added to this system (yielding the Böhm forests), which directly allow the definition of the solution spaces for the proof search for sequents. We give several examples and then show that the defined solution spaces adequately represent all the $\overline{\lambda}^{co}$ proofs of a sequent.

In Section 4, we present first the finitary system to capture the Horn fragment and then modify it to get the main result for full implicational logic.

The paper closes with discussions on related and future work in Section 5.

2 Background

We recall below the cut-free fragment of system $\overline{\lambda}$ (a.k.a. LJT), a sequent calculus for intuitionistic implication by Herbelin [3].

Letters p,q,r are used to range over a base set of propositional variables (which we also call *atoms*). Letters A,B,C are used to range over the set of formulas (= types) built from propositional variables using the implication connective (that we write $A \supset B$) that is parenthesized to the right. Often we will use the fact that any implicational formula can be uniquely decomposed as $A_1 \supset A_2 \supset \ldots \supset A_n \supset p$ with $n \geq 0$, also written in vectorial notation as $\vec{A} \supset p$. For example, if the vector $\vec{A}$ is empty the notation means simply p, and if $\vec{A} = A_1, A_2$, the notation means $A_1 \supset (A_2 \supset p)$.

The cut-free expressions of $\overline{\lambda}$ are separated into terms and lists, and are given by:

$$\begin{array}{lrcl} \text{(terms)} & t,u & ::= & xl \,|\, \lambda x^A.t \\ \text{(lists)} & l & ::= & \langle\rangle \,|\, u :: l \end{array}$$

Figure 1: Typing rules of $\overline{\lambda}$

$$\frac{}{\Gamma | \langle\rangle : p \vdash p}\ \mathit{LAx} \qquad \frac{\Gamma \vdash u : A \quad \Gamma | l : B \vdash p}{\Gamma | u :: l : A \supset B \vdash p}\ \mathit{LIntro}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A . t : A \supset B}\ \mathit{RIntro} \qquad \frac{\Gamma | l : A \vdash p \qquad (y : A) \in \Gamma}{\Gamma \vdash yl : p}\ \mathit{App}$$

where a countably infinite set of variables ranged over by letters x, y, w, z is assumed. Note that in lambda-abstractions we adopt a *domain-full* presentation, annotating the bound variable with a formula. The term constructor xl is usually called *application*. Usually in the meta-level we prefer to write $x\langle t_1, \dots, t_n\rangle$ (with $n \in \mathbb{N}_0$) to range over application constructions, and avoid speaking about lists explicitly (where obviously, the notation $\langle t_1, \dots, t_n\rangle$ means $\langle\rangle$ if $n = 0$ and $t_1 :: l$, if $\langle t_2, \dots, t_n\rangle$ means l). In the meta-level, when we know $n = 0$, instead of $x\langle t_1, \dots, t_n\rangle$, we simply write the variable x.

We will view contexts Γ as finite lists of declarations $x : A$, where no variable x occurs twice. The context $\Gamma, x : A$ is obtained from Γ by adding the declaration $x : A$, and will only be written if this yields again a valid context, i. e., if x is not declared in Γ. The system has a form of sequent for each class of expressions:

$$\Gamma \vdash t : A \qquad\qquad \Gamma | l : A \vdash p.$$

Note the restriction to *atomic sequents* (the RHS formula is an atom) in the case of list sequents.

The rules of $\overline{\lambda}$ for deriving sequents are in Figure 1. Note that, as list sequents are atomic, the conclusion of the application rule is also atomic. This is not the case in Herbelin's original system [3], where list sequents can have a non-atomic formula on the RHS. In the variant of cut-free $\overline{\lambda}$ we adopted, the only rule available for deriving a term sequent whose RHS is an implication is *RIntro*. Still, our atomic restriction will not cause loss of completeness of the system for intuitionistic implication. This restriction is typically adopted in systems tailored for proof search, as for example systems of focused proofs. In fact, $\overline{\lambda}$ corresponds to a focused backward chaining system where all atoms are *asynchronous* (see e. g. Liang and Miller [7]).

We will need the following properties of $\overline{\lambda}$.

Lemma 1 (Type uniqueness) *1. Given Γ and t, there is at most one A such that $\Gamma \vdash t : A$.*

2. Given Γ, l and A, there is at most one p such that $\Gamma | l : A \vdash p$.

Proof Simultaneous induction on derivability. □

Since the empty list $\langle\rangle$ has no type index, we need to know A in the second statement of the previous lemma.

Lemma 2 (Inversion of typing) *In $\overline{\lambda}$:*

1. $\Gamma \vdash \lambda x^A . t : B$ iff there exists C s.t. $B = A \supset C$ and $\Gamma, x : A \vdash t : C$;

2. $\Gamma \vdash x\langle t_1, \dots, t_k\rangle : A$ iff $A = p$ and there exists $\vec{B}$ s.t. $x : \vec{B} \supset p \in \Gamma$ and $\Gamma \vdash t_i : B_i$, for any i.

Proof 1. is immediate and 2. follows with the help of the fact that: $\Gamma | \langle t_1, \dots, t_k\rangle : B \vdash p$ iff there exist $B_1, \dots, B_k$ s.t. $B = B_1 \supset \dots \supset B_k \supset p$ and, for any i, $\Gamma \vdash t_i : B_i$ (proved by induction on k). □

Figure 2: Typing rules of $\overline{\lambda}_{\mathsf{Horn}}$

$$\frac{}{\Gamma|\langle\rangle : p \vdash p}\ LAx \qquad \frac{\Gamma \vdash u : p \quad \Gamma|l : H \vdash q}{\Gamma|u :: l : p \supset H \vdash q}\ LIntro$$

$$\frac{\Gamma|l : H \vdash p \qquad (y : H) \in \Gamma}{\Gamma \vdash yl : p}\ App$$

Now we identify the *Horn fragment* of cut-free $\overline{\lambda}$, that we denote by $\overline{\lambda}_{\mathsf{Horn}}$. The class of *Horn formulas* (also called *Horn clauses*) is given by the grammar:

$$\text{(Horn formulas)} \qquad H \quad ::= \quad p \,|\, p \supset H$$

where p ranges over the set of propositional variables. Note that for Horn formulas, in the vectorial notation $\vec{H} \supset p$, the vector components H_i are necessarily propositional variables, i. e., any Horn formula is of the form $\vec{q} \supset p$.

The Horn fragment is obtained by restricting sequents as follows:

1. contexts are restricted to *Horn contexts*, i. e., contexts where all formulas are Horn formulas;
2. term sequents are restricted to atomic sequents, i. e., term sequents are of the form $\Gamma \vdash t : p$.

As a consequence, the λ-abstraction construction and the rule *RIntro*, that types it, are no longer needed. The restricted typing rules are presented in Figure 2.

3 Coinductive representation of proof search in lambda-bar

We want to represent the whole search space for cut-free proofs in $\overline{\lambda}$. This is profitably done with coinductive structures. Of course, we only consider locally correct proofs. Since proof search may fail when infinite branches occur (depth-first search could be trapped there), we will consider such infinite proofs as proofs in an extended sense and represent them as well, thus we will introduce expressions that comprise all the possible well-founded and non-wellfounded proofs in cut-free $\overline{\lambda}$.

The raw syntax of these possibly non-wellfounded proofs is presented as follows

$$N ::=_{co} \lambda x^A.N \,|\, x\langle N_1, \ldots, N_k\rangle \quad ,$$

yielding the (co)terms of system $\overline{\lambda}^{co}$ (read coinductively, as indicated by the index *co*). Note that instead of a formal class of lists l as in the $\overline{\lambda}$-system, we adopt here the more intuitive notation $\langle N_1, \ldots, N_k\rangle$ to represent finite lists.

Since the raw syntax is interpreted coinductively, also the typing rules have to be interpreted coinductively, which is symbolized by the double horizontal line in Figure 3, a notation that we learnt from Nakata, Uustalu and Bezem [9]. (Of course, the formulas/types stay inductive.) As expected, the restriction of the typing relation to the finite $\overline{\lambda}$-terms coincides with the typing relation of the $\overline{\lambda}$ system:

Lemma 3 *For any $t \in \overline{\lambda}$, $\Gamma \vdash t : A$ in $\overline{\lambda}$ iff $\Gamma \vdash t : A$ in $\overline{\lambda}^{co}$.*

Proof By induction on t, with the help of Lemma 2. □

Figure 3: Typing rules of $\overline{\lambda}^{co}$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A.t : A \supset B}\ \mathit{RIntro} \qquad \frac{(x : B_1, \ldots, B_k \supset p) \in \Gamma \quad \Gamma \vdash N_i : B_i, i = 1, \ldots, k}{\Gamma \vdash x\langle N_1, \ldots, N_k\rangle : p}\ \mathit{LVecIntro}$$

Figure 4: Extra typing rule of $\overline{\lambda}^{co}_{\Sigma}$ w. r. t. $\overline{\lambda}^{co}$

$$\frac{\Gamma \vdash E_i : p, i = 1, \ldots, n}{\Gamma \vdash E_1 + \cdots + E_n : p}\ \mathit{Alts}$$

Example 4 *Consider $\omega := \lambda f^{p \supset p}.\lambda x^p.N$ with $N = f\langle N\rangle$ of type p. This infinite term N is also denoted f^∞.*

It is quite common to describe elements of coinductive syntax by (systems of) fixed point equations. As a notation on the *meta-level* for unique solutions of fixed-point equations, we will use the binder ν for the solution, writing $\nu N.M$, where N typically occurs in the term M. Intuitively, $\nu N.M$ is the N s. t. $N = M$. (The letter ν indicates interpretation in coinductive syntax.)

Example 5 *ω of Example 4 can be written as $\lambda f^{p \supset p}.\lambda x^p.\nu N.f\langle N\rangle$. $\Gamma, f : p \supset p, x : p \vdash \nu N.f\langle N\rangle : p$ is seen coinductively, so we get $\Gamma \vdash \omega : (p \supset p) \supset p \supset p$.*

We now come to the representation of whole search spaces. The set of coinductive cut-free $\overline{\lambda}$-terms with finite numbers of elimination alternatives is denoted by $\overline{\lambda}^{co}_{\Sigma}$ and is given by the following grammar:

$$\begin{array}{lrcl} \text{(co-terms)} & N & ::=_{co} & \lambda x^A.N \,|\, E_1 + \cdots + E_n \\ \text{(elim. alternatives)} & E & ::=_{co} & x\langle N_1, \ldots, N_k\rangle \end{array}$$

where both $n, k \geq 0$ are arbitrary. Note that summands cannot be lambda-abstractions.[1] We will often use $\sum\limits_i E_i$ instead of $E_1 + \cdots + E_n$ if the dependency of E_i on i is clear, as well as the number of elements. Likewise, we write $\langle N_i\rangle_i$ instead of $\langle N_1, \ldots, N_k\rangle$. If $n = 0$, we write $\mathbb{O}$ for $E_1 + \cdots + E_n$. If $n = 1$, we write E_1 for $E_1 + \cdots + E_n$ (in particular this injects the category of elimination alternatives into the category of co-terms) and do as if $+$ was a binary operation on (co)terms. However, this will always have a unique reading in terms of our raw syntax of $\overline{\lambda}^{co}_{\Sigma}$. In particular, this reading makes $+$ associative and $\mathbb{O}$ its neutral element.

Co-terms of $\overline{\lambda}^{co}_{\Sigma}$ will also be called Böhm forests. Their coinductive typing rules are the ones of $\overline{\lambda}^{co}$, together with the rule given in Figure 4, where the sequents for (co)terms and elimination alternatives are not distinguished notationally.

Notice that $\Gamma \vdash \mathbb{O} : p$ for all Γ and p.

Below we consider sequents $\Gamma \Rightarrow A$ with Γ a context and A an implicational formula (corresponding to term sequents of $\overline{\lambda}$ without proof terms – in fact, $\Gamma \Rightarrow A$ is nothing but the pair consisting of Γ and A, but which is viewed as a problem description: to prove formula A in context Γ).

[1] The division into two syntactic categories also forbids the generation of an infinite sum (for which $n = 2$ would suffice had the categories for N and E been amalgamated).

Definition 6 *The function $\mathscr{S}$, which takes a sequent $\Gamma \Rightarrow A$ and produces a Böhm forest which is a coinductive representation of the sequent's solution space, is given corecursively as follows: In the case of an implication,*

$$\mathscr{S}(\Gamma \Rightarrow A \supset B) := \lambda x^A.\mathscr{S}(\Gamma, x : A \Rightarrow B) \ ,$$

since RIntro is the only way to prove the implication.

In the case of an atom p, for the definition of $\mathscr{S}(\Gamma \Rightarrow p)$, let $y_i : A_i$ be the i-th variable in Γ with A_i of the form $\vec{B_i} \supset p$. Let $\vec{B_i} = B_{i,1}, \dots, B_{i,k_i}$. Define $N_{i,j} := \mathscr{S}(\Gamma \Rightarrow B_{i,j})$. Then, $E_i := y_i\langle N_{i,j}\rangle_j$, and finally,

$$\mathscr{S}(\Gamma \Rightarrow p) := \sum_i E_i \ .$$

This is more sloppily written as

$$\mathscr{S}(\Gamma \Rightarrow p) := \sum_{y:\vec{B}\supset p \in \Gamma} y\langle \mathscr{S}(\Gamma \Rightarrow B_j)\rangle_j \ .$$

In this manner, we can even write the whole definition in one line:

$$\mathscr{S}(\Gamma \Rightarrow \vec{A} \supset p) := \lambda \vec{x} : \vec{A}. \sum_{y:\vec{B}\supset p \in \Delta} y\langle \mathscr{S}(\Delta \Rightarrow B_j)\rangle_j \quad \textit{with } \Delta := \Gamma, \vec{x} : \vec{A}$$

This is a well-formed definition: for every Γ and A, $\mathscr{S}(\Gamma \Rightarrow A)$ is a Böhm forest and as such rather a semantic object.

Lemma 7 *Given Γ and A, the typing $\Gamma \vdash \mathscr{S}(\Gamma \Rightarrow A) : A$ holds in $\overline{\lambda}^{co}_{\Sigma}$.*

Let us illustrate the function $\mathscr{S}$ at work with some examples.

Example 8 *We consider first the formula $A = (p \supset p) \supset p \supset p$ and the empty context. We have:*

$$\mathscr{S}(\Rightarrow (p \supset p) \supset p \supset p) = \lambda f^{p\supset p}.\lambda x^p.\mathscr{S}(f : p \supset p, x : p \Rightarrow p)$$

Now, observe that $\mathscr{S}(f : p \supset p, x : p \Rightarrow p) = f\langle \mathscr{S}(f : p \supset p, x : p \Rightarrow p)\rangle + x$. We identify $\mathscr{S}(f : p \supset p, x : p \Rightarrow p)$ as the solution for N of the equation $N = f\langle N\rangle + x$. Using ν as means to communicate solutions of fixed-point equations on the meta-level *as for $\overline{\lambda}^{co}$, we have*

$$\mathscr{S}(\Rightarrow (p \supset p) \supset p \supset p) = \lambda f^{p\supset p}.\lambda x^p.\nu N.f\langle N\rangle + x$$

By unfolding of the fixpoint and by making a choice at each of the elimination alternatives, we can collect *from this coterm as the finitary solutions of the sequent all the Church numerals ($\lambda f^{p\supset p}.\lambda x^p.f^n\langle x\rangle$ with $n \in \mathbb{N}_0$), together with the infinitary solution $\lambda f^{p\supset p}.\lambda x^p.f^\infty$, studied before as example for $\overline{\lambda}^{co}$ (corresponding to always making the f-choice at the elimination alternatives).*

Example 9 *We consider now an example in the Horn fragment. Let $\Gamma = x : p \supset q \supset p, y : q \supset p \supset q, z : p$ (again with $p \neq q$). Note that the solution spaces of p and q relative to this sequent are mutually dependent and they give rise to the following system of equations:*

$$\begin{array}{rcl} N_p & = & x\langle N_p, N_q\rangle + z \\ N_q & = & y\langle N_q, N_p\rangle \end{array}$$

Figure 5: Membership relations

$$\frac{\mathsf{mem}(M,N)}{\mathsf{mem}(\lambda x^A.M,\lambda x^A.N)} \qquad \frac{\mathsf{mem}_E(M,E_i)}{\mathsf{mem}(M,E_1+\cdots+E_n)}\ \text{(for some } i\text{)}$$

$$\frac{\mathsf{mem}(M_1,N_1) \quad \ldots \quad \mathsf{mem}(M_k,N_k)}{\mathsf{mem}_E(x\langle M_1,\ldots,M_k\rangle,x\langle N_1,\ldots,N_k\rangle)}$$

and so we have

$$\begin{array}{rcl}\mathscr{S}(\Gamma\Rightarrow p) & = & \nu N_p.x\langle N_p,\nu N_q.y\langle N_q,N_p\rangle\rangle+z\\ \mathscr{S}(\Gamma\Rightarrow q) & = & \nu N_q.y\langle N_q,\nu N_p.x\langle N_p,N_q\rangle+z\rangle\end{array}$$

Whereas for p we can collect one finite solution (z), for q we can only collect infinite solutions. Because in the Horn case the recursive calls of the $\mathscr{S}$ function are all relative to the same (initial) context, in this fragment the solution space of a sequent can always be expressed as a finite system of equations (one for each atom occurring in the sequent), see Theorem 15.

Example 10 *Let us consider one further example where $A=((((p\supset q)\supset p)\supset p)\supset q)\supset q$ (a formula that can be viewed as double negation of Pierce's law, when q is viewed as absurdity). We have the following (where in sequents we omit formulas on the LHS)*

$$\begin{array}{rcl}N_0 & = & \mathscr{S}(\Rightarrow A)=\lambda x^{(((p\supset q)\supset p)\supset p)\supset q}.N_1\\ N_1 & = & \mathscr{S}(x\Rightarrow q)=x\langle N_2\rangle\\ N_2 & = & \mathscr{S}\big(x\Rightarrow((p\supset q)\supset p)\supset p\big)=\lambda y^{(p\supset q)\supset p}.N_3\\ N_3 & = & \mathscr{S}(x,y\Rightarrow p)=y\langle N_4\rangle\\ N_4 & = & \mathscr{S}(x,y\Rightarrow p\supset q)=\lambda z^p.N_5\\ N_5 & = & \mathscr{S}(x,y,z\Rightarrow q)=x\langle N_6\rangle\\ N_6 & = & \mathscr{S}\big(x,y,z\Rightarrow((p\supset q)\supset p)\supset p\big)=\lambda y_1^{(p\supset q)\supset p}.N_7\\ N_7 & = & \mathscr{S}(x,y,z,y_1\Rightarrow p)=y\langle N_8\rangle+z+y_1\langle N_8\rangle\\ N_8 & = & \mathscr{S}(x,y,z,y_1\Rightarrow p\supset q)=\lambda z_1^p.N_9\\ N_9 & = & \mathscr{S}(x,y,z,y_1,z_1\Rightarrow q)\end{array}$$

Now, in N_9 observe that y,y_1 both have type $(p\supset q)\supset p$ and z,z_1 both have type p, and we are back at N_5 but with the duplicates y_1 of y and z_1 of z. Later, we will call this duplication phenomenon co-contraction, *and we will give a finitary description of N_0 and, more generally, of all $\mathscr{S}(\Gamma\Rightarrow A)$, see Theorem 24. Of course, by taking the middle alternative in N_7, we obtain a finite proof, showing that A is provable in $\overline{\lambda}$.*

We now define a membership semantics for co-terms and elimination alternatives of $\overline{\lambda}^{co}_{\Sigma}$ in terms of sets of (co)terms in $\overline{\lambda}^{co}$.

The *membership relations* $\mathsf{mem}(M,N)$ and $\mathsf{mem}_E(M,E)$ are contained in $\overline{\lambda}^{co}\times\overline{\lambda}^{co}_{\Sigma}$ and $\overline{\lambda}^{co}\times E\overline{\lambda}^{co}_{\Sigma}$ respectively (where $E\overline{\lambda}^{co}_{\Sigma}$ stands for the set of elimination alternatives of $\overline{\lambda}^{co}_{\Sigma}$) and are given coinductively by the rules in Fig. 5.

Proposition 11 *For any $N\in\overline{\lambda}^{co}$, $\mathsf{mem}(N,\mathscr{S}(\Gamma\Rightarrow A))$ iff $\Gamma\vdash N:A$ in $\overline{\lambda}^{co}$.*

Proof "If". Consider the relations

$$\begin{array}{l}R:=\{(N,\mathscr{S}(\Gamma\Rightarrow A))\mid\Gamma\vdash N:A\}\\ R_E:=\{(x\langle N_i\rangle_i,x\langle\mathscr{S}(\Gamma\Rightarrow B_i)\rangle_i)\mid(x:B_1,\ldots,B_k\supset p)\in\Gamma\wedge\Gamma\vdash x\langle N_1,\ldots,N_k\rangle:p\}\end{array}$$

It suffices to show that $R \subseteq \mathsf{mem}$, but this cannot be proven alone since mem and mem_E are defined simultaneously. We also prove $R_E \subseteq \mathsf{mem}_E$, and to prove both by coinduction on the membership relations, it suffices to show that the relations R, R_E are *backwards closed*, i. e.:

1. $(\lambda x^A.M, \lambda x^A.N) \in R$ implies $(M,N) \in R$;
2. $(M, E_1 + \cdots + E_n) \in R$ implies for some i, $(M, E_i) \in R_E$;
3. $(x\langle M_1,\dots,M_k\rangle, x\langle N_1,\dots,N_k\rangle) \in R_E$ implies for all i, $(M_i,N_i) \in R$

We illustrate one case. Consider $(N, \mathscr{S}(\Gamma \Rightarrow A)) \in R$, with $\mathscr{S}(\Gamma \Rightarrow A) = E_1 + \cdots + E_n$. We must show that, for some i, $(N,E_i) \in R_E$. From $\mathscr{S}(\Gamma \Rightarrow A) = E_1 + \cdots + E_n$, we must have $A = p$. Now, from $\Gamma \vdash N : p$, there must exist $(x : B_1,\dots,B_k \supset p) \in \Gamma$ and $N_1,...,N_k$ s. t. $N = x\langle N_1,\dots,N_k\rangle$. By definition of $\mathscr{S}(\Gamma \Rightarrow A)$, there is i s. t. $E_i = x\langle \mathscr{S}(\Gamma \Rightarrow B_1),\dots,\mathscr{S}(\Gamma \Rightarrow B_k)\rangle$.

"Only if". By coinduction on the typing relation of $\overline{\lambda}^{co}$. This is conceptually easier than the other direction since $\vdash$ is a single coinductively defined notion. We define a relation R for which it is sufficient to prove $R \subseteq \vdash$:

$$R := \{(\Gamma, N, A) \mid \mathsf{mem}(N, \mathscr{S}(\Gamma \Rightarrow A))\}$$

Proving $R \subseteq \vdash$ by coinduction amounts to showing that R is backwards closed – with respect to the typing relation of $\overline{\lambda}^{co}$, i. e., we have to show:

1. $(\Gamma, \lambda x^A.t, A \supset B) \in R$ implies $((\Gamma, x : A), t, B) \in R$;
2. $(\Gamma, x\langle N_1,\dots,N_k\rangle, p) \in R$ implies the existence of $B_1,\dots,B_k$ s. t. $(x : B_1,\dots,B_k \supset p) \in \Gamma$ and, for all $i = 1,\dots,k$, $(\Gamma, N_i, B_i) \in R$.

We show the second case (relative to rule *LVecIntro*). So, we have $\mathsf{mem}(N, \mathscr{S}(\Gamma \Rightarrow A))$ with $N = x\langle N_1,\dots,N_k\rangle$ and $A = p$, and we need to show that, for some $(x : B_1,\dots,B_k \supset p) \in \Gamma$, we have, for all i, $\mathsf{mem}(N_i, \mathscr{S}(\Gamma \Rightarrow B_i))$. Since $A = p$, $\mathscr{S}(\Gamma \Rightarrow A) = E_1 + \cdots + E_n$. Hence, the second rule for mem was used to infer $\mathsf{mem}(N, \mathscr{S}(\Gamma \Rightarrow A))$, i. e., there is a j s. t. $\mathsf{mem}_E(N, E_j)$. Therefore, $E_j = x\langle M_1,\dots,M_k\rangle$ with terms $M_1, \dots, M_k$, and, for all i, $\mathsf{mem}(N_i, M_i)$. By the definition of $\mathscr{S}(\Gamma \Rightarrow A)$, this means that there are formulas $B_1, \dots, B_k$ s. t. $(x : B_1,\dots,B_k \supset p) \in \Gamma$ and, for all i, $M_i = \mathscr{S}(\Gamma \Rightarrow B_i)$. □

Example 12 *Let us consider the case of Pierce's law that is not valid intuitionistically. We have (for $p \neq q$):*

$$\mathscr{S}(\Rightarrow ((p \supset q) \supset p) \supset p) = \lambda x^{(p\supset q)\supset p}.x\langle \lambda y^p.\mathbb{O}\rangle$$

The fact that we arrived at $\mathbb{O}$ and found no elimination alternatives on the way annihilates *the co-term and implies there are no terms in the solution space of $\Rightarrow ((p \supset q) \supset p) \supset p$ (hence no proofs, not even infinite ones).*

Corollary 13 (Adequacy of the co-inductive representation of proof search in $\overline{\lambda}$) *For any $t \in \overline{\lambda}$, we have* $\mathsf{mem}(t, \mathscr{S}(\Gamma \Rightarrow A))$ *iff $\Gamma \vdash t : A$ (where the latter is the inductive typing relation of $\overline{\lambda}$).*

Proof By the proposition above and Lemma 3. □

4 Finitary representation of proof search in lambda-bar

In the first section we define a calculus of finitary representations. In the third section we obtain our main result (Theorem 24): given $\Gamma \Rightarrow C$, there is a finitary representation of $\mathscr{S}(\Gamma \Rightarrow C)$ in the finitary calculus. To make the proof easier to understand, we first develop in the second section the particular case of the Horn fragment.

4.1 The finitary calculus

The set of inductive cut-free $\overline{\lambda}$-terms with finite numbers of elimination alternatives, and a fixpoint operator is denoted by $\overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ and is given by the following grammar (read inductively):

$$\begin{array}{lrcl} \text{(terms)} & N & ::= & \lambda x^A.N \,|\, \mathsf{gfp}\,X.E_1 + \cdots + E_n \,|\, X \\ \text{(elim. alternatives)} & E & ::= & x\langle N_1, \ldots, N_k\rangle \end{array}$$

where X is assumed to range over a countably infinite set of *fixpoint variables* (letters Y, Z will also be used to range over fixpoint variables that may also be thought of as meta-variables), and where both $n, k \geq 0$ are arbitrary. Below, when we refer to *finitary terms* we have in mind the terms of $\overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$. The fixed-point operator is called gfp ("greatest fixed point") to indicate that its semantics is (now) defined in terms of infinitary syntax, but there, fixed points are unique. Hence, the reader may just read this as "the fixed point".

We now give a straightforward interpretation of the formal fixed points (built with gfp) of $\overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ in terms of the coinductive syntax of $\overline{\lambda}_{\Sigma}^{co}$ (using the ν operation on the meta-level).

Definition 14 *We call* environment *a function from the set of fixpoint variables into the set of (co)terms of $\overline{\lambda}_{\Sigma}^{co}$. The interpretation of a finitary term (relative to an environment) is a (co)term of $\overline{\lambda}_{\Sigma}^{co}$ given via a family of functions $[\![-]\!]_{\xi} : \overline{\lambda}_{\Sigma}^{\mathsf{gfp}} \to \overline{\lambda}_{\Sigma}^{co}$ indexed by environments, which is recursively defined as follows:*

$$\begin{array}{rcl} [\![X]\!]_{\xi} & = & \xi(X) \\ [\![\lambda x^A.N]\!]_{\xi} & = & \lambda x^A.[\![N]\!]_{\xi} \\ [\![\mathsf{gfp}\,X.\sum_i E_i]\!]_{\xi} & = & \nu N.\sum_i [\![E_i]\!]_{\xi\cup[X\mapsto N]} \\ [\![x\langle N_1,\ldots,N_k\rangle]\!]_{\xi} & = & x\langle [\![N_1]\!]_{\xi},\ldots,[\![N_k]\!]_{\xi}\rangle \end{array}$$

where the notation $\xi \cup [X \mapsto N]$ stands for the environment obtained from ξ by setting X to N.

Remark that the recursive definition above has an embedded corecursive case (pertaining to the gfp-operator). Its definition is well-formed since every elimination alternative starts with a head/application variable and the occurrences of N are thus guarded.

When a finitary term N has no free occurrences of fixpoint variables, all environments determine the same coterm, and in this case we simply write $[\![N]\!]$ to denote that coterm.

4.2 Equivalence of the representations: Horn case

Theorem 15 (Equivalence for the Horn fragment) *Let Γ be a Horn context. Then, for any atom r, there exists $N_r \in \overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ with no free occurrences of fixpoint variables such that $[\![N_r]\!] = \mathscr{S}(\Gamma \Rightarrow r)$.*

Proof

Let us assume there are k atoms occurring in $\Gamma \Rightarrow r$. We define simultaneously k functions $N_p(\overrightarrow{X:q})$ (one for each atom p occurring in $\Gamma \Rightarrow r$), parameterized by a vector of declarations of the form $X : q$. The vector is written $\overrightarrow{X:q}$ and is such that no fixpoint variable and no atom occurs twice. The simultaneous definition is by recursion on the number of atoms of $\Gamma \Rightarrow r$ not occurring in $\overrightarrow{X:q}$, and is as follows:

$$N_p(\overrightarrow{X:q}) = \begin{cases} X_i & \text{if } p = q_i \\ \mathsf{gfp}\,X_p. \sum\limits_{(y:\vec{r}\supset p)\in\Gamma} y\langle N_{r_j}(\overrightarrow{X:q}, X_p : p)\rangle_j & \text{otherwise} \end{cases}$$

where vector $\overrightarrow{X:q}, X_p : p$ is obtained by adding the component $X_p : p$ to the vector $\overrightarrow{X:q}$. Observe that only fixpoint variables among the fixpoint variables declared in the vector have free occurrences in $N_p(\overrightarrow{X:q})$.

By induction on the number of atoms of (the fixed sequent) $\Gamma \Rightarrow r$ not in (the variable) $\overrightarrow{X:q}$, we prove that:

$$[\![N_p(\overrightarrow{X:q})]\!]_\xi = \mathscr{S}(\Gamma \Rightarrow p) \text{ if } \xi(X_i) = \mathscr{S}(\Gamma \Rightarrow q_i), \text{ for any } i. \tag{1}$$

Case $p = q_i$, for some i. Then,

$$LHS = [\![X_i]\!]_\xi = \xi(X_i) = \mathscr{S}(\Gamma \Rightarrow q_i) = RHS.$$

Otherwise,

$$LHS = [\![\mathsf{gfp}\, X_p . \sum_{(y:\vec{r}\supset p)\in\Gamma} y\langle N_{r_j}(\overrightarrow{X:q}, X_p : p)\rangle_j]\!]_\xi = N^\infty$$

where N^∞ is given as the unique solution of the following equation:

$$N^\infty = \sum_{(y:\vec{r}\supset p)\in\Gamma} y\langle [\![N_{r_j}(\overrightarrow{X:q}, X_p : p)]\!]_{\xi\cup[X_p \mapsto N^\infty]}\rangle_j \tag{2}$$

Now observe that, by I.H., the following equations (3) and (4) are equivalent.

$$\mathscr{S}(\Gamma \Rightarrow p) = \sum_{(y:\vec{r}\supset p)\in\Gamma} y\langle [\![N_{r_j}(\overrightarrow{X:q}, X_p : p)]\!]_{\xi\cup[X_p \mapsto \mathscr{S}(\Gamma\Rightarrow p)]}\rangle_j \tag{3}$$

$$\mathscr{S}(\Gamma \Rightarrow p) = \sum_{(y:\vec{r}\supset p)\in\Gamma} y\langle \mathscr{S}(\Gamma \Rightarrow r_j)\rangle_j \tag{4}$$

By definition of $\mathscr{S}(\Gamma \Rightarrow p)$, (4) holds; hence – because of (3) – $\mathscr{S}(\Gamma \Rightarrow p)$ is the solution N^∞ of (2), concluding the proof that $LHS = RHS$.

Finally, the theorem follows as the particular case of (1) where $p = r$ and the vector of fixpoint variable declarations is empty. □

4.3 Equivalence of the representations: full implicational case

The main difference with exhaustive proof search in the case of Horn formulas is that the backwards application of *RIntro* brings new variables into the context that may have the same type as an already existing declaration, and so, for the purpose of proof search, they should be treated the same way.

We illustrate this phenomenon with the following definition and lemma and then generalize it to the form that will be needed for the main theorem (Theorem 24).

Definition 16 *For N and E in $\overline{\lambda}^{co}_{\Sigma}$, we define $[x_1 + \cdots + x_n/y]N$ and $[x_1 + \cdots + x_n/y]E$ by simultaneous corecursion as follows:*

$$\begin{array}{rcll}
[x_1+\cdots+x_n/y](\lambda x^A.N) & = & \lambda x^A.[x_1+\cdots+x_n/y]N & \\
[x_1+\cdots+x_n/y]\sum\limits_i E_i & = & \sum\limits_i [x_1+\cdots+x_n/y]E_i & \\
[x_1+\cdots+x_n/y]\big(z\langle N_i\rangle_i\big) & = & z\langle [x_1+\cdots+x_n/y]N_i\rangle_i & \textit{if } z \neq y \\
[x_1+\cdots+x_n/y]\big(y\langle N_i\rangle_i\big) & = & \sum\limits_{1\leq j\leq n} x_j\langle [x_1+\cdots+x_n/y]N_i\rangle_i &
\end{array}$$

Lemma 17 (Co-contraction: invertibility of contraction) *If $x_1, x_2, y \notin \Gamma$, then*

$$\mathscr{S}(\Gamma, x_1 : A, x_2 : A \Rightarrow C) = [x_1 + x_2/y]\mathscr{S}(\Gamma, y : A \Rightarrow C) \ .$$

Proof The proof is omitted since Lemma 20 below is essentially a generalization of this result. □

We now capture when a context Γ' is an inessential extension of context Γ:

Definition 18 *1.* $|\Gamma| = \{A : \exists x \ s.t.(x : A) \in \Gamma\}$.

2. $\Gamma \leq \Gamma'$ *if* $\Gamma \subseteq \Gamma'$ *and* $|\Gamma| = |\Gamma'|$.

3. $(\Gamma \Rightarrow p) \leq (\Gamma' \Rightarrow p')$ *if* $\Gamma \leq \Gamma'$ *and* $p = p'$.

Let σ range over sequents of the form $\Gamma \Rightarrow p$. Thus, the last definition clause defines in general when $\sigma \leq \sigma'$.

Definition 19 *1. Let $\Gamma \leq \Gamma'$. For N and E in $\overline{\lambda}^{co}_{\Sigma}$, we define $[\Gamma'/\Gamma]N$ and $[\Gamma'/\Gamma]E$ by simultaneous corecursion as follows:*

$$\begin{array}{lcll}
[\Gamma'/\Gamma](\lambda x^A.N) & = & \lambda x^A.[\Gamma', (x:A)/\Gamma, (x:A)]N & \\
[\Gamma'/\Gamma]\sum\limits_i E_i & = & \sum\limits_i [\Gamma'/\Gamma]E_i & \\
[\Gamma'/\Gamma]\big(z\langle N_i\rangle_i\big) & = & z\langle[\Gamma'/\Gamma]N_i\rangle_i & \mathit{if}\ z \notin \mathit{dom}(\Gamma) \\
[\Gamma'/\Gamma]\big(z\langle N_i\rangle_i\big) & = & \sum\limits_{(w:\Gamma(z))\in\Gamma'} w\langle[\Gamma'/\Gamma]N_i\rangle_i & \mathit{if}\ z \in \mathit{dom}(\Gamma)
\end{array}$$

2. Let $\sigma \leq \sigma'$. $[\sigma'/\sigma]N = [\Gamma'/\Gamma]N$ where $\sigma = (\Gamma \Rightarrow p)$ and $\sigma' = (\Gamma' \Rightarrow p)$. Similarly for $[\sigma'/\sigma]E$.

Lemma 20 (Co-contraction) *If $\Gamma \leq \Gamma'$ then $\mathscr{S}(\Gamma' \Rightarrow C) = [\Gamma'/\Gamma](\mathscr{S}(\Gamma \Rightarrow C))$.*

Proof Let $R := \{(\mathscr{S}(\Gamma' \Rightarrow C), [\Gamma'/\Gamma](\mathscr{S}(\Gamma \Rightarrow C))) \mid \Gamma \leq \Gamma', C \text{ arbitrary}\}$. We prove that R is backward closed relative to the canonical equivalence $=$ generated by the coinductive definition of terms of $\overline{\lambda}^{co}_{\Sigma}$ (but see the comments following the proof), whence $R \subseteq =$.

$$\mathscr{S}(\Gamma' \Rightarrow C) = \lambda z_1^{A_1} \cdots z_n^{A_n}. \sum_{(z:\vec{B}\supset p)\in\Delta'} z\langle \mathscr{S}(\Delta' \Rightarrow B_j)\rangle_j \tag{5}$$

and

$$[\Gamma'/\Gamma](\mathscr{S}(\Gamma \Rightarrow C)) = \lambda z_1^{A_1} \cdots z_n^{A_n}. \sum_{(y:\vec{B}\supset p)\in\Delta} \ \sum_{(w:\Delta(y))\in\Delta'} w\langle [\Delta'/\Delta]\mathscr{S}(\Delta \Rightarrow B_j)\rangle_j \tag{6}$$

where $\Delta := \Gamma \cup \{z_1 : A_1, \cdots, z_n : A_n\}$ and $\Delta' := \Gamma' \cup \{z_1 : A_1, \cdots, z_n : A_n\}$.

From $\Gamma \leq \Gamma'$ we get $\Delta \leq \Delta'$, hence

$$(\mathscr{S}(\Delta' \Rightarrow B_j), [\Delta'/\Delta]\mathscr{S}(\Delta \Rightarrow B_j)) \in R \ .$$

To conclude the proof, it suffices to show that (i) each head-variable z that is a "capability" of the summation in (5) is matched by a head-variable w that is a "capability" of the summation in (6); and (ii) vice-versa.

(i) Let $z \in \mathit{dom}(\Delta')$. We have to exhibit $y \in \mathit{dom}(\Delta)$ such that $(z : \Delta(y)) \in \Delta'$. First case: $z \in \mathit{dom}(\Delta)$. By $\Delta \leq \Delta'$, $(z : \Delta(z)) \in \Delta'$. So we may take $y = z$. Second and last case: $z \in \Gamma' \backslash \Gamma$. By $\Gamma \leq \Gamma'$, there is $y \in \Gamma$ such that $(z : \Gamma(y)) \in \Gamma'$. But then $(z : \Delta(y)) \in \Delta'$.

(ii) We have to show that, for all $y \in \mathit{dom}(\Delta)$, and all $(w : \Delta(y)) \in \Delta'$, $w \in \mathit{dom}(\Delta')$. But this is immediate. □

Notice that we cannot expect that the summands appear in the same order in (5) and (6). Therefore, we have to be more careful with the notion of equality of Böhm forests. It is not just bisimilarity, but we assume that the sums of elimination alternatives are treated as if they were sets of alternatives, i. e., we further assume that $+$ is symmetric and idempotent. It has been shown by Picard and the second author [10] that bisimulation up to permutations in unbounded lists of children can be managed in a coinductive type even with the interactive proof assistant Coq. In analogy, this coarser notion of equality (even abstracting away from the number of occurrences of an alternative) should not present a major obstacle for a fully formal presentation.

In the rest of the paper – in particular in Theorem 24 – we assume that sums of alternatives are treated as if they were sets.

Example 21 (Example 10 continued) *Thanks to the preceding lemma, N_9 is obtained by co-contraction from N_5:*

$$N_9 = [x:\cdot, y:(p\supset q)\supset p, z:p, y_1:(p\supset q)\supset p, z_1:p\,/\,x:\cdot, y:(p\supset q)\supset p, z:p]N_5 \ \ ,$$

where the type of x has been omitted. Hence, N_6, N_7, N_8 and N_9 can be eliminated, and N_5 can be expressed as the (meta-level) fixed point:

$$N_5 = \nu N.x\langle\lambda y_1^{(p\supset q)\supset p}.y\langle\lambda z_1^p.[x,y,z,y_1,z_1/x,y,z]N\rangle + z + y_1\langle\lambda z_1^p.[x,y,z,y_1,z_1/x,y,z]N\rangle\rangle \ \ ,$$

now missing out all types in the context substitution. Finally, we obtain the closed Böhm forest

$$\mathscr{S}(\Rightarrow A) = \lambda x^{(((p\supset q)\supset p)\supset p)\supset q}.x\langle\lambda y^{(p\supset q)\supset p}.y\langle\lambda z^p.N_5\rangle\rangle$$

The question is now how to give a finitary meaning to terms like N_5 in the example above, which are defined by fixed points over variables subject to context substitution. We might expect to use the equation defining N_5 to obtain a finitary representation in $\overline{\lambda}^{\mathsf{gfp}}_\Sigma$, provided context substitution is defined on this system. But how to do that? Applying say $[x,y,z,y_1,z_1/x,y,z]$ to a plain fixed-point variable cannot make much sense.

The desired finitary representation in the full implicational case is obtained by adjusting the terms of $\overline{\lambda}^{\mathsf{gfp}}_\Sigma$ used in the Horn case as follows:

$$\text{(terms)} \qquad N \quad ::= \quad (\cdots)\,|\,\mathsf{gfp}X^\sigma.E_1 + \cdots + E_n\,|\,X^\sigma$$

Hence fixpoint variables are "typed" with *sequents* σ.

Different free occurrences of the same X may be "typed" with different σ's, as long as a lower bound of these σ's can be found w.r.t. $\leq$ (Definition 18).

Relatively to Definition 14, an environment ξ now assigns (co)terms N of $\overline{\lambda}^{co}_\Sigma$ to "typed" fixpoint variables X^σ, provided X does not occur with two different "types" in the domain of ξ, for all X; we also change the following clauses:

$$\begin{array}{rcll} [\![X^{\sigma'}]\!]_\xi & = & [\sigma'/\sigma]\xi(X^\sigma) & \quad\text{if } \sigma\leq\sigma' \\ [\![\mathsf{gfp}X^\sigma.\sum\limits_i E_i]\!]_\xi & = & \nu N.\sum\limits_i [\![E_i]\!]_{\xi\cup[X^\sigma\mapsto N]} & \end{array}$$

We will have to assign some default value to $X^{\sigma'}$ in case there is no such σ, but this will not play a role in the main result below.

Map $N_p(\overrightarrow{X:q})$ used in the proof of Theorem 15 is replaced by the following:

Definition 22 *Let $\Xi := \overrightarrow{X : \Theta \Rightarrow q}$ be a vector of $m \geq 0$ declarations $(X_i : \Theta_i \Rightarrow q_i)$ where no fixpoint variable and no sequent occurs twice. $N_{\Gamma\Rightarrow\vec{A}\supset p}(\Xi)$ is defined as follows:*

If, for some $1 \leq i \leq m$, $p = q_i$ and $\Theta_i \subseteq \Gamma$ and $|\Theta_i| = |\Delta|$, then

$$N_{\Gamma\Rightarrow\vec{A}\supset p}(\Xi) = \lambda z_1^{A_1} \cdots z_n^{A_n}.X_i^{\sigma}$$

otherwise,

$$N_{\Gamma\Rightarrow\vec{A}\supset p}(\Xi) = \lambda z_1^{A_1} \cdots z_n^{A_n}.\mathsf{gfp}\, Y^{\sigma}. \sum_{(y:\vec{B}\supset p)\in\Delta} y\langle N_{\Delta\Rightarrow B_j}(\Xi, Y : \sigma)\rangle_j$$

where, in both cases, $\Delta := \Gamma \cup \{z_1 : A_1, \cdots, z_n : A_n\}$ and $\sigma := \Delta \Rightarrow p$.

The definition of $N_p(\overrightarrow{X : q})$ in the proof of Theorem 15 was by recursion on a certain number of atoms. The following lemma spells out the measure that is recursively decreasing in the definition of $N_{\Gamma\Rightarrow C}(\Xi)$.

Lemma 23 *For all $\Gamma \Rightarrow C$, $N_{\Gamma\Rightarrow C}(\cdot)$ is well-defined, where $\cdot$ denotes the empty vector.*

Proof Let us call *recursive call* a "reduction"

$$N_{\Gamma\Rightarrow\vec{A}\supset p}(\overrightarrow{X : \Theta \Rightarrow q}) \rightsquigarrow N_{\Delta\Rightarrow B_j}(\overrightarrow{X : \Theta \Rightarrow q}, Y : \sigma) \tag{7}$$

where the if-guard in Def. 22 fails; Δ and σ are defined as in the same definition; and, for some y, $(y : \vec{B} \supset p) \in \Delta$. We want to prove that every sequence of recursive calls from $N_{\Gamma\Rightarrow C}(\cdot)$ is finite.

First we introduce some definitions. $\mathscr{A}^{sub} := \{B \mid \text{there is } A \in \mathscr{A} \text{ such that } B \text{ is subformula of } A\}$, for $\mathscr{A}$ a finite set of formulas. We say $\mathscr{A}$ is *subformula-closed* if $\mathscr{A}^{sub} = \mathscr{A}$. A *stripped sequent* is a pair $(\mathscr{B}, p)$, where $\mathscr{B}$ is a finite set of formulas. If $\sigma = \Gamma \Rightarrow p$, then $|\sigma|$ denotes the stripped sequent $(|\Gamma|, p)$. We say $(\mathscr{B}, p)$ *is over* $\mathscr{A}$ if $\mathscr{B} \subseteq \mathscr{A}$ and $p \in \mathscr{A}$. There are $size(\mathscr{A}) := a \cdot 2^k$ stripped sequents over $\mathscr{A}$, if a (resp. k) is the number of atoms (resp. formulas) in $\mathscr{A}$.

Let $\mathscr{A}$ be subformula-closed. We say $\Gamma \Rightarrow C$ and $\Xi := \overrightarrow{X : \Theta \Rightarrow q}$ satisfy the *$\mathscr{A}$-invariant* if:

(i) $|\Gamma| \cup \{C\} \subseteq \mathscr{A}$;

(ii) $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_m = \Gamma$ (if $m = 0$ then this is meant to be vacuously true);

(iii) For $1 \leq j \leq m$, $q_j \in |\Gamma|^{sub}$,

where $m \geq 0$ is the length of vector Ξ (if $m = 0$, also item (iii) is vacuously true). In particular, $|\sigma|$ is over $\mathscr{A}$, for all $\sigma \in \Xi$. We prove that, if $\Gamma \Rightarrow C$ and Ξ satisfy the $\mathscr{A}$-invariant for some $\mathscr{A}$, then every sequence of recursive calls from $N_{\Gamma\Rightarrow C}(\Xi)$ is finite. The proof is by induction on $size(\mathscr{A}) - size(\Xi)$, where $size(\Xi)$ is the number of elements of $|\Xi|$ and $|\Xi| := \{|\sigma| : \sigma \in \Xi\}$.

Let $C = \vec{A} \supset p$. We analyze an arbitrary recursive call (7) and prove that every sequence of recursive calls from $N_{\Delta\Rightarrow B_j}(\Xi, Y : \sigma)$ is finite. This is achieved by proving:

(I) $\Delta \Rightarrow B_j$ and $\Xi, Y : \sigma$ satisfy the $\mathscr{A}$-invariant;

(II) $size(\Xi, Y : \sigma) > size(\Xi)$.

Proof of (I). By assumption, (i), (ii), and (iii) above hold. We want to prove:

(i') $|\Delta| \cup \{B_j\} \subseteq \mathscr{A}$;

(ii') $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_m \subseteq \Delta = \Delta$;

(iii') For $1 \leq j \leq m+1$, $q_j \in |\Delta|^{sub}$.

Proof of (i'). $|\Delta| = |\Gamma| \cup \{A_1, \cdots, A_n\} \subseteq \mathscr{A}$ by (i) and $\mathscr{A}$ subformula-closed. B_j is a subformula of $\vec{B} \supset p$ and $\vec{B} \supset p \in |\Delta|$ because $(y : \vec{B} \supset p) \in \Delta$, for some y.

Proof of (ii'). Immediate by (ii) and $\Gamma \subseteq \Delta$.

Proof of (iii'). For $1 \leq j \leq m$, $q_j \in |\Gamma|^{sub} \subseteq |\Delta|^{sub}$, by (iii) and $\Gamma \subseteq \Delta$. On the other hand, $q_{j+1} = p \in |\Delta|^{sub}$ because $(y : \vec{B} \supset p) \in \Delta$, for some y.

Proof of (II). Given that the if-guard of Def. 22 fails, and that $\Theta_i \subseteq \Gamma$ due to (ii), we conclude: for all $1 \leq i \leq m$, $p \neq q_i$ or $|\Theta_i| \neq |\Delta|$. But this means that $|\Delta \Rightarrow p| \notin |\Xi|$, hence $size(\Xi, Y : \sigma) > size(\Xi)$.

Now, by I.H., every sequence of recursive calls from $N_{\Delta \Rightarrow B_j}(\Xi, Y : \sigma)$ is finite. This concludes the proof by induction.

Finally let $\mathscr{A} = (|\Gamma| \cup \{C\})^{sub}$ and observe that $\Gamma \Rightarrow C$ and $\Xi = \cdot$ satisfy the $\mathscr{A}$-invariant. □

Theorem 24 (Equivalence) *For any Γ and C, there exists $N_{\Gamma \Rightarrow C} \in \overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ with no free occurrences of fixpoint variables such that $[\![N_{\Gamma \Rightarrow C}]\!] = \mathscr{S}(\Gamma \Rightarrow C)$.*

Proof We prove: if, for all i, $\xi(X_i^{\Theta_i \Rightarrow q_i}) = \mathscr{S}(\Theta_i \Rightarrow q_i)$, then

$$[\![N_{\Gamma \Rightarrow \vec{A} \supset p}(\Xi)]\!]_{\xi} = \mathscr{S}(\Gamma \Rightarrow \vec{A} \supset p) \ , \tag{8}$$

where $\Xi := \overrightarrow{X : \Theta \Rightarrow q}$. In this proof we re-use the concepts introduced in the proof of Lemma 23. Let $\mathscr{A} := (|\Gamma| \cup \{\vec{A} \supset p\})^{sub}$. The proof is by induction on $size(\mathscr{A}) - size(\Xi)$.

Case $p = q_i$ and $\Theta_i' \subseteq \Gamma$ and $|\Theta_i'| = |\Delta|$, for some $1 \leq i \leq m$, with m the length of Ξ. Then,

$$\begin{array}{rcll}
LHS & = & \lambda z_1^{A_1} \cdots z_n^{A_n}.[\![X_i^{\Delta \Rightarrow q_i}]\!]_{\xi} & \text{(by definition)} \\
& = & \lambda z_1^{A_1} \cdots z_n^{A_n}.[\Delta \Rightarrow q_i / \Theta_i \Rightarrow q_i]\xi(X_i^{\Theta_i \Rightarrow q_i}) & \text{(by definition and (*) below)} \\
& = & \lambda z_1^{A_1} \cdots z_n^{A_n}.[\Delta \Rightarrow q_i / \Theta_i \Rightarrow q_i]\mathscr{S}(\Theta_i \Rightarrow q_i) & \text{(by assumption)} \\
& = & \lambda z_1^{A_1} \cdots z_n^{A_n}.\mathscr{S}(\Delta \Rightarrow q_i) & \text{(by Lemma 20 and (*))} \\
& = & RHS & \text{(by definition)}
\end{array}$$

where $\Delta := \Gamma \cup \{z_1 : A_1, \cdots, z_n : A_n\}$, which implies $(\Theta_i \Rightarrow q_i) \leq (\Delta \Rightarrow q_i)$. The latter fact is the justification (*) used above.

The inductive case is an easy extension of the inductive case in Theorem 15. Suppose the case above holds for no $1 \leq i \leq m$. Then $LHS = \lambda z_1^{A_1} \cdots z_n^{A_n}.N^{\infty}$, where N^{∞} is the unique solution of the following equation

$$N^{\infty} \quad = \sum_{(y : \vec{B} \supset p) \in \Delta} y \langle [\![N_{\Delta \Rightarrow B_j}(\Xi, Y : \sigma)]\!]_{\xi \cup [Y^{\sigma} \mapsto N^{\infty}]} \rangle_j \tag{9}$$

and, again, $\Delta := \Gamma \cup \{z_1 : A_1, \cdots, z_n : A_n\}$. Now observe that, by I.H., the following equations (10) and (11) are equivalent.

$$\mathscr{S}(\Delta \Rightarrow p) \quad = \sum_{(y : \vec{B} \supset p) \in \Delta} y \langle [\![N_{\Delta \Rightarrow B_j}(\Xi, Y : \sigma)]\!]_{\xi \cup [Y^{\sigma} \mapsto \mathscr{S}(\Delta \Rightarrow p)]} \rangle_j \tag{10}$$

$$\mathscr{S}(\Delta \Rightarrow p) \quad = \sum_{(y : \vec{B} \supset p) \in \Delta} y \langle \mathscr{S}(\Delta \Rightarrow B_j) \rangle_j \tag{11}$$

By definition of $\mathscr{S}(\Delta \Rightarrow p)$, (11) holds; hence - because of (10) - $\mathscr{S}(\Delta \Rightarrow p)$ is the solution N^{∞} of (9). Therefore $LHS = \lambda z_1^{A_1} \cdots z_n^{A_n}.\mathscr{S}(\Delta \Rightarrow p)$, and the latter is RHS by definition of $\mathscr{S}(\Gamma \Rightarrow \vec{A} \supset p)$.

Finally, the theorem follows as the particular case of (8) where $C = \vec{A} \supset p$ and the vector of fixpoint variable declarations is empty. □

5 Conclusion

We proposed a coinductive approach to proof search, which we illustrated in the case of the cut-free system LJT for intuitionistic implication (and its proof-annotated version $\overline{\lambda}$). As the fundamental tool, we introduced the coinductive calculus $\overline{\lambda}_{\Sigma}^{co}$, which besides the coinductive reading of $\overline{\lambda}$, introduces a construction for finite alternatives. The (co)terms of this calculus (also called Böhm forests) are used to represent the solution space of proof search for LJT-sequents, and this is achieved by means of a corecursive function, whose definition arises naturally by taking a reductive view of the inference rules and by using the finite alternatives construction to account for multiple alternatives in deriving a given sequent.

We offered also a finitary representation of proof search in LJT, based on the inductive calculus $\overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ with finite alternatives and a fixed point construction, and showed equivalence of the representations. The equivalence results turned out to be an easy task in the case of the Horn fragment, but demanded for co-contraction of contexts (contraction bottom-up) in the case of full implication.

With Pym and Ritter [11] we share the general goal of setting a framework for studying proof search, and the reductive view of inference rules, by which each inference rule is seen as a reduction operator (from a putative conclusion to a collection of sufficient premises), and reduction (the process of repeatedly applying reduction operators) may fail to yield a (finite) proof. However, the methods are very different. Instead of using a coinductive approach, Pym and Ritter introduce the $\lambda\mu\nu\varepsilon$-calculus for classical sequent calculus as the means for representing derivations and for studying intuitionistic proof search (a task that is carried out both in the context of the sequent calculus LJ and of intuitionistic resolution).

In the context of logic programming with classical first-order Horn clauses, and building on their previous work [6, 4], Komendantskaya and Power [5] establish a coalgebraic semantics uniform for both finite and infinite SLD-resolutions. In particular, a notion of coinductive (and-or) derivation tree of an atomic goal w. r. t. a (fixed) program is introduced. Soundness and completeness results of SLD-resolution relative to coinductive derivation trees and to the coalgebraic semantics are also proved. Logic programming is viewed as search for uniform proofs in sequent calculus by Miller *et al.* [8]. For intuitionistic implication, uniform proofs correspond to the class of (η-)expanded normal natural deductions (see Dyckoff and Pinto [2]), hence to the typed $\overline{\lambda}$-terms we considered in this paper (recall the restriction to atoms in rule *Der* of Fig. 1 for typing application). Under this view, our work relates to Komendantskaya and Power [5], as both works adopt a coinductive approach in the context of proof search. However, the two approaches are different in methods and in goals. As the basis of the coinductive representation of the search space, instead of and-or infinite trees, we follow the Curry-Howard view of proofs as terms, and propose the use of a typed calculus of coinductive lambda-terms. Whereas Komendantskaya and Power [5] are already capable of addressing first-order quantification, we only consider intuitionistic implication. Still, as we consider full intuitionistic implication, our study is not contained in classical Horn logic. The fact that we need to treat negative occurrences of implication, raises on the logic programming side the need for dealing with programs to which clauses can be added dynamically.

As a priority for future work, we plan to develop notions of normalisation for the calculi $\overline{\lambda}_{\Sigma}^{co}$ and $\overline{\lambda}_{\Sigma}^{\mathsf{gfp}}$ in connection with aspects of proof search like pruning search spaces and reading off (finite) proofs.

In order to test for the generality of our approach, we intend to extend it to treat the first-order case. Staying within intuitionistic implication, but changing the proofs searched for, another case study we intend to investigate is Dyckhoff's contraction-free system [1].

Acknowledgments We thank our anonymous referees for their helpful comments. José Espírito Santo and Luís Pinto have been financed by FEDER funds through "Programa Operacional Factores de Competitividade – COMPETE" and by Portuguese funds through FCT – "Fundação para a Ciência e a Tecnologia", within the project PEst-C/MAT/UI0013/2011. Ralph Matthes thanks the Centro de Matemática of Universidade do Minho for funding research visits to José Espírito Santo and Luís Pinto to start this research (2011/2012). Subsequently, he has been funded by the *Climt* project (ANR-11-BS02-016 of the French Agence Nationale de la Recherche).

References

[1] Roy Dyckhoff (1992): *Contraction-Free Sequent Calculi for Intuitionistic Logic*. *J. Symb. Log.* 57(3), pp. 795–807, doi:10.2307/2275431.

[2] Roy Dyckhoff & Luís Pinto (1994): *Uniform Proofs and Natural Deductions*. In Didier Galmiche & Lincoln Wallen, editors: *Proceedings of CADE–12 Workshop on Proof Search in Type-Theoretic Languages*, INRIA Lorraine – CRIN, pp. 717–23. Available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.9659.

[3] H. Herbelin (1995): *A λ-calculus structure isomorphic to a Gentzen-style sequent calculus structure*. In L. Pacholski & J. Tiuryn, editors: *Proceedings of CSL'94*, *Lecture Notes in Computer Science* 933, Springer-Verlag, pp. 61–75, doi:10.1007/BFb0022247.

[4] Ekaterina Komendantskaya, Guy McCusker & John Power (2010): *Coalgebraic Semantics for Parallel Derivation Strategies in Logic Programming*. In Michael Johnson & Dusko Pavlovic, editors: *AMAST*, *Lecture Notes in Computer Science* 6486, Springer, pp. 111–127, doi:10.1007/978-3-642-17796-5_7.

[5] Ekaterina Komendantskaya & John Power (2011): *Coalgebraic Derivations in Logic Programming*. In Marc Bezem, editor: *CSL*, *LIPIcs* 12, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 352–366, doi:10.4230/LIPIcs.CSL.2011.352.

[6] Ekaterina Komendantskaya & John Power (2011): *Coalgebraic Semantics for Derivations in Logic Programming*. In Andrea Corradini, Bartek Klin & Corina Cîrstea, editors: *CALCO*, *Lecture Notes in Computer Science* 6859, Springer, pp. 268–282, doi:10.1007/978-3-642-22944-2_19.

[7] Chuck Liang & Dale Miller (2009): *Focusing and Polarization in Linear, Intuitionistic, and Classical Logic*. *Theoretical Computer Science* 410, pp. 4747–4768, doi:10.1016/j.tcs.2009.07.041.

[8] Dale Miller, Gopalan Nadathur, Frank Pfenning & Andre Scedrov (1991): *Uniform Proofs as a Foundation for Logic Programming*. *Annals of Pure and Applied Logic* 51(1-2), pp. 125–157, doi:10.1016/0168-0072(91)90068-W.

[9] Keiko Nakata, Tarmo Uustalu & Marc Bezem (2011): *A Proof Pearl with the Fan Theorem and Bar Induction - Walking through Infinite Trees with Mixed Induction and Coinduction*. In Hongseok Yang, editor: *APLAS*, *LNCS* 7078, Springer, pp. 353–368, doi:10.1007/978-3-642-25318-8_26.

[10] Celia Picard & Ralph Matthes (2012): *Permutations in Coinductive Graph Representation*. In Dirk Pattinson & Lutz Schröder, editors: *Coalgebraic Methods in Computer Science (CMCS 2012)*, *Lecture Notes in Computer Science, IFIP subseries* 7399, Springer, pp. 218–237, doi:10.1007/978-3-642-32784-1_12.

[11] D.J. Pym & E. Ritter (2004): *Reductive Logic and Proof-search: Proof Theory, Semantics, and Control*. Oxford Logic Guides, Oxford University Press, Incorporated, doi:10.1093/acprof:oso/9780198526339.001.0001.

Infinitary Axiomatization of the Equational Theory of Context-Free Languages

Niels Bjørn Bugge Grathwohl
Department of Computer Science (DIKU)
University of Copenhagen
Universitetsparken 5
DK-2100 Copenhagen, Denmark
bugge@diku.dk

Fritz Henglein
Department of Computer Science (DIKU)
University of Copenhagen
Universitetsparken 5
DK-2100 Copenhagen, Denmark
henglein@diku.dk

Dexter Kozen
Department of Computer Science
Cornell University
Ithaca, NY 14853-7501, USA
kozen@cs.cornell.edu

We give a natural complete infinitary axiomatization of the equational theory of the context-free languages, answering a question of Leiß (1992).

1 Introduction

Algebraic reasoning about programming language constructs has been a popular research topic for many years. At the propositional level, the theory of flowchart programs and linear recursion are well handled by such systems as Kleene algebra and iteration theories, systems that characterize the equational theory of the regular sets. To handle more general forms of recursion including procedures with recursive calls, one must extend to the context-free languages, and here the situation is less well understood. One reason for this is that, unlike the equational theory of the regular sets, the equational theory of the context-free languages is not recursively enumerable. This has led some researchers to declare its complete axiomatization an insurmountable task [13].

Whereas linear recursion can be characterized with the star operator * of Kleene algebra or the dagger operation † of iteration theories, the theory of context-free languages requires a more general fixpoint operator μ. The characterization of the context-free languages as least solutions of algebraic inequalities involving μ goes back to a 1971 paper of Gruska [7]. More recently, several researchers have given equational axioms for semirings with μ and have developed fragments of the equational theory of context-free languages [3, 5, 6, 8, 9, 13].

In this paper we consider another class of models satisfying a condition called *μ-continuity* analogous to the star-continuity condition of Kleene algebra:

$$a(\mu x.p)b = \sum_{n\geq 0} a(nx.p)b,$$

where the summation symbol denotes supremum with respect to the natural order in the semiring, and

$$0x.p = 0 \qquad\qquad (n+1)x.p = p[x/nx.p].$$

This infinitary axiom combines the assertions that $\mu x.p$ is the supremum of its finite approximants $nx.p$ and that multiplication in the semiring is continuous with respect to these suprema. Analogous to a

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 44–55, doi:10.4204/EPTCS.126.4

similar result for star-continuous Kleene algebra, we show that all context-free languages over a μ-continuous idempotent semiring have suprema. Our main result is that the μ-continuity condition, along with the axioms of idempotent semirings, completely axiomatize the equational theory of the context-free languages. This is the first completeness result for the equational theory of the context-free languages, answering a question of Leiß [13].

1.1 Related Work

Courcelle [3] investigates *regular systems*, finite systems of fixpoint equations over first-order terms over a ranked alphabet with a designated symbol $+$ denoting set union, thereby restricting algebras to power set algebras. He stages their interpretation by first interpreting recursion over first-order terms as infinite trees, essentially as the final object in the corresponding coalgebra, then interpreting the signature symbols in ω-complete algebras. He provides soundness and completeness for transforming regular systems that preserve all solutions and soundness, but not completeness for preserving their least solutions. Courcelle's approach is syntactic since it employs unfolding of terms in fixpoint equations.

Leiß [13] investigates three classes of idempotent semirings with a syntactic least fixpoint operator μ. The three classes are called KAF, KAR, and KAG in increasing order of specificity. All these classes are assumed to satisfy the fundamental *Park axioms*

$$p[x/\mu x.p] \le \mu x.p \qquad\qquad p \le x \;\Rightarrow\; \mu x.p \le x,$$

which say that $\mu x.p$ is the least solution of the inequality $p \le x$. The classes KAR and KAG further assume

$$\mu x.(b+ax) = \mu x.(1+xa)\cdot b \qquad\qquad \mu x.(b+xa) = b\cdot \mu x.(1+ax)$$

and

$$\mu x.(s+rx) = \mu x.(\mu y.(1+yr)\cdot s) \qquad\qquad \mu x.(s+xr) = \mu x.(s\cdot \mu y.(1+ry)),$$

respectively. These axioms can be viewed as imposing continuity properties of the semiring operators with respect to μ. All standard interpretations, including the context-free languages over an alphabet X, are continuous and satisfy the KAG axioms. Ésik and Leiß [5, 6] show that conversion to Greibach normal form can be performed purely algebraically under these assumptions.

Ésik and Kuich [4] introduce *continuous semirings*, which are required to have suprema for all directed sets, and they employ domain theory to solve polynomial fixpoint equations. Idempotent continuous semirings are μ-continuous Chomsky algebras as defined here, but not conversely. As we shall prove, the family of context-free languages over any alphabet constitutes a μ-continuous Chomsky algebra. It is not a continuous semiring, however, since the union of context-free languages is not necessarily context-free.

2 Chomsky Algebras

2.1 Polynomials

Let $(C, +, \cdot, 0, 1)$ be an idempotent semiring and X a fixed set of variables. A *polynomial over indeterminates X with coefficients in C* is an element of $C[X]$, where $C[X]$ is the coproduct (direct sum) of C and the free idempotent semiring on generators X in the category of idempotent semirings. For example, if $a,b,c \in C$ and $x,y \in X$, then the following are polynomials:

$$0 \qquad a \qquad axbycx+1 \qquad ax^2byx+by^2xc \qquad 1+x+x^2+x^3$$

The elements of $C[X]$ are not purely syntactic, as they satisfy all the equations of idempotent semirings and identities of C. For example, if $a^2 = b^2 = 1$ in C, then

$$(axa+byb)^2 = ax^2a+axabyb+bybaxa+by^2b.$$

Every polynomial can be written as a finite sum of monomials of the form

$$a_0x_0a_1x_1\cdots a_{n-1}x_{n-1}a_n,$$

where each $a_i \in C-\{0\}$ and $x_i \in X$. The *free variables* of such an expression p are the elements of X appearing in it and are denoted $\mathsf{FV}(p)$. The representation is unique up to associativity of multiplication and associativity, commutativity, and idempotence of addition.

2.2 Polynomial Functions and Evaluation

Let $C[X]$ be the semiring of polynomials over indeterminates X and let D be an idempotent semiring containing C as a subalgebra. By general considerations of universal algebra, any valuation $\sigma : X \to D$ extends uniquely to a semiring homomorphism $\hat{\sigma} : C[X] \to D$ preserving C pointwise. Formally, the functor $X \mapsto C[X]$ is left adjoint to a forgetful functor that takes an idempotent semiring D to its underlying set. Intuitively, $\hat{\sigma}$ is the *evaluation morphism* that evaluates a polynomial at the point $\sigma \in D^X$. Thus each polynomial $p \in C[X]$ determines a *polynomial function* $[\![p]\!] : D^X \to D$, where $[\![p]\!](\sigma) = \hat{\sigma}(p)$.

The set of all functions $D^X \to D$ with the pointwise semiring operations is itself an idempotent semiring with C as an embedded subalgebra under the embedding $c \mapsto \lambda\sigma.c$. The map $[\![\cdot]\!] : C[X] \to (D^X \to D)$ is actually $\hat{\tau}$, where $\tau(x) = \lambda f.f(x)$.

For the remainder of the paper, we write σ for $\hat{\sigma}$, as there is no longer any need to distinguish them.

2.3 Algebraic Closure and Chomsky Algebras

A *system of polynomial inequalities over* C is a set

$$p_1 \le x_1,\ p_2 \le x_2,\ \ldots,\ p_n \le x_n \tag{1}$$

where $x_i \in X$ and $p_i \in C[X]$, $1 \le i \le n$. A *solution* of (1) in C is a valuation $\sigma : X \to C$ such that $\sigma(p_i) \le \sigma(x_i)$, $1 \le i \le n$. The solution σ is a *least solution* if $\sigma \le \tau$ pointwise for any other solution τ. If a least solution exists, then it is unique.

An idempotent semiring C is said to be *algebraically closed* if every finite system of polynomial inequalities over C has a least solution in C.

The category of *Chomsky algebras* consists of algebraically closed idempotent semirings along with semiring homomorphisms that preserve least solutions of systems of polynomial inequalities.

The canonical example of a Chomsky algebra is the family of context-free languages $\mathsf{CF}\,X$ over an alphabet X. A system of polynomial inequalities (1) can be regarded as context-free grammar, and the least solution of the system is the context-free language generated by the grammar. For example, the set of strings in $\{a,b\}^\star$ with equally many a's and b's is generated by the grammar

$$S \to \varepsilon \mid aB \mid bA \qquad A \to aS \mid bAA \qquad B \to bS \mid aBB, \tag{2}$$

which corresponds to the system

$$1 + aB + bA \leq S \qquad aS + bAA \leq A \qquad bS + aBB \leq B, \tag{3}$$

where the symbols a, b are interpreted as the singleton sets $\{a\}, \{b\}$, the symbols S, A, B are variables ranging over sets of strings, and the semiring operations $+$, $\cdot$, 0, and 1 are interpreted as set union, set product $AB = \{xy \mid x \in A,\ y \in B\}$, $\emptyset$, and $\{\varepsilon\}$, respectively.

2.4 μ-Expressions

Let X be a set of indeterminates. Leiß [13] and Ésik and Leiß [5, 6] consider *μ-expressions* defined by the grammar

$$t ::= x \mid t + t \mid t \cdot t \mid 0 \mid 1 \mid \mu x.t$$

where $x \in X$. These expressions provide a syntax with which least solutions of polynomial systems can be named. Scope, bound and free occurrences of variables, α-conversion, and safe substitution are defined as usual (see e.g. [1]). We denote by $t[x/u]$ the result of substituting u for all free occurrences of x in t, renaming bound variables as necessary to avoid capture. Let $\mathsf{T}X$ denote the set of μ-expressions over indeterminates X.

Let C be a Chomsky algebra and X a set of indeterminates. An *interpretation* over C is a map $\sigma : \mathsf{T}X \to C$ that is a homomorphism with respect to the semiring operations and such that

$$\sigma(\mu x.t) = \text{the least } a \in C \text{ such that } \sigma[x/a](t) \leq a, \tag{4}$$

where $\sigma[x/a]$ denotes σ with x rebound to a. The element a exists and is unique: Informally, each μ-expression t can be associated with a system of polynomial inequalities such that $\sigma(t)$ is a designated component of its least solution, which exists by algebraic closure.

Every set map $\sigma : X \to C$ extends uniquely to such a homomorphism. An interpretation σ *satisfies* the equation $s = t$ if $\sigma(s) = \sigma(t)$ and satisfies the inequality $s \leq t$ if $\sigma(s) \leq \sigma(t)$. All interpretations over Chomsky algebras satisfy the axioms of idempotent semirings, α-conversion (renaming of bound variables), and the *Park axioms*

$$t[x/\mu x.t] \leq \mu x.t \qquad t \leq x \ \Rightarrow\ \mu x.t \leq x. \tag{5}$$

The Park axioms say intuitively that $\mu x.t$ is the least solution of the single inequality $t \leq x$. It follows easily that

$$t[x/\mu x.t] = \mu x.t. \tag{6}$$

Thus Chomsky algebras are essentially the ordered Park μ-semirings of [6] with the additional restriction that $+$ is idempotent and the order is the natural order $x \leq y \Leftrightarrow x + y = y$.

2.5 Bekić's Theorem

It is well known that the ability to name least solutions of single inequalities with μ gives the ability to name least solutions of all finite systems of inequalities. This is known as Bekić's theorem [2]. The construction is analogous to the definition of $M^\star$ for a matrix M over a Kleene algebra.

Bekić's theorem can be proved by regarding a system of inequalities as a single inequality on a Cartesian product, partitioning into two systems of smaller dimension, then applying the result for the 2×2 case inductively. The 2×2 system

$$p(x,y) \leq x \qquad\qquad q(x,y) \leq y$$

has least solution a_0, b_0, where

$$a(y) = \mu x.p(x,y) \qquad\qquad b_0 = \mu y.q(a(y),y) \qquad\qquad a_0 = a(b_0),$$

as can be shown using the Park axioms (5); see [14] or [6] for a comprehensive treatment.

For example, in the context-free languages, the set of strings in $\{a,b\}^\star$ with equally many a's and b's is represented by the term

$$\mu S.(1 + a \cdot \mu B.(bS + aBB) + b \cdot \mu A.(aS + bAA)) \tag{7}$$

obtained from the system (2) by this construction.

2.6 μ-Continuity

Let $nx.t$ be an abbreviation for the n-fold composition of t applied to 0, defined inductively by

$$0x.t = 0 \qquad\qquad (n{+}1)x.t = t[x/nx.t].$$

A Chomsky algebra is called *μ-continuous* if it satisfies the *μ-continuity axiom*:

$$a(\mu x.t)b = \sum_{n \geq 0} a(nx.t)b, \tag{8}$$

where the summation symbol denotes supremum with respect to the natural order $x \leq y \Leftrightarrow x + y = y$. Note that the supremum of a and b is $a + b$.

The family $\mathsf{CF}X$ of context-free languages over an alphabet X forms a μ-continuous Chomsky algebra. The *canonical interpretation* over this algebra is $L_X : \mathsf{T}X \to \mathsf{CF}X$, where

$$\begin{aligned} L_X(x) &= \{x\} & L_X(t+u) &= L_X(t) \cup L_X(u) \\ L_X(0) &= \emptyset & L_X(tu) &= \{xy \mid x \in L_X(t),\ y \in L_X(u)\} \\ L_X(1) &= \{\varepsilon\} & L_X(\mu x.t) &= \bigcup_{n \geq 0} L_X(nx.t). \end{aligned} \tag{9}$$

Under L_X, every term in $\mathsf{T}X$ represents a context-free language over its free variables (note that x is not free in $nx.t$). In the example (7) of §2.5, the free variables are a, b and the bound variables are S, A, B, corresponding to the terminal and nonterminal symbols, respectively, of the grammar (2) of §2.3.

2.7 Relation to Other Axiomatizations

In this section we show that the various axiomatizations considered in [5, 6, 13] are valid in all μ-continuous Chomsky algebras.

A *μ-semiring* [6] is a semiring $(A, +, \cdot, 0, 1)$ satisfying the *μ-congruence* and *substitution* properties:

$$t = u \Rightarrow \mu x.t = \mu x.u \qquad\qquad \sigma(t[y/u]) = \sigma[y/\sigma(u)](t).$$

Idempotence is not assumed.

Lemma 2.1. *Every Chomsky algebra is a μ-semiring.*

Proof. The μ-congruence property is immediate from the definition of the μ operation (4). The substitution property is a general property of systems with variable bindings; see [1, Lemma 5.1.5]. It can be proved by induction. For the case of $\mu x.t$, we assume without loss of generality that $y \neq x$ (otherwise there is nothing to prove) and that x is not free in u.

$$\begin{aligned}\sigma((\mu x.t)[y/u]) &= \sigma(\mu x.(t[y/u]))\\ &= \text{least } a \text{ such that } \sigma[x/a](t[y/u]) \leq a\\ &= \text{least } a \text{ such that } \sigma[x/a][y/\sigma(u)](t) \leq a\\ &= \text{least } a \text{ such that } \sigma[y/\sigma(u)][x/a](t) \leq a\\ &= \sigma[y/\sigma(u)](\mu x.t).\end{aligned}$$

□

We now consider various axioms proposed in [13].

Lemma 2.2. *In all μ-continuous Chomsky algebras,*

$$\mu x.(1+ax) = \mu x.(1+xa), \quad x \notin \mathsf{FV}(a).$$

Proof. By μ-continuity, it suffices to show that $nx.(1+ax) = nx.(1+xa)$ for all n. We show by induction that for all n, $nx.(1+ax) = nx.(1+xa) = \sum_{i=0}^{n} a^i$. The basis $n = 0$ is trivial. For the inductive case,

$$(n{+}1)x.(1+ax) = 1 + a(nx.(1+ax)) = 1 + a(\textstyle\sum_{i=0}^{n} a^i) = \sum_{i=0}^{n+1} a^i,$$

and this is equal to $(n{+}1)x.(1+xa)$ by a symmetric argument. □

Lemma 2.3. *The following two equations hold in all μ-continuous Chomsky algebras:*

$$a(\mu x.(1+xb)) = \mu x.(a+xb) \qquad\qquad (\mu x.(1+bx))a = \mu x.(a+bx).$$

Proof. We show the first equation only; the second follows from a symmetric argument. By μ-continuity, we need only show that the equation holds for any n. The basis $n = 0$ is trivial. For the inductive case,

$$\begin{aligned}a((n{+}1)x.(1+xb)) &= a + a(nx.(1+xb))b\\ &= a + (nx.(a+xb))b\\ &= (n{+}1)x.(a+xb),\end{aligned}$$

where the induction hypothesis has been used in the second step. □

These properties also show that μ-continuous Chomsky algebras are algebraically complete semirings in the sense of [5, 6].

Lemma 2.4. *The* Greibach inequalities

$$\mu x.s(\mu y.(1+ry)) \leq \mu x.(s+xr) \qquad\qquad \mu x.(\mu y.(1+yr))s \leq \mu x.(s+rx)$$

of KAG [13] *hold in all μ-continuous Chomsky algebras.*

Proof. For the left-hand inequality, let $u = \mu x.(s+xr)$. By the Park axioms, it suffices to show that $s(\mu y.(1+ry))[x/u] \leq u$. But

$$\begin{aligned} s(\mu y.(1+ry))[x/u] &= s[x/u](\mu y.(1+r[x/u]y)) \\ &= s[x/u](\mu y.(1+yr[x/u])) \\ &= \mu y.(s[x/u]+yr[x/u]) \\ &= \mu x.(s+xr), \end{aligned}$$

where Lemmas 2.2 and 2.3 have been used.

The right-hand ineuuality can be proved by a symmetric argument. □

Various other axioms of [5, 6, 13] follow from the Park axioms.

The μ-continuity condition (8) implies the Park axioms (5), but we must defer the proof of this fact until §3. For now we just observe a related property of the canonical interpretation L_X.

Lemma 2.5. *For any $s,t \in \mathsf{T}X$ and $y \in X$,*

$$L_X(s[y/\mu y.t]) = \bigcup_{n\geq 0} L_X(s[y/ny.t]).$$

Proof. We proceed by induction on the structure of s. The cases for $+$ and $\cdot$ are quite easy, using the facts that for chains of sets of strings $A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ and $B_0 \subseteq B_1 \subseteq B_2 \subseteq \cdots$,

$$\bigcup_m A_m \cup \bigcup_n B_n = \bigcup_n A_n \cup B_n \qquad \bigcup_m A_m \cdot \bigcup_n B_n = \bigcup_n A_n B_n.$$

The base cases are also straightforward. For $\mu x.s$, assume without loss of generality that $y \neq x$ and x is not free in t.

$$\begin{aligned} L_X((\mu x.s)[y/\mu y.t]) &= \bigcup_m L_X((mx.s)[y/\mu y.t]) \\ &= \bigcup_m \bigcup_n L_X((mx.s)[y/ny.t]) \\ &= \bigcup_n \bigcup_m L_X((mx.s)[y/ny.t]) \\ &= \bigcup_n L_X((\mu x.s)[y/ny.t]). \end{aligned}$$

□

3 Main Results

Our main result depends on an analog of a result of [10] (see [12]). It asserts that the supremum of a context-free language over a μ-continuous Chomsky algebra K exists, interpreting strings over K as products in K. Moreover, multiplication is continuous with respect to suprema of context-free languages.

Lemma 3.1. *Let $\sigma : \mathsf{T}X \to K$ be any interpretation over a μ-continuous Chomsky algebra K. Let $\tau : \mathsf{T}X \to \mathsf{CF}X$ be any interpretation over the context-free languages $\mathsf{CF}X$ such that for all $x \in X$ and $s,u \in \mathsf{T}X$,*

$$\sigma(sxu) = \sum_{y\in\tau(x)} \sigma(syu).$$

Then for any $s,t,u \in \mathsf{T}X$,

$$\sigma(stu) = \sum_{y \in \tau(t)} \sigma(syu).$$

In particular,

$$\sigma(stu) = \sum_{y \in L_X(t)} \sigma(syu), \tag{10}$$

where L_X is the canonical interpretation defined in §2.6.

Remark 1. Note carefully that the lemma does not assume *a priori* knowledge of the existence of the suprema. The equations should be interpreted as asserting that the supremum on the right-hand side exists and is equal to the expression on the left-hand side.

Proof. The proof is by induction on the structure of t, that is by induction on the subexpression relation $t+u \succ t, t+u \succ u, t \cdot u \succ t, t \cdot u \succ u, \mu x.t \succ nx.t$, which is well-founded [11].

All cases are similar to the proof in [12, Lemma 7.1] for star-continuous Kleene algebra, with the exception of the case $t = \mu x.p$.

For variables $t = x \in X$, the desired property holds by assumption. For the constants $t = 0$ and $t = 1$,

$$\begin{aligned}
\sigma(s0u) &= 0 = \sum \emptyset = \sum_{y \in \emptyset} \sigma(syu) = \sum_{y \in \tau(0)} \sigma(syu) \\
\sigma(s1u) &= \sigma(su) = \sum_{y \in \{\varepsilon\}} \sigma(syu) = \sum_{y \in \tau(1)} \sigma(syu).
\end{aligned}$$

For sums $t = p + q$,

$$\begin{aligned}
\sigma(s(p+q)u) &= \sigma(spu) + \sigma(squ) \\
&= \sum_{x \in \tau(p)} \sigma(sxu) + \sum_{y \in \tau(q)} \sigma(syu) && (11) \\
&= \sum_{z \in \tau(p) \cup \tau(q)} \sigma(szu) && (12) \\
&= \sum_{z \in \tau(p+q)} \sigma(szu). && (13)
\end{aligned}$$

Equation (11) is by two applications of the induction hypothesis. Equation (12) is by the properties of supremum. Equation (13) is by the definition of sum in $\mathsf{CF}X$.

For products $t = pq$,

$$\begin{aligned}
\sigma(spqu) &= \sum_{x \in \tau(p)} \sum_{y \in \tau(q)} \sigma(sxyu) && (14) \\
&= \sum_{z \in \tau(p) \cdot \tau(q)} \sigma(szu) && (15) \\
&= \sum_{z \in \tau(pq)} \sigma(szu). && (16)
\end{aligned}$$

Equation (14) is by two applications of the induction hypothesis. Equations (15) and (16) are by the definition of product in $\mathsf{CF}X$.

Finally, for $t = \mu x.p$,

$$\begin{aligned}
\sigma(s(\mu x.p)u) &= \sum_n \sigma(s(nx.p)u) && (17)\\
&= \sum_n \sum_{y\in\tau(nx.p)} \sigma(syu) && (18)\\
&= \sum_{y\in\bigcup_n \tau(nx.p)} \sigma(syu) && (19)\\
&= \sum_{y\in\tau(\mu x.p)} \sigma(syu). && (20)
\end{aligned}$$

Equation (17) is just the μ-continuity property (8). Equation (18) is by the induction hypothesis, observing that $\mu x.p \succ nx.p$. Equation (19) is a basic property of suprema. Finally, equation (20) is by the definition of $\tau(\mu x.p)$ in $\mathsf{CF}X$.

The result (10) for the special case of $\tau = L_X$ is immediate, observing that L_X satisfies the assumption of the lemma: for $x \in X$,

$$\sigma(sxu) = \sum_{y\in\{x\}} \sigma(syu) = \sum_{y\in L_X(x)} \sigma(syu).$$

□

At this point we can show that the μ-continuity condition implies the Park axioms.

Theorem 3.2. *The μ-continuity condition* (8) *implies the Park axioms* (5).

Proof. We first show $p \le x \Rightarrow \mu x.p \le x$ in any idempotent semiring satisfying the μ-continuity condition. Let σ be a valuation such that $\sigma(\mu x.p) = \sum_n \sigma(nx.p)$. Suppose that $\sigma(p) \le \sigma(x)$. We show by induction that for all $n \ge 0$, $\sigma(nx.p) \le \sigma(x)$. This is certainly true for $0x.p = 0$. Now suppose it is true for $nx.p$. Using monotonicity,

$$\sigma((n+1)x.p) = \sigma(p[x/nx.p]) \le \sigma(p[x/x]) = \sigma(p) \le \sigma(x).$$

By μ-continuity, $\sigma(\mu x.p) = \sum_n \sigma(nx.p) \le \sigma(x)$.

Now we show that $p[x/\mu x.p] \le \mu x.p$. This requires the stronger property that a μ-expression is chain-continuous with respect to suprema of context-free languages as a function of its free variables. Using Lemmas 2.5 and 3.1,

$$\begin{aligned}
\sigma(p[x/\mu x.p]) &= \sum \{\sigma(y) \mid y \in L_X(p[x/\mu x.p])\}\\
&= \sum \left\{\sigma(y) \mid y \in \bigcup_n L_X(p[x/nx.p])\right\}\\
&= \sum_n \sum \{\sigma(y) \mid y \in L_X(p[x/nx.p])\}\\
&= \sum_n \sigma(p[x/nx.p])\\
&= \sum_n \sigma((n+1)x.p)\\
&= \sigma(\mu x.p).
\end{aligned}$$

□

The following is our main theorem.

Theorem 3.3. *Let X be an arbitrary set and let $s,t \in \mathsf{T}X$. The following are equivalent:*

(i) *The equation $s = t$ holds in all μ-continuous Chomsky algebras; that is, $s = t$ is a logical consequence of the axioms of idempotent semirings and the μ-continuity condition*

$$a(\mu x.t)b = \sum_{n \geq 0} a(nx.t)b, \tag{21}$$

or equivalently, the universal formulas

$$a(nx.t)b \leq a(\mu x.t)b, \quad n \geq 0 \tag{22}$$

$$\left(\bigwedge_{n \geq 0} (a(nx.t)b \leq w) \right) \Rightarrow a(\mu x.t)b \leq w. \tag{23}$$

(ii) *The equation $s = t$ holds in the semiring of context-free languages $\mathsf{CF}Y$ over any set Y.*

(iii) *$L_X(s) = L_X(t)$, where $L_X : \mathsf{T}X \to \mathsf{CF}X$ is the standard interpretation mapping a μ-expression to a context-free language of strings over its free variables.*

Thus the axioms of idempotent semirings and μ-continuity are sound and complete for the equational theory of the context-free languages.

Proof. The implication (i) ⇒ (ii) holds since $\mathsf{CF}Y$ is a μ-continuous Chomsky algebra, and (iii) is a special case of (ii). Finally, if (iii) holds, then by two applications of Lemma 3.1, for any interpretation $\sigma : \mathsf{T}X \to K$ over a μ-continuous Chomsky algebra K,

$$\sigma(s) = \sum_{x \in L_K(s)} \sigma(x) = \sum_{x \in L_K(t)} \sigma(x) = \sigma(t),$$

which proves (i). □

Theorem 3.4. *The context-free languages over the alphabet X form the free μ-continuous Chomsky algebra on generators X.*

Proof. Let K be a μ-continuous Chomsky algebra. Any map $\sigma : X \to K$ extends uniquely to an interpretation $\sigma : \mathsf{T}X \to K$. By Lemma 3.1, this decomposes as

$$\sigma = \sum \circ \mathsf{CF}\,\sigma \circ L_X,$$

where $L_X : \mathsf{T}X \to \mathsf{CF}X$ is the canonical interpretation in the context-free languages over X, $\mathsf{CF}\,\sigma : \mathsf{CF}X \to \mathsf{CF}K$ is the map $\mathsf{CF}\,\sigma(A) = \{\sigma(x) \mid x \in A\}$, and $\sum : \mathsf{CF}K \to K$ takes the supremum of a context-free language over K, which is guaranteed to exist by Lemma 3.1. The unique morphism $\mathsf{CF}X \to K$ corresponding to σ is $\sum \circ \mathsf{CF}\,\sigma$. Thus CF is left adjoint to the forgetful functor from μ-continuous Chomsky algebras to Set. The maps $x \mapsto \{x\} : X \to \mathsf{CF}X$ and $\sum : \mathsf{CF}K \to K$ are the unit and counit, respectively, of the adjunction. □

4 Conclusion

We have given a natural complete infinitary axiomatization of the equational theory of the context-free languages. Leiß [13] states as an open problem:

> Are there natural equations between μ-regular expressions that are valid in all continuous models of KAF, but go beyond KAG?

We have identified such a system in this paper, thereby answering Leiß's question. He does not state axiomatization as an open problem, but observes that the set of pairs of equivalent context-free grammars is not recursively enumerable, then goes on to state:

> Since there is an effective translation between context-free grammars and μ–regular expressions . . . , the equational theory of context-free languages in terms of μ-regular expressions is not axiomatizable at all.

Nevertheless, we have given an axiomatization. How do we reconcile these two views? Leiß is apparently using "axiomatization" in the sense of "recursive axiomatization." But observe that the axiom (23) is an infinitary Horn formula. To use it as a rule of inference, one would need to establish infinitely many premises of the form $x(ny.p)z \leq w$. But this in itself is a Π^0_1-complete problem. One can show that it is Π^0_1-complete to determine whether a given context-free grammar G over a two-letter alphabet generates all strings. By coding G as a μ-expression w, the problem becomes $\mu x.(1+ax+bx) \leq w$, which by (21) is equivalent to showing that $nx.(1+ax+bx) \leq w$ for all n.

Acknowledgments

We thank Zoltán Ésik, Hans Leiß, and the anonymous referees for helpful comments. The DIKU-affiliated authors express their thanks to the Department of Computer Science at Cornell University for hosting them in the Spring 2013 and to the Danish Council for Independent Research for financial support for this work under Project 11-106278, "Kleene Meets Church (KMC): Regular Expressions and Types".

References

[1] Henk Barendregt (1984): *The Lambda Calculus: Its Syntax and Semantics*. *Studies in Logic and the Foundations of Mathematics* 103, North-Holland.

[2] Hans Bekić (1984): *Definable operations in general algebras, and the theory of automata and flowcharts*. In C.B. Jones, editor: *Programming Languages and Their Definition*, *Lecture Notes in Computer Science* 177, Springer Berlin Heidelberg, pp. 30–55, doi:10.1007/BFb0048939.

[3] Bruno Courcelle (1986): *Equivalences and Transformations of Regular Systems – Applications to Recursive Program Schemes and Grammars*. *Theoretical Computer Science* 42, pp. 1–122, doi:10.1016/0304-3975(86)90050-2.

[4] Zoltán Ésik & Werner Kuich (2007): *Modern automata theory*. Unpublished manuscript.

[5] Zoltán Ésik & Hans Leiß (2002): *Greibach Normal Form in Algebraically Complete Semirings*. In: *CSL '02: Proceedings of the 16th International Workshop and 11th Annual Conference of the EACSL on Computer Science Logic*, Springer-Verlag, London, UK, pp. 135–150, doi:10.1007/3-540-45793-3_10.

[6] Zoltán Ésik & Hans Leiß (2005): *Algebraically Complete Semirings and Greibach Normal Form*. *Annals of Pure and Applied Logic* 133, pp. 173–203, doi:10.1016/j.apal.2004.10.008.

[7] Jozef Gruska (1971): *A characterization of context-free languages*. *J. Comput. Syst. Sci.* 5(4), pp. 353–364, doi:10.1016/S0022-0000(71)80023-5.

[8] Mark Hopkins (2008): *The Algebraic Approach I: The Algebraization of the Chomsky Hierarchy*. In R. Berghammer, B. Möller & G. Struth, editors: *Proc. 10th Int. Conf. Relational Methods in Computer Science and 5th Int. Conf. Applications of Kleene Algebra (RelMiCS/AKA 2008)*, *Lecture Notes in Computer Science* 4988, Springer-Verlag, Berlin Heidelberg, pp. 155–172, doi:10.1007/978-3-540-78913-0_13.

[9] Mark Hopkins (2008): *The Algebraic Approach II: Dioids, Quantales and Monads*. In R. Berghammer, B. Möller & G. Struth, editors: *Proc. 10th Int. Conf. Relational Methods in Computer Science and 5th Int. Conf. Applications of Kleene Algebra (RelMiCS/AKA 2008)*, *Lecture Notes in Computer Science* 4988, Springer-Verlag, Berlin Heidelberg, pp. 173–190, doi:10.1007/978-3-540-78913-0_14.

[10] Dexter Kozen (1981): *On Induction vs. *-Continuity*. In: *Proc. Logics of Programs*, *Lecture Notes in Computer Science (LNCS)* 131, Springer, pp. 167–176, doi:10.1007/BFb0025769.

[11] Dexter Kozen (1983): *Results on the propositional [mu]-calculus*. *Theoretical Computer Science* 27(3), pp. 333 – 354, doi:10.1016/0304-3975(82)90125-6.

[12] Dexter Kozen (1991): *The Design and Analysis of Algorithms*. Springer-Verlag, New York, doi:10.1007/978-1-4612-4400-4.

[13] Hans Leiß (1992): *Towards Kleene Algebra with Recursion*. In: *CSL ’91: Proceedings of the 5th Workshop on Computer Science Logic*, Springer-Verlag, London, UK, pp. 242–256, doi:10.1007/BFb0023771.

[14] Glynn Winskel (1993): *The Formal Semantics of Programming Languages*. MIT Press.

The Fixed-Point Theory of Strictly Contracting Functions on Generalized Ultrametric Semilattices[*]

Eleftherios Matsikoudis
University of California, Berkeley
ematsi@eecs.berkeley.edu

Edward A. Lee
University of California, Berkeley
eal@eecs.berkeley.edu

We introduce a new class of abstract structures, which we call generalized ultrametric semilattices, and in which the meet operation of the semilattice coexists with a generalized distance function in a tightly coordinated way. We prove a constructive fixed-point theorem for strictly contracting functions on directed-complete generalized ultrametric semilattices, and introduce a corresponding induction principle. We cite examples of application in the semantics of logic programming and timed computation, where, until now, the only tool available has been the non-constructive fixed-point theorem of Priess-Crampe and Ribenboim for strictly contracting functions on spherically complete generalized ultrametric semilattices.

1 Introduction

Fixed-point semantics in computer science has almost invariably been based on the fixed-point theory of order-preserving functions on ordered sets, or that of contraction mappings on metric spaces. More recently, however, there have been instances of fixed-point problems involving strictly contracting functions on generalized ultrametric spaces, such as in the semantics of logic programming (e.g., see [6], [19]), or the study of timed systems (e.g., see [17], [11]), that are not amenable to classical methods (see [15, thm. A.2 and thm. A.4]). Until recently, the only tool available for dealing with such problems was a non-constructive fixed-point theorem of Priess-Crampe and Ribenboim (see [18]). But in [15], a constructive theorem was obtained, tailored to the general form in which these problems typically appear in computer science, also delivering an induction principle for proving properties of the constructed fixed-points. What is interesting is that the proof of that theorem involved, not just the generalized ultrametric structure of the spaces of interest, but also a natural, inherent ordering of these spaces, and more importantly, the interplay between the two, which was distilled in two simple properties of the following form:

1. if $d(x_1,x_2) \leq d(x_1,x_3)$, then $x_1 \sqcap x_3 \sqsubseteq x_1 \sqcap x_2$;
2. $d(x_1 \sqcap x_2, x_1 \sqcap x_3) \leq d(x_2,x_3)$.

As it turns out, these two simple properties imply all formal properties of the relationship between the generalized distance function and the order relation in those spaces (see [14]).

The purpose of this work is to formulate the fixed-point theory of [15] as an abstract theory that can be readily applied to different fields and problems, such as the question of meaning of logic programs or the study of feedback in timed systems. To this end, we introduce a new class of abstract structures,

[*] This work was supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF awards #0720882 (CSR-EHS: PRET), #0931843 (CPS: Large: ActionWebs), and #1035672 (CPS: Medium: Ptides)), the Naval Research Laboratory (NRL #N0013-12-1-G015), and the following companies: Bosch, National Instruments, and Toyota.

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 56–71, doi:10.4204/EPTCS.126.5

which we call *generalized ultrametric semilattices*, prove a constructive fixed-point theorem of strictly contracting functions on directed-complete generalized ultrametric semilattices, and introduce a corresponding induction principle.

2 Generalized Ultrametric Semilattices

We assume that the reader is familiar with the concept of many-sorted signature, which is, of course, a straightforward generalization of that in the one-sorted case (e.g., see [7, chap. 1.1]).

We write Σ for a two-sorted signature consisting of two sorts A and D, and the following symbols:

1. an infix function symbol $\sqcap$ of type $\mathrm{A} \times \mathrm{A} \to \mathrm{A}$;
2. an infix relation symbol $\leq$ of type $\mathrm{D} \times \mathrm{D}$;
3. a constant symbol 0 of type $1 \to \mathrm{D}$;
4. a function symbol d of type $\mathrm{A} \times \mathrm{A} \to \mathrm{D}$.

Definition 2.1. A Σ-structure is a function $\mathfrak{A}$ from the set of sorts and symbols of Σ such that $\mathfrak{A}(\mathrm{A})$ and $\mathfrak{A}(\mathrm{D})$ are non-empty sets, and the following are true:

1. $\mathfrak{A}(\sqcap)$ is a function from $\mathfrak{A}(\mathrm{A}) \times \mathfrak{A}(\mathrm{A})$ to $\mathfrak{A}(\mathrm{A})$;
2. $\mathfrak{A}(\leq)$ is a subset of $\mathfrak{A}(\mathrm{D}) \times \mathfrak{A}(\mathrm{D})$;
3. $\mathfrak{A}(0)$ is a member of $\mathfrak{A}(\mathrm{D})$;
4. $\mathfrak{A}(\mathrm{d})$ is a function from $\mathfrak{A}(\mathrm{A}) \times \mathfrak{A}(\mathrm{A})$ to $\mathfrak{A}(\mathrm{D})$.

Assume a Σ-structure $\mathfrak{A}$.

We write $|\mathfrak{A}|_{\mathrm{A}}$ for $\mathfrak{A}(\mathrm{A})$, $|\mathfrak{A}|_{\mathrm{D}}$ for $\mathfrak{A}(\mathrm{D})$, $\sqcap^{\mathfrak{A}}$ for $\mathfrak{A}(\sqcap)$, $\leq^{\mathfrak{A}}$ for $\mathfrak{A}(\leq)$, $0^{\mathfrak{A}}$ for $\mathfrak{A}(0)$, and $\mathrm{d}^{\mathfrak{A}}$ for $\mathfrak{A}(\mathrm{d})$.

We call $|\mathfrak{A}|_{\mathrm{A}}$ the *carrier* of $\mathfrak{A}$ of sort A, or the *abstract set* of $\mathfrak{A}$, and $|\mathfrak{A}|_{\mathrm{D}}$ the *carrier* of $\mathfrak{A}$ of sort D, or the *distance set* of $\mathfrak{A}$.

It is, of course, possible to define concepts of homomorphism, substructure, etc., for Σ-structures as instances of the standard concepts homomorphism, substructure, etc., for many-sorted structures, which are, of course, straightforward generalizations of those for one-sorted structures (e.g., see [7, chap. 1.2]) (see [14]).

The Σ-structures that we are interested in are those in which the function assigned to $\sqcap$ behaves as the meet operation of a semilattice, the function assigned to d as the generalized distance function of a generalized ultrametric space, and the two satisfy a couple of simple properties.

Definition 2.2. A *generalized ultrametric semilattice* is a Σ-structure $\mathfrak{A}$ such that the following are true:

1. $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqcap^{\mathfrak{A}} \rangle$ is a semilattice[1];
2. $\langle |\mathfrak{A}|_{\mathrm{D}}, \leq^{\mathfrak{A}}, 0^{\mathfrak{A}} \rangle$ is a pointed[2] ordered set;

[1] For every set S, and every binary operation $\sqcap$ on S, $\langle S, \sqcap \rangle$ is a *semilattice* if and only if for any $s_1, s_2, s_3 \in S$, the following are true:

(a) $(s_1 \sqcap s_2) \sqcap s_3 = s_1 \sqcap (s_2 \sqcap s_3)$;

(b) $s_1 \sqcap s_2 = s_2 \sqcap s_1$;

(c) $s_1 \sqcap s_1 = s_1$.

[2] An ordered set[3] is *pointed* if and only if it has a least element. We write $\langle P, \leqslant, 0 \rangle$ for a pointed ordered set $\langle P, \leqslant \rangle$ with least element 0.

[3] An *ordered set* is an ordered pair $\langle P, \leqslant \rangle$ such that P is a set, and $\leqslant$ is a reflexive, transitive, and antisymmetric binary relation on P.

3. $\langle |\mathfrak{A}|_{\mathrm{A}}, |\mathfrak{A}|_{\mathrm{D}}, \leq^{\mathfrak{A}}, 0^{\mathfrak{A}}, \mathrm{d}^{\mathfrak{A}} \rangle$ is a generalized ultrametric space[4];
4. for every $a_1, a_2, a_3 \in |\mathfrak{A}|_{\mathrm{A}}$, the following are true:
   (a) if $\mathrm{d}^{\mathfrak{A}}(a_1,a_2) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_1,a_3)$, then $(a_1 \sqcap^{\mathfrak{A}} a_3) \sqcap^{\mathfrak{A}} (a_1 \sqcap^{\mathfrak{A}} a_2) = a_1 \sqcap^{\mathfrak{A}} a_3$;
   (b) $\mathrm{d}^{\mathfrak{A}}(a_1 \sqcap^{\mathfrak{A}} a_2, a_1 \sqcap^{\mathfrak{A}} a_3) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_2,a_3)$.

Notice that, in Definition 2.2.1, a semilattice is viewed as an algebraic structure. For the most part, it will be more convenient to view a semilattice as an ordered set.[5] The two views are closely connected, and one may seamlessly switch between them (e.g., see [3, lem. 2.8]). Formally, it is simpler to work with a meet operation than with an order relation (see [14]). But informally, we will recover the order relation from the meet operation, and for every $a_1, a_2 \in |\mathfrak{A}|_{\mathrm{A}}$, write $a_1 \sqsubseteq^{\mathfrak{A}} a_2$ if and only if $a_1 \sqcap^{\mathfrak{A}} a_2 = a_1$. In particular, we may rewrite Definition 2.2.4 in the following form:

4. for every $a_1, a_2, a_3 \in |\mathfrak{A}|_{\mathrm{A}}$, the following are true:
   (a) if $\mathrm{d}^{\mathfrak{A}}(a_1,a_2) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_1,a_3)$, then $a_1 \sqcap^{\mathfrak{A}} a_3 \sqsubseteq^{\mathfrak{A}} a_1 \sqcap^{\mathfrak{A}} a_2$;
   (b) $\mathrm{d}^{\mathfrak{A}}(a_1 \sqcap^{\mathfrak{A}} a_2, a_1 \sqcap^{\mathfrak{A}} a_3) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_2,a_3)$.

Of course, all this can be done formally, but we shall not worry ourselves over the details.

For notational convenience, we will informally write $\sqsubset^{\mathfrak{A}}$ for the irreflexive part of $\sqsubseteq^{\mathfrak{A}}$, and $<^{\mathfrak{A}}$ for the irreflexive part of $\leq^{\mathfrak{A}}$.

Assume a generalized ultrametric semilattice $\mathfrak{A}$.

We say that $\mathfrak{A}$ is *directed-complete* if and only if $\langle |\mathfrak{A}|, \sqsubseteq^{\mathfrak{A}} \rangle$ is directed-complete[6].

If $\mathfrak{A}$ is directed-complete, then for every $D \subseteq |\mathfrak{A}|_{\mathrm{A}}$ that is directed in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, we write $\bigsqcup^{\mathfrak{A}} D$ for the least upper bound of D in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$.

We say that $\mathfrak{A}$ is *spherically complete* if and only if $\langle |\mathfrak{A}|_{\mathrm{A}}, |\mathfrak{A}|_{\mathrm{D}}, \leq^{\mathfrak{A}}, 0^{\mathfrak{A}}, \mathrm{d}^{\mathfrak{A}} \rangle$ is spherically complete[8].

The paradigmatic example of a generalized ultrametric semilattice is the standard generalized ultrametric semilattice $\mathfrak{S}[\langle T, \leq_T \rangle, V]$ of all linear signals from some totally ordered set $\langle T, \leq_T \rangle$ to some non-empty set V (see [14]). Indeed, the definition of generalized ultrametric semilattices was motivated by the fact that every generalized ultrametric semilattice with a totally ordered distance set is isomorphic to a standard generalized ultrametric semilattice of linear signals (see [14, thm. 2]).

An example of a non-standard generalized ultrametric semilattice of linear signals is the set of all finite and infinite sequences over some non-empty set of values, equipped with the standard prefix relation and the so-called "Baire-distance function" (e.g., see [1]).

[4] A *generalized ultrametric space* is a quintuple $\langle A, P, \leqslant, 0, d \rangle$ such that A is a set, $\langle P, \leqslant, 0 \rangle$ is a pointed ordered set, d is a function from $A \times A$ to P, and for any $a_1, a_2, a_3 \in A$ and every $p \in P$, the following are true:

(a) $d(a_1,a_2) = 0$ if and only if $a_1 = a_2$;
(b) $d(a_1,a_2) = d(a_2,a_1)$;
(c) if $d(a_1,a_2) \leqslant p$ and $d(a_2,a_3) \leqslant p$, then $d(a_1,a_3) \leqslant p$.

We refer to clause 3a as the *identity of indiscernibles*, clause 3b as *symmetry*, and clause 3c as the *generalized ultrametric inequality*.

[5] An ordered set $\langle P, \leqslant \rangle$ is a *semilattice* (also called a *meet-semilattice* or a lower semilattice) if and only if for any $p_1, p_2 \in P$, there is a greatest lower bound (also called a meet) of p_1 and p_2 in $\langle P, \leqslant \rangle$.

[6] An ordered set $\langle P, \leqslant \rangle$ is *directed-complete* if and only if every subset of P that is directed[7] in $\langle P, \leqslant \rangle$ has a least upper bound in $\langle P, \leqslant \rangle$.

[7] For every ordered set $\langle P, \leqslant \rangle$, and every $D \subseteq P$, D is *directed* in $\langle P, \leqslant \rangle$ if and only if $D \neq \emptyset$, and every finite subset of D has an upper bound in $\langle D, \leqslant_D \rangle$, where $\leqslant_D$ is the restriction of $\leqslant$ to D.

[8] A generalized ultrametric space $\langle A, P, \leqslant, 0, d \rangle$ is *spherically complete* if and only if for every non-empty chain C of balls[9] in $\langle A, P, \leqslant, 0, d \rangle$, $\bigcap C \neq \emptyset$.

[9] For every generalized ultrametric space $\langle A, P, \leqslant, 0, d \rangle$, and every $B \subseteq A$, B is a *ball* in $\langle A, P, \leqslant, 0, d \rangle$ if and only if there is $a \in A$ and $p \in P$ such that $B = \{a' \in A \mid d(a', a) \leqslant p\}$.

Example 2.3. Let V be a non-empty set.

Let $\mathfrak{A}$ be a Σ-structure such that $|\mathfrak{A}|_{\mathrm{A}}$ is the set of all finite and infinite sequences over V, $|\mathfrak{A}|_{\mathrm{D}} = \mathbb{R}_{\geq 0}$,[10] and the following are true:

1. $\sqcap^{\mathfrak{A}}$ is a binary operation on $|\mathfrak{A}|_{\mathrm{A}}$ such that for every $s_1, s_2 \in |\mathfrak{A}|_{\mathrm{A}}$, $s_1 \sqcap^{\mathfrak{A}} s_2$ is the greatest common prefix of s_1 and s_2;
2. $\leq^{\mathfrak{A}}$ is the standard order on $\mathbb{R}_{\geq 0}$;
3. $0^{\mathfrak{A}} = 0$;
4. $\mathrm{d}^{\mathfrak{A}}$ is a function from $|\mathfrak{A}|_{\mathrm{A}} \times |\mathfrak{A}|_{\mathrm{A}}$ to $|\mathfrak{A}|_{\mathrm{D}}$ such that for every $s_1, s_2 \in |\mathfrak{A}|_{\mathrm{A}}$,

$$\mathrm{d}^{\mathfrak{A}}(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 = s_2; \\ 2^{-\min\{n \mid n \in \mathbb{N} \text{ and } s_1(n) \not\simeq s_2(n)\}} & \text{otherwise.}^{11} \end{cases}$$

It is easy to verify that $\mathfrak{A}$ is a directed-complete and spherically complete generalized ultrametric semilattice.

Notice that the generalized ultrametric space associated with the generalized ultrametric semilattice $\mathfrak{A}$ of Example 2.3 is a standard ultrametric space. In such a case, we may omit the term "generalized", and speak simply of an *ultrametric semilattice*.

Another example of a non-standard ultrametric semilattice of linear signals, one that is of particular interest to the study of timed computation, is the set of all discrete-event[12] real-time signals over some non-empty set of values, equipped with the standard prefix relation and the so-called "Cantor metric" (e.g., see [10], [9]).

Example 2.4. Let V be a non-empty set.

Let $\mathfrak{A}$ be a Σ-structure such that $|\mathfrak{A}|_{\mathrm{A}}$ is the set of all discrete-event signals from $\langle \mathbb{R}, \leq_{\mathbb{R}} \rangle$ to V,[13] $|\mathfrak{A}|_{\mathrm{D}} = \mathbb{R}_{\geq 0}$, and the following are true:

1. $\sqcap^{\mathfrak{A}}$ is a binary operation on $|\mathfrak{A}|_{\mathrm{A}}$ such that for every $s_1, s_2 \in |\mathfrak{A}|_{\mathrm{A}}$, $s_1 \sqcap^{\mathfrak{A}} s_2$ is the greatest common prefix of s_1 and s_2;
2. $\leq^{\mathfrak{A}}$ is the standard order on $\mathbb{R}_{\geq 0}$;
3. $0^{\mathfrak{A}} = 0$;
4. $\mathrm{d}^{\mathfrak{A}}$ is a function from $|\mathfrak{A}|_{\mathrm{A}} \times |\mathfrak{A}|_{\mathrm{A}}$ to $|\mathfrak{A}|_{\mathrm{D}}$ such that for every $s_1, s_2 \in |\mathfrak{A}|_{\mathrm{A}}$,

$$\mathrm{d}^{\mathfrak{A}}(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 = s_2; \\ 2^{-\min\{r \mid r \in \mathbb{R} \text{ and } s_1(r) \not\simeq s_2(r)\}} & \text{otherwise.} \end{cases}$$

Notice that since the domain of every signal in $|\mathfrak{A}|_{\mathrm{A}}$ is well ordered by $\leq_{\mathbb{R}}$, for every $s_1, s_2 \in |\mathfrak{A}|_{\mathrm{A}}$, $\{r \mid r \in \mathbb{R} \text{ and } s_1(r) \not\simeq s_2(r)\}$ is also well ordered by $\leq_{\mathbb{R}}$, and thus, $\min\{r \mid r \in \mathbb{R} \text{ and } s_1(r) \not\simeq s_2(r)\}$ is well defined.

It is easy to verify that $\mathfrak{A}$ is a directed-complete and spherically complete ultrametric semilattice.

[10] We write $\mathbb{R}_{\geq 0}$ for the set of all non-negative real numbers.

[11] We write $\mathbb{N}$ for the set of all natural numbers, and $\leq_{\mathbb{N}}$ for the standard order on $\mathbb{N}$.

[12] A signal s from $\langle T, \leq_T \rangle$ to V is *discrete-event* if and only if there is an order-embedding of $\langle \mathsf{dom}\, s, \leq_{\mathsf{dom}\, s} \rangle$ into $\langle \mathbb{N}, \leq_{\mathbb{N}} \rangle$, where $\leq_{\mathsf{dom}\, s}$ is the restriction of $\leq_T$ to $\mathsf{dom}\, s$.

[13] We write $\mathbb{R}$ for the set of all real numbers, and $\leq_{\mathbb{R}}$ for the standard order on $\mathbb{R}$.

Finally, we include an example from the field of logic programming. We assume familiarity with the basic concepts of logic programming (e.g., see [12]). Our notation is based on [6].

Example 2.5. Let P be a normal logic program.

Let α be a non-empty countable ordinal, and l a function from H_P, the Herbrand base of P, to α.

Let $\mathfrak{A}$ be a Σ-structure such that $|\mathfrak{A}|_{\mathrm{A}}$ is the set of all subsets of H_P, $|\mathfrak{A}|_{\mathrm{D}} = \alpha \cup \{\alpha\}$, and the following are true:

1. $\sqcap^{\mathfrak{A}}$ is a binary operation on $|\mathfrak{A}|_{\mathrm{A}}$ such that for every $I_1, I_2 \in |\mathfrak{A}|_{\mathrm{A}}$,

$$I_1 \sqcap^{\mathfrak{A}} I_2 = \{A \mid A \in I_1, A \in I_2, \text{and for every } A' \text{ such that } l(A') \in l(A) \text{ or } l(A') = l(A), A' \in I_1 \text{ if and only if } A' \in I_2\};$$

2. $\leq^{\mathfrak{A}}$ is a binary relation on $|\mathfrak{A}|_{\mathrm{D}}$ such that for every $\beta, \gamma \in |\mathfrak{A}|_{\mathrm{D}}$,

$$\beta \leq^{\mathfrak{A}} \gamma \iff \gamma \in \beta \text{ or } \beta = \gamma.$$

3. $0^{\mathfrak{A}} = \alpha$;

4. $\mathrm{d}^{\mathfrak{A}}$ is a function from $|\mathfrak{A}|_{\mathrm{A}} \times |\mathfrak{A}|_{\mathrm{A}}$ to $|\mathfrak{A}|_{\mathrm{D}}$ such that for every $I_1, I_2 \in |\mathfrak{A}|_{\mathrm{A}}$,

$$\mathrm{d}^{\mathfrak{A}}(I_1, I_2) = \{\beta \mid \beta \in \alpha, \text{and for every } A \text{ such that } l(A') \in \beta \text{ or } l(A') = \beta, A' \in I_1 \text{ if and only if } A' \in I_2\}.$$

Let $\leq_a$ be a binary relation on α such that for every $\beta, \gamma \in \alpha$,

$$\beta \leq_a \gamma \iff \beta \in \gamma \text{ or } \beta = \gamma.$$

Clearly, $\langle \alpha, \leq_\alpha \rangle$ is an ordered set.

It is easy to verify that $\mathfrak{A}$ is a directed-complete and spherically complete generalized ultrametric semilattice.

3 Contracting and Strictly Contracting Functions

Assume a function F on $\mathfrak{A}$.

We say that F is *contracting* if and only if for every $a_1, a_2 \in |\mathfrak{A}|_{\mathrm{A}}$,

$$\mathrm{d}^{\mathfrak{A}}(F(a_1), F(a_2)) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_1, a_2).$$

In other words, a function is contracting just as long as the generalized distance between any two elements in the range of the function is smaller than or equal to that between the elements in the domain of the function that map to them. Notice that, because $\leq^{\mathfrak{A}}$ is not necessarily a total order, this is different, in general, from the generalized distance between any two elements in the domain of the function being no bigger than that between the elements in the range of the function that those map to, which is why we have opted for the term "contracting" over the term "non-expanding".

We say that F is *strictly contracting* if and only if for every $a_1, a_2 \in |\mathfrak{A}|_{\mathrm{A}}$ such that $a_1 \neq a_2$,

$$\mathrm{d}^{\mathfrak{A}}(F(a_1), F(a_2)) <^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a_1, a_2).$$

The following is immediate:

Proposition 3.1. *If F is strictly contracting, then F is contracting.*

To return to Example 2.4, the contracting and strictly contracting functions on the generalized ultrametric semilattice of all discrete-event real-time signals over V are exactly the causal and strictly causal functions respectively on such signals (see [15], [16]). And in the case of Example 2.5, if the normal logic program P is a so-called "locally hierarchical" program, then the level mapping l can be chosen so that P can be modelled as a strictly contracting function on $\mathfrak{A}$ (see [6]).

Now, contracting functions need not have fixed points (e.g., see [15, exam. 3.4]). But what about strictly contracting functions?

Proposition 3.2. *If F is strictly contracting, then F has at most one fixed point.*

Proof. Suppose that F is strictly contracting.

Suppose, toward contradiction, that a_1 and a_2 are two distinct fixed points of F. Then

$$\mathrm{d}^{\mathfrak{A}}(F(a_1), F(a_2)) = \mathrm{d}^{\mathfrak{A}}(a_1, a_2),$$

obtaining a contradiction.

Thus, F has at most one fixed point. □

Theorem 3.3. *If $\mathfrak{A}$ is spherically complete, then every strictly contracting function on $\mathfrak{A}$ has exactly one fixed point.*

Theorem 3.3 follows immediately from the fixed-point theorem of Priess-Crampe and Ribenboim for strictly contracting functions on spherically complete generalized ultrametric spaces (see [18, thm. 1]), which is sometimes, and perhaps a little too liberally, referred to as a generalization of the *Banach Fixed-Point Theorem*. The following, which follows immediately from another theorem of Priess-Crampe and Ribenboim (e.g., see *Banach's Fixed Point Theorem* in [23]), justifies the use of the stronger property of spherical completeness in place of the standard property of Cauchy-completeness used in the latter:

Theorem 3.4. *If $\langle |\mathfrak{A}|_{\mathrm{D}}, \leq^{\mathfrak{A}} \rangle$ is totally ordered, then $\mathfrak{A}$ is spherically complete if and only if every strictly contracting function on $\mathfrak{A}$ has a fixed point.*

Note that the hypothesis of $\langle \mathrm{T}, \preceq \rangle$ being totally ordered in Theorem 3.4 cannot be discarded (see [15, thm. 5.5 and exam. 5.8]).

4 Fixed-Point Theory

We now develop the rudiments of a constructive fixed-point theory for strictly contracting functions.

4.1 Existence

We start by proving another fixed-point existence result for strictly contracting functions, which is similar to Theorem 3.3, but has a different premise. The proof is more like Naundorf's proof in [17], but, as also possible in the case of the existence part of Theorem 3.3 (see [18, p. 229]), our main theorem applies to a more general type of function.

Assume a function F on $\mathfrak{A}$.

We say that F is *strictly contracting on orbits* if and only if for every $a \in |\mathfrak{A}|_{\mathrm{A}}$ such that $a \neq F(a)$,

$$\mathrm{d}^{\mathfrak{A}}(F(a), F(F(a))) <^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, F(a)).$$

In other words, F is strictly contracting on orbits just as long as the generalized distance between every two successive elements in the orbit[14] of every $a \in |\mathfrak{A}|_{\mathrm{A}}$ under F gets smaller and smaller along the orbit.

The following is immediate:

Proposition 4.1. *If F is strictly contracting, then F is strictly contracting on orbits.*

Theorem 4.2. *If $\mathfrak{A}$ is directed-complete, then every contracting function on $\mathfrak{A}$ that is strictly contracting on orbits has a fixed point.*

Before we embark on the proof of the theorem, we prove two important lemmas that will be useful throughout this section.

For every function F on $\mathfrak{A}$, and every $a \in |\mathfrak{A}|_{\mathrm{A}}$, we say that a is a *post-fixed point* of F if and only if $a \sqsubseteq^{\mathfrak{A}} F(a)$.

Lemma 4.3. *For every contracting function F on $\mathfrak{A}$, and every $a \in |\mathfrak{A}|_{\mathrm{A}}$, the following are true:*

1. *$F(a) \sqcap^{\mathfrak{A}} F(F(a))$ is a post-fixed point of F;*
2. *if a is a post-fixed point of F, then $a \sqsubseteq^{\mathfrak{A}} F(a) \sqcap^{\mathfrak{A}} F(F(a))$.*

Proof. Assume a contracting function F on $\mathfrak{A}$, and $a \in |\mathfrak{A}|_{\mathrm{A}}$.

Since F is contracting, by Definition 2.2.4b,

$$\begin{aligned} \mathrm{d}^{\mathfrak{A}}(F(F(a) \sqcap^{\mathfrak{A}} F(F(a))),F(F(a))) &\preceq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(F(a) \sqcap^{\mathfrak{A}} F(F(a)),F(a)) \\ &= \mathrm{d}^{\mathfrak{A}}(F(a) \sqcap^{\mathfrak{A}} F(F(a)),F(a) \sqcap^{\mathfrak{A}} F(a)) \\ &\preceq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(F(a),F(F(a))), \end{aligned}$$

and thus, by Definition 2.2.4a,

$$\begin{aligned} F(a) \sqcap^{\mathfrak{A}} F(F(a)) &\sqsubseteq^{\mathfrak{A}} F(F(a) \sqcap^{\mathfrak{A}} F(F(a))) \sqcap^{\mathfrak{A}} F(F(a)) \\ &\sqsubseteq^{\mathfrak{A}} F(F(a) \sqcap^{\mathfrak{A}} F(F(a))). \end{aligned}$$

Thus, 1 is true.

Suppose that $a \sqsubseteq^{\mathfrak{A}} F(a)$.

Since F is contracting,

$$\mathrm{d}^{\mathfrak{A}}(F(a),F(F(a))) \preceq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a,F(a)),$$

and thus, by Definition 2.2.4a,

$$a \sqcap^{\mathfrak{A}} F(a) \sqsubseteq^{\mathfrak{A}} F(a) \sqcap^{\mathfrak{A}} F(F(a)).$$

And since $a \sqsubseteq^{\mathfrak{A}} F(a)$, $a \sqcap^{\mathfrak{A}} F(a) = a$, and thus,

$$a \sqsubseteq^{\mathfrak{A}} F(a) \sqcap^{\mathfrak{A}} F(F(a)).$$

Thus, 2 is true. □

Lemma 4.4. *For every contracting function F on $\mathfrak{A}$, and any set P of post-fixed points of F, if P has a least upper bound in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, then $\bigsqcup^{\mathfrak{A}} P$ is a post-fixed point of F.*

[14] For every set A, every function f on A, and any $a \in A$, the *orbit* of a under f is the sequence $\langle f^n(a) \mid n \in \omega \rangle$.

Proof. Assume a contracting function F on $\mathfrak{A}$, and a set P of post-fixed points of F that has a least upper bound in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$.

Assume $a \in P$.

Since F is contracting,

$$\mathrm{d}^{\mathfrak{A}}(F(a), F(\bigsqcup^{\mathfrak{A}} P)) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, \bigsqcup^{\mathfrak{A}} P). \tag{1}$$

By Definition 2.2.4b and (1),

$$\mathrm{d}^{\mathfrak{A}}((\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(a), (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P)) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, \bigsqcup^{\mathfrak{A}} P). \tag{2}$$

Also, since a is a post-fixed point of F, by Definition 2.2.4b,

$$\begin{aligned} \mathrm{d}^{\mathfrak{A}}(a, (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(a)) &= \mathrm{d}^{\mathfrak{A}}(F(a) \sqcap^{\mathfrak{A}} a, F(a) \sqcap^{\mathfrak{A}} \bigsqcup^{\mathfrak{A}} P) \\ &\leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, \bigsqcup^{\mathfrak{A}} P). \end{aligned} \tag{3}$$

By (2), (3), and the generalized ultrametric inequality,

$$\mathrm{d}^{\mathfrak{A}}(a, (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P)) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, \bigsqcup^{\mathfrak{A}} P).$$

Then, by the generalized ultrametric inequality,

$$\mathrm{d}^{\mathfrak{A}}(\bigsqcup^{\mathfrak{A}} P, (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P)) \leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a, \bigsqcup^{\mathfrak{A}} P),$$

and thus, by Definition 2.2.4a,

$$\begin{aligned} a \sqcap^{\mathfrak{A}} \bigsqcup^{\mathfrak{A}} P &\sqsubseteq^{\mathfrak{A}} (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P) \\ &= (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P). \end{aligned}$$

However, since $a \in P$, $a \sqsubseteq^{\mathfrak{A}} \bigsqcup^{\mathfrak{A}} P$, and thus, $a \sqcap^{\mathfrak{A}} \bigsqcup^{\mathfrak{A}} P = a$. Thus,

$$\begin{aligned} a &\sqsubseteq^{\mathfrak{A}} (\bigsqcup^{\mathfrak{A}} P) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P) \\ &\sqsubseteq^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P). \end{aligned}$$

Thus, by generalization, $F(\bigsqcup^{\mathfrak{A}} P)$ is an upper bound of P in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$. And since $\bigsqcup^{\mathfrak{A}} P$ is the least upper bound of P in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, $\bigsqcup^{\mathfrak{A}} P \sqsubseteq^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} P)$. Thus, $\bigsqcup^{\mathfrak{A}} P$ is a post-fixed point of F. □

Proof of Theorem 4.2. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a contracting function F on $\mathfrak{A}$ that is strictly contracting on orbits.

Let $P = \{a \mid a \text{ is a post-fixed point of } F\}$.

Let a be a member of $|\mathfrak{A}|_{\mathrm{A}}$.

By Lemma 4.3.1,

$$F(a) \sqcap^{\mathfrak{A}} F(F(a)) \sqsubseteq^{\mathfrak{A}} F(F(a) \sqcap^{\mathfrak{A}} F(F(a))),$$

and thus, $P \neq \emptyset$. Then, by Kuratowski's Lemma (see [3, sec. 10.2]), every chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$ is contained in a $\subset$-maximal chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$.

Let C be a $\subset$-maximal chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$.

Since $\mathfrak{A}$ is directed-complete, C has a least upper bound in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$.

We claim that $\bigsqcup^{\mathfrak{A}} C$ is a fixed point of F.

Suppose, toward contradiction, that $\bigsqcup^{\mathfrak{A}} C$ is not a fixed point of F.

Let $x = F(\bigsqcup^{\mathfrak{A}} C) \sqcap^{\mathfrak{A}} F(F(\bigsqcup^{\mathfrak{A}} C))$.

By Lemma 4.4, $\bigsqcup^{\mathfrak{A}} C \sqsubseteq^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} C)$, and thus, by Lemma 4.3.2, $\bigsqcup^{\mathfrak{A}} C \sqsubseteq^{\mathfrak{A}} x$.

Suppose, toward contradiction, that $\bigsqcup^{\mathfrak{A}} C = x$. Since F is strictly contracting on orbits, and $\bigsqcup^{\mathfrak{A}} C$ is not a fixed point of F,

$$\mathrm{d}^{\mathfrak{A}}(F(\bigsqcup^{\mathfrak{A}} C), F(F(\bigsqcup^{\mathfrak{A}} C))) <^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(\bigsqcup^{\mathfrak{A}} C, F(\bigsqcup^{\mathfrak{A}} C)). \tag{4}$$

However, since $x = F(\bigsqcup^{\mathfrak{A}} C) \sqcap^{\mathfrak{A}} F(F(\bigsqcup^{\mathfrak{A}} C))$ and $\bigsqcup^{\mathfrak{A}} C = x$, by Definition 2.2.4b,

$$\begin{aligned}
\mathrm{d}^{\mathfrak{A}}(\bigsqcup^{\mathfrak{A}} C, F(\bigsqcup^{\mathfrak{A}} C)) &= \mathrm{d}^{\mathfrak{A}}(F(\bigsqcup^{\mathfrak{A}} C), \bigsqcup^{\mathfrak{A}} C) \\
&= \mathrm{d}^{\mathfrak{A}}(F(\bigsqcup^{\mathfrak{A}} C), F(\bigsqcup^{\mathfrak{A}} C) \sqcap^{\mathfrak{A}} F(F(\bigsqcup^{\mathfrak{A}} C))) \\
&= \mathrm{d}^{\mathfrak{A}}(F(\bigsqcup^{\mathfrak{A}} C) \sqcap^{\mathfrak{A}} F(\bigsqcup^{\mathfrak{A}} C), F(\bigsqcup^{\mathfrak{A}} C) \sqcap^{\mathfrak{A}} F(F(\bigsqcup^{\mathfrak{A}} C))) \\
&\leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(F(\bigsqcup^{\mathfrak{A}} C), F(F(\bigsqcup^{\mathfrak{A}} C))),
\end{aligned}$$

contrary to (4).

Therefore, $\bigsqcup^{\mathfrak{A}} C \sqsubset^{\mathfrak{A}} x$. Thus, $x \notin C$. And by Lemma 4.3.1, $x \sqsubseteq^{\mathfrak{A}} F(x)$, and thus, $x \in P$. Thus, $C \cup \{x\}$ is a chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$, and $C \subset C \cup \{x\}$, contrary to C being a $\subset$-maximal chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$.

Therefore, $\bigsqcup^{\mathfrak{A}} C$ is a fixed point of F. □

There are two things to notice here. First, the proof of Theorem 4.2 is inherently non-constructive, overtly appealing to the Axiom of Choice through the use of Kuratowski's Lemma. And second, there need not be only one fixed point; indeed, the identity function on $\mathfrak{A}$ is trivially causal and strictly contracting on orbits, yet every element is a fixed point of it.

The following is immediate from Proposition 3.1, 3.2, and 4.1, and Theorem 4.2:

Theorem 4.5. *If $\mathfrak{A}$ is directed-complete, then every strictly contracting function on $\mathfrak{A}$ has exactly one fixed point.*

If $\mathfrak{A}$ is directed-complete, then for every strictly contracting function F on $\mathfrak{A}$, we write $\mathsf{fix} F$ for the unique fixed point of F.

The following is immediate from Theorem 3.4 and 4.5:

Corollary 4.6. *If $\langle |\mathfrak{A}|_{\mathrm{D}}, \leq^{\mathfrak{A}} \rangle$ is totally ordered, then if $\mathfrak{A}$ is directed-complete, then $\mathfrak{A}$ is spherically complete.*

We note that the hypothesis of $\langle \mathrm{T}, \preceq \rangle$ being totally ordered in Corollary 4.6 cannot be discarded (see [15, exam. 5.8]). As a consequence, Theorem 3.3 and 4.5 are incomparable with respect to deduction; that is, one cannot deduce Theorem 4.5 from Theorem 3.3, nor Theorem 3.3 from Theorem 4.5.

4.2 Construction

Although theoretically pleasing, mere existence of fixed points is practically moot. Theorem 4.2 and 4.5, just like Theorem 3.3, offer little if no means of deductive reasoning about the fixed points ascertained to exist.

But how are we to construct these fixed points? Theorem A.2 and A.4 in [15] seem to render standard fixed-point theories of ordered sets and metric spaces more or less irrelevant. At the same time, it may well be that the relevant fixed-point theorem of Priess-Crampe and Ribenboim is independent of the theory of generalized ultrametric spaces in the classical Zermelo-Fraenkel set theory without choice, thus lacking a constructive proof altogether.[15]

The answer lies in the non-constructive proof of Theorem 4.2. Indeed, the proof contains all the ingredients of a transfinite recursion facilitating the construction of a chain that may effectively substitute for the maximal one only asserted to exist therein by an appeal to Kuratowski's Lemma. We may start with any arbitrary post-fixed point of the function F, and iterate through the function $\lambda a : |\mathfrak{A}|_{\mathrm{A}} \,.\, F(a) \sqcap^{\mathfrak{A}} F(F(a))$ to form an ascending chain of such points. Every so often, we may take the supremum of all post-fixed points theretofore constructed, and resume the process therefrom, until no further progress can be made. Of course, the phrase "every so often" is to be interpreted rather liberally here, and certain groundwork is required before we can formalize its transfinite intent.

We henceforth assume some familiarity with transfinite set theory, and in particular, ordinal numbers. The unversed reader may refer to any introductory textbook on set theory for details (e.g., see [4]).

We write $\mathbf{1m2}^{\mathfrak{A}}F$ for a function on $\mathfrak{A}$, such that for any $a \in |\mathfrak{A}|_{\mathrm{A}}$,

$$(\mathbf{1m2}^{\mathfrak{A}}F)(a) = F(a) \sqcap^{\mathfrak{A}} F(F(a)).$$

In other words, $\mathbf{1m2}^{\mathfrak{A}}F$ is the function $\lambda a : |\mathfrak{A}|_{\mathrm{A}} \,.\, F(a) \sqcap^{\mathfrak{A}} F(F(a))$.

Assume a post-fixed point a of F.

We let

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{0}(a) = a,$$

for every ordinal α,

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha+1}(a) = (\mathbf{1m2}^{\mathfrak{A}}F)((\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a)),$$

and for every limit ordinal λ,

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\lambda}(a) = \bigsqcup^{\mathfrak{A}}\{(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a) \mid \alpha \in \lambda\}.$$

The following implies that for every ordinal α, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a)$ is well defined:

Lemma 4.7. *If $\mathfrak{A}$ is directed-complete, then for every contracting function F on $\mathfrak{A}$, any post-fixed point a of F, and every ordinal α,*

1. $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a) \sqsubseteq^{\mathfrak{A}} F((\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a))$;
2. *for any $\beta \in \alpha$, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) \sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a)$.*

Proof. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a contracting function F on $\mathfrak{A}$, a post-fixed point a of F, and an ordinal α.

We use transfinite induction on the ordinal α to jointly prove that 1 and 2 are true.

If $\alpha = 0$, then $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a) = a$. Thus, 1 is trivially true, whereas 2 is vacuously true.

Suppose that there is an ordinal β such that $\alpha = \beta + 1$.

Then

$$\begin{aligned}(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a) &= (\mathbf{1m2}^{\mathfrak{A}}F)((\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a)) \\ &= F((\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a)) \sqcap^{\mathfrak{A}} F(F((\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a))). \end{aligned} \tag{5}$$

[15] A purportedly constructive proof for the fixed-point theorem of Priess-Crampe and Ribenboim under the hypothesis of a totally ordered set of distances was presented in [5, thm. 1.3.9]. However, the proof covertly appeals to the Axiom of Choice through a potentially transfinite sequence of choices.

Thus, by Lemma 4.3.1, 1 is true.

For every $\gamma \in \alpha$, either $\gamma = \beta$, or $\gamma \in \beta$, and thus, by the induction hypothesis,

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\gamma}(a) \sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a). \tag{6}$$

Also, by the induction hypothesis,

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \sqsubseteq^{\mathfrak{A}} F((\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a)).$$

Thus, by Lemma 4.3.2 and (5),

$$\begin{aligned}(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) &\sqsubseteq^{\mathfrak{A}} F((\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a)) \sqcap^{\mathfrak{A}} F(F((\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a))) \\ &= (\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a). \end{aligned} \tag{7}$$

And by (6) and (7), $(\mathbf{1m2}^{\mathfrak{A}} F)^{\gamma}(a) \sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a)$. Thus, 2 is true.

Otherwise, α is a limit ordinal. By the induction hypothesis, $\langle \{(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \mid \beta \in \alpha\}, \sqsubseteq^{\mathfrak{A}} \rangle$ is totally ordered, and thus, $\{(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \mid \beta \in \alpha\}$ is directed in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$. And since $\mathfrak{A}$ is directed-complete, $\{(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \mid \beta \in \alpha\}$ has a least upper bound in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, and

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a) = \bigsqcup^{\mathfrak{A}} \{(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \mid \beta \in \alpha\}.$$

Thus, 2 is trivially true.

By the induction hypothesis, for every $\beta \in \alpha$, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \sqsubseteq^{\mathfrak{A}} F((\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a))$. Thus, by Lemma 4.4, 1 is true. □

By Lemma 4.7.2, and a simple cardinality argument, there is an ordinal α such that for every ordinal β such that $\alpha \in \beta$, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a)$. In fact, there is a least ordinal α such that for every contracting function F on $\mathfrak{A}$, any post-fixed point a of F, and every ordinal β such that $\alpha \in \beta$, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a)$.

We write $\mathsf{oh}\,\mathfrak{A}$ for the least ordinal α such that there is no function φ from α to $|\mathfrak{A}|_{\mathrm{A}}$ such that for every $\beta, \gamma \in \alpha$, if $\beta \in \gamma$, then $\varphi(\beta) \sqsubset^{\mathfrak{A}} \varphi(\gamma)$.

In other words, $\mathsf{oh}\,\mathfrak{A}$ is the least ordinal that cannot be orderly embedded in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, which we may think of as the *ordinal height* of $\mathfrak{A}$. Notice that the Hartogs number of $|\mathfrak{A}|_{\mathrm{A}}$ is an ordinal that cannot be orderly embedded in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}} \rangle$, and thus, $\mathsf{oh}\,\mathfrak{A}$ is well defined, and in particular, smaller than or equal to the Hartogs number of $|\mathfrak{A}|_{\mathrm{A}}$.

Lemma 4.8. *If $\mathfrak{A}$ is directed-complete, then for every contracting function F on $\mathfrak{A}$, any post-fixed point a of F, and every ordinal α, if $(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a)$ is not a fixed point of $\mathbf{1m2}^{\mathfrak{A}} F$, then $\alpha + 2 \in \mathsf{oh}\,\mathfrak{A}$.*

Proof. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a contracting function F on $\mathfrak{A}$, a post-fixed point a of F, and an ordinal α.

Suppose that $(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}(a)$ is not a fixed point of $\mathbf{1m2}^{\mathfrak{A}} F$.

We claim that for any $\beta, \gamma \in \alpha + 2$, if $\beta \neq \gamma$, then

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}(a) \neq (\mathbf{1m2}^{\mathfrak{A}} F)^{\gamma}(a).$$

Suppose, toward contradiction, that there are $\beta,\gamma \in \alpha+2$ such that $\beta \neq \gamma$, but

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}}F)^{\gamma}(a).$$

Without loss of generality, assume that $\beta \in \gamma$. Since F is contracting, by Lemma 4.7.2,

$$\begin{aligned}(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) &\sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}}F)^{\beta+1}(a)\\ &\sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}}F)^{\gamma}(a),\end{aligned}$$

and thus,

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}}F)^{\beta+1}(a).$$

And since $\beta \in \gamma \in \alpha+2$, either $\beta \in \alpha$, or $\beta = \alpha$. Thus, by an easy transfinite induction,

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a),$$

contrary to the assumption that $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}(a)$ is not a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$.

Therefore, for any $\beta,\gamma \in \alpha+2$,

$$(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}(a) = (\mathbf{1m2}^{\mathfrak{A}}F)^{\gamma}(a)$$

if and only if $\beta = \gamma$. Thus, since F is contracting, by Lemma 4.7.2, there is a function φ from $\alpha+2$ to $|\mathfrak{A}|_{\mathrm{A}}$ such that for every $\beta,\gamma \in \alpha+2$, if $\beta \in \gamma$, then $\varphi(\beta) \sqsubset^{\mathfrak{A}} \varphi(\gamma)$. Thus, by definition of $\mathsf{oh}\,\mathfrak{A}$, $\alpha+2 \in \mathsf{oh}\,\mathfrak{A}$. □

By Lemma 4.8, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ is a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$. Nevertheless, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ need not be a fixed point of F as intended. Indeed, the recursion process might start stuttering at points that are not fixed under the function in question (e.g., see [15, exam.3.4]). If the function is strictly contracting on orbits, however, progress at such points is guaranteed.

Lemma 4.9. *For every function F on $\mathfrak{A}$ that is strictly contracting on orbits, a is a fixed point of F if and only if a is a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$.*

Proof. Assume a function F on $\mathfrak{A}$ that is strictly contracting on orbits.

If a is a fixed point of F, then

$$\begin{aligned}a &= F(a)\\ &= F(F(a)),\end{aligned}$$

and thus,

$$\begin{aligned}a &= F(a) \sqcap^{\mathfrak{A}} F(F(a))\\ &= (\mathbf{1m2}^{\mathfrak{A}}F)(a).\end{aligned}$$

Conversely, suppose that a is a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$.

Then, by Definition 2.2.4b,

$$\begin{aligned}\mathrm{d}^{\mathfrak{A}}(a,F(a)) &= \mathrm{d}^{\mathfrak{A}}((\mathbf{1m2}^{\mathfrak{A}}F)(a),F(a))\\ &= \mathrm{d}^{\mathfrak{A}}(F(a) \sqcap^{\mathfrak{A}} F(F(a)),F(a))\\ &= \mathrm{d}^{\mathfrak{A}}(F(a) \sqcap^{\mathfrak{A}} F(F(a)),F(a) \sqcap^{\mathfrak{A}} F(a))\\ &\leq^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(F(a),F(F(a))).\end{aligned} \tag{8}$$

Suppose, toward contradiction, that a is not a fixed point of F. Then, since F is strictly contracting on orbits,

$$\mathrm{d}^{\mathfrak{A}}(F(a),F(F(a))) <^{\mathfrak{A}} \mathrm{d}^{\mathfrak{A}}(a,F(a)),$$

contrary to (8).

Therefore, a is a fixed point of F. □

We may at last put all the different pieces together to obtain a constructive version of Theorem 4.2.

Theorem 4.10. *If $\mathfrak{A}$ is directed-complete, then for every contracting function F on $\mathfrak{A}$ that is strictly contracting on orbits, and any post-fixed point a of F, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ is a fixed point of F.*

Proof. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a contracting function F on $\mathfrak{A}$ that is strictly contracting on orbits, and a post-fixed point a of F.

Suppose, toward contradiction, that $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ is not a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$. Then, by Lemma 4.8, $\mathsf{oh}\,\mathfrak{A}+2 \in \mathsf{oh}\,\mathfrak{A}$, a contradiction.

Therefore, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ is a fixed point of $\mathbf{1m2}^{\mathfrak{A}}F$. And since F is strictly contracting on orbits, by Lemma 4.9, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$ is a fixed point of F. □

To be pedantic, Theorem 4.10 does not directly prove that F has a fixed point; unless there is a post-fixed point of F, the theorem is true vacuously. But by Lemma 4.3.1, for every $a \in |\mathfrak{A}|_{\mathrm{A}}$, $(\mathbf{1m2}^{\mathfrak{A}}F)(a)$ is a post-fixed point of F.

The following is immediate from Proposition 3.1 and 4.1, Lemma 4.3.1, and Theorem 4.10:

Theorem 4.11. *If $\mathfrak{A}$ is directed-complete, then for every strictly contracting function F on $\mathfrak{A}$, and every $a \in |\mathfrak{A}|_{\mathrm{A}}$,*

$$\mathsf{fix}F = (\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}((\mathbf{1m2}^{\mathfrak{A}}F)(a)).$$

This construction of fixed points as "limits of stationary transfinite iteration sequences" is very similar to the construction of extremal fixed points of monotone operators in [2] and references therein, where the function iterated is not $\mathbf{1m2}^{\mathfrak{A}}F$, but F itself. Notice, however, that if F preserves $\sqsubseteq^{\mathfrak{A}}$, then for any post-fixed point a of F, $(\mathbf{1m2}^{\mathfrak{A}}F)(a) = F(a)$.

The astute reader will at this point anticipate the following:

Theorem 4.12. *If $\mathfrak{A}$ is directed-complete, then for every strictly contracting function F on $\mathfrak{A}$,*

$$\mathsf{fix}F = \bigsqcup^{\mathfrak{A}}\{a \mid a \text{ is a post-fixed point of } F\}.$$

Proof. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a strictly contracting function F on $\mathfrak{A}$.

Assume a post-fixed point a of F.

By Lemma 4.7.2, $a \sqsubseteq^{\mathfrak{A}} (\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}(a)$, and thus, since F is strictly contracting, by Proposition 3.1 and 4.1, Lemma 4.7.2, and Theorem 4.10, $a \sqsubseteq^{\mathfrak{A}} \mathsf{fix}F$.

Thus, by generalization, $\mathsf{fix}F$ is an upper bound of $\{a \mid a \text{ is a post-fixed point of } F\}$ in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}}\rangle$. And since $\mathsf{fix}F$ is a post-fixed point of F, for every upper bound u of $\{a \mid a \text{ is a post-fixed point of } F\}$ in $\langle |\mathfrak{A}|_{\mathrm{A}}, \sqsubseteq^{\mathfrak{A}}\rangle$, $\mathsf{fix}F \sqsubseteq^{\mathfrak{A}} u$. Thus,

$$\mathsf{fix}F = \bigsqcup^{\mathfrak{A}}\{a \mid a \text{ is a post-fixed point of } F\}.$$ □

In retrospect, we find that Theorem 4.12 may be derived directly from first principles. In particular, and under the premise of the corollary, it is easy to establish without any use of Theorem 4.10 that for every $a \in |\mathfrak{A}|_{\mathrm{A}}$, $a \sqsubseteq^{\mathfrak{A}} \mathsf{fix} F$ if and only if $a \sqsubseteq^{\mathfrak{A}} F(a)$, as the reader may wish to verify.

The construction of Theorem 4.12 is identical in form to Tarski's well known construction of greatest fixed points of order-preserving functions on complete lattices (see [25, thm. 1]).

Finally, we note that $\mathbf{1m2}^{\mathfrak{A}} F$ is not, in general, order-preserving under the above premises (see [15, exam. 2.15]), as might be suspected, and thus, our fixed-point theorem is not a reduction to a standard order-theoretic one.

In view of Example 2.4 and 2.5, and the comments in the paragraph following Proposition 3.1, Theorem 4.11 and 4.12 can be directly applied to study the behaviour of strictly causal discrete-event components in feedback (see [15], [16]), and obtain, constructively, the unique supported model of locally hierarchical normal logic programs (see [6]).

4.3 Induction

Having used transfinite recursion to construct fixed points, we may use transfinite induction to prove properties of them. And in the case of strictly contracting endofunctions, which have exactly one fixed point, we may use Theorem 4.11 to establish a special proof rule.

Assume $P \subseteq |\mathfrak{A}|_{\mathrm{A}}$.

We say that P is *strictly inductive* if and only if every non-empty chain in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$ has a least upper bound in $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$.

Note that P is strictly inductive if and only if $\langle P, \sqsubseteq^{\mathfrak{A}} \rangle$ is directed-complete (see [13, cor. 2]).

Theorem 4.13. *If $\mathfrak{A}$ is directed-complete, then for every strictly contracting function F on $\mathfrak{A}$, and every non-empty, strictly inductive $P \subseteq |\mathfrak{A}|_{\mathrm{A}}$, if for every $a \in P$, $(\mathbf{1m2}^{\mathfrak{A}} F)(a) \in P$, then* $\mathsf{fix} F \in P$.

Proof. Suppose that $\mathfrak{A}$ is directed-complete.

Assume a strictly contracting function F on $\mathfrak{A}$, and non-empty, strictly inductive $P \subseteq |\mathfrak{A}|_{\mathrm{A}}$.

Suppose that for every $a \in P$, $(\mathbf{1m2}^{\mathfrak{A}} F)(a) \in P$.

Let a be a member of P.

By Lemma 4.3.1, $(\mathbf{1m2}^{\mathfrak{A}} F)(a)$ is a post-fixed point of F.

We use transfinite induction to prove that for every ordinal α, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \in P$.

If $\alpha = 0$, then

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) = (\mathbf{1m2}^{\mathfrak{A}} F)(a),$$

and thus, since P is closed under $\mathbf{1m2}^{\mathfrak{A}} F$, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \in P$.

If there is an ordinal β such that $\alpha = \beta + 1$, then

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) = (\mathbf{1m2}^{\mathfrak{A}} F)((\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}((\mathbf{1m2}^{\mathfrak{A}} F)(a))).$$

By the induction hypothesis, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \in P$, and thus, since P is closed under $\mathbf{1m2}^{\mathfrak{A}} F$, $(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \in P$.

Otherwise, α is a limit ordinal, and thus,

$$(\mathbf{1m2}^{\mathfrak{A}} F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) = \bigsqcup^{\mathfrak{A}} \{ (\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \mid \beta \in \alpha \}.$$

By the induction hypothesis,

$$\{ (\mathbf{1m2}^{\mathfrak{A}} F)^{\beta}((\mathbf{1m2}^{\mathfrak{A}} F)(a)) \mid \beta \in \alpha \} \subseteq P,$$

and by Lemma 4.7.2, $\langle\{(\mathbf{1m2}^{\mathfrak{A}}F)^{\beta}((\mathbf{1m2}^{\mathfrak{A}}F)(a)) \mid \beta \in \alpha\}, \sqsubseteq^{\mathfrak{A}}\rangle$ is totally ordered. Thus, since P is strictly inductive, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}}F)(a)) \in P$.

Therefore, by transfinite induction, for every ordinal α, $(\mathbf{1m2}^{\mathfrak{A}}F)^{\alpha}((\mathbf{1m2}^{\mathfrak{A}}F)(a)) \in P$.

By Theorem 4.11,

$$\mathsf{fix}\, F = (\mathbf{1m2}^{\mathfrak{A}}F)^{\mathsf{oh}\,\mathfrak{A}}((\mathbf{1m2}^{\mathfrak{A}}F)(a)),$$

and thus, $\mathsf{fix}\, F \in P$. □

Theorem 4.13 is an induction principle that one may use to prove properties of fixed points of strictly contracting endofunctions. We think of properties extensionally here; that is, a property is a subset of $|\mathfrak{A}|_{\mathrm{A}}$. And the properties that are admissible for use with this principle are those that are non-empty and strictly inductive. According to the principle, then, for every strictly contracting function F on any directed-complete generalized ultrametric semilattice $\mathfrak{A}$, every non-empty, strictly inductive property that is preserved by $\mathbf{1m2}^{\mathfrak{A}}F$ is true of $\mathsf{fix}\, F$.

We refer to [15, sec. 5.3] for a comparison between this principle with the fixed-point induction principle for order-preserving functions on complete partial orders (see [24]), and the fixed-point induction principle for contraction mappings on complete metric spaces (see [20], [22], [21], [8]).

References

[1] Jaco W. de Bakker & Erik P. de Vink (1998): *Denotational models for programming languages: applications of Banach's Fixed Point Theorem*. *Topology and its Applications* 85(1-3), pp. 35–52, doi:10.1016/S0166-8641(97)00140-5.

[2] Patrick Cousot & Radhia Cousot (1979): *Constructive versions of Tarski's fixed point theorems*. *Pacific J. of Math.* 82(1), pp. 43–57, doi:10.2140/pjm.1979.82.43.

[3] Brian A. Davey & Hilary A. Priestley (2002): *Introduction to Lattices and Order*, second edition. Cambridge University Press, doi:10.1017/CBO9780511809088.

[4] Herbert B. Enderton (1977): *Elements of Set Theory*. Academic Press.

[5] Pascal Hitzler (2001): *Generalized Metrics and Topology in Logic Programming Semantics*. Ph.D. thesis, Department of Mathematics, National University of Ireland, University College Cork.

[6] Pascal Hitzler & Anthony Karel Seda (2003): *Generalized metrics and uniquely determined logic programs*. *Theoretical Computer Science* 305(1-3), pp. 187–219, doi:10.1016/S0304-3975(02)00709-0.

[7] Wilfrid Hodges (1993): *Model Theory*. *Encyclopedia of Mathematics and its Applications* 42, Cambridge University Press, doi:10.1017/CBO9780511551574.

[8] Dexter Kozen & Nicholas Ruozzi (2007): *Applications of Metric Coinduction*. In: *CALCO'07: Proceedings of the 2nd international conference on Algebra and coalgebra in computer science*, Springer-Verlag, Berlin, Heidelberg, pp. 327–341, doi:10.1007/978-3-540-73859-6_22.

[9] Edward A. Lee (1999): *Modeling concurrent real-time processes using discrete events*. *Annals of Software Engineering* 7(1), pp. 25–45, doi:10.1023/A:1018998524196.

[10] Edward A. Lee & Alberto Sangiovanni-Vincentelli (1998): *A Framework for Comparing Models of Computation*. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 17(12), pp. 1217–1229, doi:10.1109/43.736561.

[11] Xiaojun Liu, Eleftherios Matsikoudis & Edward A. Lee (2006): *Modeling Timed Concurrent Systems*. In Christel Baier & Holger Hermanns, editors: *CONCUR 2006 – Concurrency Theory*, *Lecture Notes in Computer Science* 4137, Springer Berlin / Heidelberg, pp. 1–15, doi:10.1007/11817949_1.

[12] John W. Lloyd (1987): *Foundations of Logic Programming*, second, extended edition. Springer-Verlag, doi:10.1007/978-3-642-83189-8.

[13] George Markowsky (1976): *Chain-complete posets and directed sets with applications*. Algebra Universalis 6(1), pp. 53–68, doi:10.1007/BF02485815.

[14] Eleftherios Matsikoudis & Edward A. Lee (2013): *An Axiomatization of the Theory of Generalized Ultrametric Semilattices of Linear Signals*. In Leszek Gąsieniec & Frank Wolter, editors: Fundamentals of Computation Theory, Lecture Notes in Computer Science 8070, Springer Berlin Heidelberg, pp. 248–258, doi:10.1007/978-3-642-40164-0_24.

[15] Eleftherios Matsikoudis & Edward A. Lee (2013): *The Fixed-Point Theory of Strictly Causal Functions*. Technical Report UCB/EECS-2013-122, EECS Department, University of California, Berkeley.

[16] Eleftherios Matsikoudis & Edward A. Lee (2013): *On Fixed Points of Strictly Causal Functions*. In Víctor Braberman & Laurent Fribourg, editors: Formal Modeling and Analysis of Timed Systems, Lecture Notes in Computer Science 8053, Springer Berlin Heidelberg, pp. 183–197, doi:10.1007/978-3-642-40229-6_13.

[17] Holger Naundorf (2000): *Strictly causal functions have a unique fixed point*. Theoretical Computer Science 238(1-2), pp. 483–488, doi:10.1016/S0304-3975(99)00165-6.

[18] Sibylla Priess-Crampe & Paulo Ribenboim (1993): *Fixed Points, Combs and Generalized Power Series*. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 63(1), pp. 227–244, doi:10.1007/BF02941344.

[19] Sibylla Priess-Crampe & Paulo Ribenboim (2000): *Ultrametric spaces and logic programming*. The Journal of Logic Programming 42(2), pp. 59–70, doi:10.1016/S0743-1066(99)00002-3.

[20] George M. Reed & A. William Roscoe (1986): *A Timed Model for Communicating Sequential Processes*. In Laurent Kott, editor: Automata, Languages and Programming, Lecture Notes in Computer Science 226, Springer Berlin / Heidelberg, pp. 314–323, doi:10.1007/3-540-16761-7_81.

[21] A. William Roscoe (1991): *Topology, computer science, and the mathematics of convergence*. In G. M. Reed, A. W. Roscoe & R. F. Wachter, editors: Topology and category theory in computer science, chapter 1, Oxford University Press, Inc., New York, NY, USA, pp. 1–27.

[22] William C. Rounds (1985): *Applications of topology to semantics of communicating processes*. In Stephen Brookes, Andrew Roscoe & Glynn Winskel, editors: Seminar on Concurrency, Lecture Notes in Computer Science 197, Springer Berlin / Heidelberg, pp. 360–372, doi:10.1007/3-540-15670-4_17.

[23] Erwin Schörner (2003): *Ultrametric Fixed Point Theorems and Applications*. In: Valuation Theory and its Applications, Fields Institute Communications II, American Mathematical Society, pp. 353–359.

[24] Dana S. Scott & Jaco W. de Bakker (1969): *A theory of programs*. Unpublished notes, Seminar on Programming, IBM Research Center, Vienna, Austria.

[25] Alfred Tarski (1955): *A Lattice-Theoretical Fixpoint Theorem and its Applications*. Pacific J. of Math. 5(2), pp. 285–309, doi:10.2140/pjm.1955.5.285.

Guard Your Daggers and Traces: On The Equational Properties of Guarded (Co-)recursion

Stefan Milius Tadeusz Litak
Chair for Theoretical Computer Science (Informatik 8)
Friedrich-Alexander University Erlangen-Nürnberg, Germany
mail@stefan-milius.eu tadeusz.litak@gmail.com

Motivated by the recent interest in models of guarded (co-)recursion we study its equational properties. We formulate axioms for guarded fixpoint operators generalizing the axioms of iteration theories of Bloom and Ésik. Models of these axioms include both standard (e.g., cpo-based) models of iteration theories and models of guarded recursion such as complete metric spaces or the topos of trees studied by Birkedal et al. We show that the standard result on the satisfaction of all Conway axioms by a unique dagger operation generalizes to the guarded setting. We also introduce the notion of guarded trace operator on a category, and we prove that guarded trace and guarded fixpoint operators are in one-to-one correspondence. Our results are intended as first steps leading to the description of classifying theories for guarded recursion and hence completeness results involving our axioms of guarded fixpoint operators in future work.

1 Introduction

Our ability to describe concisely potentially infinite computations or infinite behaviour of systems relies on recursion, corecursion and iteration. Most programming languages and specification formalisms include a fixpoint operator. In order to give semantics to such operators one usually considers either

- models based on complete partial orders where fixpoint operators are interpreted by least fixpoints using the Kleene-Knaster-Tarski theorem or
- models based on complete metric spaces and unique fixpoints via Banach's theorem or
- term models where unique fixpoints arise by unfolding specifications syntactically.

In the last of these cases, one only considers *guarded* (co-)recursive definitions; see e.g. Milner's solution theorem for CCS [21] or Elgot's iterative theories [13]. Thus, the fixpoint operator becomes a partial operator defined only on a special class of maps. For a concrete example consider complete metric spaces which form a category with all non-expansive maps as morphisms, but unique fixpoints are taken only of contractive maps.

Recently, there has been a wave of interest in expressing guardedness by a new type constructor $\rhd$, a kind of "later" modality, which allows to make the fixpoint operator total, see, e.g., Nakano [23, 24], Appel et al. [4], Benton and Tabareau [7], Krishnaswami and Benton [19, 18], Birkedal et al. [9, 8] and Atkey and McBride [5]. For example, in the case of complete metric spaces $\rhd$ can be an endofunctor scaling the metric of any given space by a fixed factor $0 < r < 1$ so that non-expansive maps of type $\rhd X \to X$ are precisely contractive maps with a contraction factor of at most r. This allows to define a guarded (parametrized) fixpoint operator on *all* morphisms of type $\rhd X \times Y \to X$ of the model. So far various models allowing the interpretation of a typed language including a guarded fixpoint operator

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 72–86, doi:10.4204/EPTCS.126.6

have been studied: complete metric spaces, the "topos of trees", i.e., presheaves on ω^{op} [9] or, more generally, sheaves on complete Heyting algebras with a well-founded basis [12, 9].

This paper initiates the study of the essential properties of guarded fixpoint operators. In the realm of ordinary fixpoint operators, it is well-known that iteration theories of Bloom and Ésik [10] completely axiomatize equalities of fixpoint terms in models based on complete partial orders (see also Simpson and Plotkin [25]). We make here the first steps towards similar completeness results in the guarded setting.

We begin with formalizing the notion of guarded fixpoint operator on a cartesian category. We discuss a number of models, including not only all those mentioned above, but also some not mentioned so far in the context of $\triangleright$-guarded (co-)recursion. In fact, we consider the inclusion of examples such as the lifting functor on CPO (which also happens to be a paradigm example of a *fixpoint monad*, see Example 2.4.6 and the concluding remark of Section 2.7) or completely iterative monads (see Section 2.2) a pleasant by-product of our work and a potentially fruitful connection for future research. Then, we formulate generalizations of standard iteration theory axioms for guarded fixpoint operators and we establish these axioms are sound in all models under consideration. In particular, the central result of Section 2 is Theorem 2.16: models with *unique* guarded fixpoint operators satisfy all our axioms.

Hasegawa [16] proved that giving a parametrized fixpoint operator on a category satisfying the so-called *Conway axioms* (see, e.g., [10, 25] and Section 2.3 below) is equivalent to giving a *traced cartesian structure* [17] on that category.[1] Section 3 lifts this result to the guarded setting. We introduce a natural notion of a guarded trace operator on a category, and we prove in Theorem 3.5 that guarded traces and guarded fixpoint operators are in one-to-one correspondence. This extends to an isomorphism between the (2-)categories of guarded traced cartesian categories and guarded Conway categories.

Section 4 concludes and discusses further work.

Proofs of the major theorems will be made available in the full version.

1.1 Notational conventions

We will assume throughout that readers are familiar with basic notions from category theory. We denote the product of two objects by

$$A \xleftarrow{\pi_\ell} A \times B \xrightarrow{\pi_r} B,$$

and $\Delta : A \to A \times A$ denotes the diagonal. For every functor F we write $\mathsf{can} = \langle F\pi_\ell, F\pi_r \rangle : F(A \times B) \to FA \times FB$ for the canonical morphism.

We denote by CPO the category of complete partial orders (cpo's), i.e. partially ordered sets (not necessarily with a least element) having joins of ω-chains. The morphisms of CPO are Scott-continuous maps, i.e. maps preserving joins of ω-chains. And $\mathsf{CPO}_\perp$ is the full subcategory of CPO given by all cpo's with a least element $\perp$. We will also consider the category CMS of complete 1-bounded metric spaces and non-expansive maps.

2 Guarded Fixpoint Operators

In this section we define the notion of a guarded fixpoint operator on a cartesian category and present an extensive list of examples. Some of these examples like the lifting functor $(-)_\perp$ on CPO (see Example 2.4.6) or completely iterative monads (see Section 2.2) do not seem to have been considered as instances of the guarded setting before. We then introduce (equational) properties of guarded fixpoint operators.

[1] *Cartesian* here refers to the monoidal product being the ordinary categorical product.

These properties are motivated by and closely resemble properties of the fixpoint operator in iteration theories of Bloom and Ésik [10]. We conclude this section with Theorem 2.16 stating that unique fixpoint operators satisfy all the properties we study.

2.1 Definition and Examples of Guarded Fixpoint Operators

Assumption 2.1. We assume throughout the rest of the paper that $(\mathscr{C}, \triangleright)$ is a pair consisting of a category $\mathscr{C}$ with finite products (also know as a *cartesian category*) and a pointed endofunctor $\triangleright : \mathscr{C} \to \mathscr{C}$, i.e. we have a natural transformation $p : \mathsf{Id} \to \triangleright$. The endofunctor $\triangleright$ is called *delay*.

Remark 2.2. In references like [9, 8], much more is assumed about both the underlying category and the delay endofunctor. Whenever one wants to model simply-typed lambda calculus, one obviously imposes the condition of being cartesian closed. Furthermore, whenever one considers dependent types, one wants to postulate conditions like being a *type-theoretic fibration category* (see, e.g., [8, Definition IV.1]). In such a case, one also wants to impose some limit-preservation or at least finite-limit-preservation condition on the delay endofunctor, see [9, Definition 6.1]—e.g., to ensure the transfer of the guarded fixpoint operator to slices. We do not impose any of those restrictions because we do not need them in this paper. It is an interesting fact that all our derivations require no more than Assumption 2.1. For more on the connection with the setting of [9], see Proposition 2.6 below.

Definition 2.3. A *guarded fixpoint operator* on $(\mathscr{C}, \triangleright)$ is a family of operations

$$\dagger_{X,Y} : \mathscr{C}(\triangleright X \times Y, X) \to \mathscr{C}(Y, X)$$

such that for every $f : \triangleright X \times Y \to X$ the following square commutes[2]:

$$\begin{tikzcd}
Y \arrow[r, "f^\dagger"] \arrow[d, "\langle f^\dagger, Y \rangle"'] & X \\
X \times Y \arrow[r, "p_X \times Y"'] & \triangleright X \times Y \arrow[u, "f"']
\end{tikzcd} \tag{2.1}$$

where (as usual) we drop the subscripts and write $f^\dagger : Y \to X$ in lieu of $\dagger_{X,Y}(f)$. We call the triple $(\mathscr{C}, \triangleright, \dagger)$ a *guarded fixpoint category*.

Usually, one either assumes that $\dagger$ satisfies further properties or even that $f^\dagger$ is unique such that (2.1) commutes. We will come to the study of properties of guarded fixpoint operators in Section 2.3. Let us begin with a list of examples.

Examples 2.4. (1) Taking as $\triangleright$ the identity functor on $\mathscr{C}$ and p_X the identity on X we arrive at the special case of categories with an ordinary fixpoint operator $\mathscr{C}(X \times Y, X) \to \mathscr{C}(Y, X)$ (see e.g. Hasegawa [16, 15] or Simpson and Plotkin [25]). Concrete examples are: the category $\mathsf{CPO}_\perp$ with its usual least fixpoint operator or (the dual of) any iteration theory of Bloom and Ésik [10].

(2) Taking $\triangleright$ to be the constant functor on the terminal object 1 and $p_X = ! : X \to 1$ the unique morphism, a trivial guarded fixpoint operator is given by the family of identity maps on the hom-sets $\mathscr{C}(Y, X)$.

(3) Take $\mathscr{C}$ to be the category CMS of complete 1-bounded metric spaces (see [19, 18] or [9, Section 5] and references therein), $\triangleright_r$ $(0 < r < 1)$ to be an endofunctor which keeps the carrier of the space and multiplies all distances by r and $p_X : X \to \triangleright_r X$ to be the obvious "contracted identity" mapping.

[2] Notice that we use the convention of simply writing objects to denote the identity morphisms on them.

Note that a non-expansive mapping $f : \rhd_r X \to X$ is the same as an *r-contractive* endomap, i.e. an endomap satisfying $d(fx, fy) \leq r \cdot d(x,y)$. A guarded fixpoint operator is given by an application of Banach's unique fixpoint theorem: for every $f : \rhd_r X \times Y \to X$ we consider the map

$$\Phi_f : \mathsf{CMS}(Y,X) \to \mathsf{CMS}(Y,X), \qquad \Phi_f(m) = f \cdot (p_X \times Y) \cdot \langle m, Y \rangle;$$

notice that $\mathsf{CMS}(Y,X)$ is a complete metric space with the sup-metric $d_{Y,X}(m,n) = \sup_{y \in Y}\{d_X(my, ny)\}$; it is then easy to show that Φ_f is an r-contractive map, and so its unique fixpoint is a unique non-expansive map $f^\dagger : Y \to X$ such that (2.1) commutes.

(4) Let $\mathscr{A}$ be a category with finite products, and let $\mathscr{C}$ be the presheaf category $\mathsf{presh}(\omega, \mathscr{A}) := \mathscr{A}^{\omega^{\mathsf{op}}}$ of ω^{op}-chains in $\mathscr{A}$. The delay functor $\rhd$ takes a presheaf $X : \omega^{\mathsf{op}} \to \mathscr{A}$ to the presheaf $\rhd X$ with $\rhd X(0) = 1$ and $\rhd X(n+1) = X(n)$ for $n \geq 0$. And p_X is given by $(p_X)_0 : X(0) \to 1$ unique and $(p_X)_{n+1} = X(n+1 \geq n) : X(n+1) \to X(n)$. For every $f : \rhd X \times Y \to X$ there is a unique $f^\dagger : Y \to X$ making (2.1) commutative; it is defined as follows: given $f : \rhd X \times Y \to X$ (i.e. $f_0 : Y(0) \to X(0)$ and $f_{n+1} : X(n) \times Y(n+1) \to X(n+1)$) one defines $f^\dagger : Y \to X$ by $f^\dagger_0 = f_0 : Y(0) \to X(0)$ and

$$f^\dagger_{n+1} = \big(Y(n+1) \xrightarrow{\langle f^\dagger_n \cdot Y(n+1 \geq n), Y(n+1)\rangle} X(n) \times Y(n+1) \xrightarrow{f_{n+1}} X(n+1)\big).$$

It is not difficult to prove that $f^\dagger$ is the unique morphism such that (2.1) commutes.

Notice that for $\mathscr{A} = \mathsf{Set}$, $\mathscr{C}$ is the "topos of trees" studied by Birkedal et al. [9]; they prove in Theorem 2.4 that $\mathsf{Set}^{\omega^{\mathsf{op}}}$ has a unique guarded fixpoint operator.

The next example generalizes this one.

(5) Assume $\mathfrak{W} := (W, <)$ is a well-founded poset, i.e, contains no infinite descending chains; for simplicity, we can assume $\mathfrak{W}$ has a root r. Furthermore, let $\mathscr{D}$ be a (small) complete category and $\mathscr{C} := \mathsf{presh}(\mathfrak{W}, \mathscr{D})$, i.e., $\mathscr{C} = \mathscr{D}^{(W,>)}$. Define $(\rhd X)(w)$ to be the limit of the diagram whose nodes are $X(u)$ for $u < w$ and whose arrows are restriction morphisms: $\rhd X(w) = \lim_{v<w} X(v)$. Then as $X(w)$ itself with restriction mappings forms a cone on that diagram, a natural $p_X : X \to \rhd X$ is given by the universal property of the limits. Note that for r, we have that $(\rhd X)(r)$ is the terminal object 1 of $\mathscr{D}$. The $\dagger$-operation is defined as follows: given $f : \rhd X \times Y \to X$ one defines $f^\dagger : Y \to X$ by induction on $(W, <)$; for the root r let $f^\dagger_r = f_r : Y(r) = 1 \times Y(r) \to X(r)$, and assuming that $f^\dagger_v$ is already defined for all $v < w$ let

$$f^\dagger_w = \big(Y(w) \xrightarrow{\langle k, Y(w)\rangle} \rhd X(w) \times Y(w) \xrightarrow{f_w} X(w)\big),$$

where $k : Y(w) \to \rhd X(w)$ is the morphism uniquely induced by the cone $f^\dagger_v \cdot Y(w > v) : Y(w) \to Y(v) \to X(v)$ for every $v < w$. One can prove that $f^\dagger$ is a morphism of presheaves and that it is the unique one such that (2.1) commutes. Details will be given in the full version. Regarding the examples given in [9], see also Proposition 2.6 below.

(6) Let $\rhd$ be the lifting functor $(-)_\perp$ on CPO, i.e. for any cpo X, $X_\perp$ is the cpo with a newly added least element. The natural transformation $p_X : X \to X_\perp$ is the embedding of X into $X_\perp$. Then CPO has a guarded fixpoint operator given by taking least fixpoints. To see this notice that the hom-sets $\mathsf{CPO}(X,Y)$ are cpos with the pointwise order: $f \leq g$ iff $f(x) \leq g(x)$ for all $x \in X$. Now any continuous $f : X_\perp \times Y \to X$ gives rise to a continuous map Φ_f on $\mathsf{CPO}(Y, X_\perp)$:

$$\Phi_f : \mathsf{CPO}(Y, X_\perp) \to \mathsf{CPO}(Y, X_\perp), \qquad \Phi_f(m) = p_X \cdot f \cdot \langle m, Y \rangle.$$

Using the least fixpoint s of Φ_f one then defines:

$$f^\dagger = (Y \xrightarrow{\langle s,Y\rangle} X_\perp \times Y \xrightarrow{f} X);$$

using that $s = \Phi_f(s)$ it is not difficult to prove that $f^\dagger$ makes (2.1) commutative.

Birkedal et al. [9] provide a general setting for topos-theoretic examples like (4) and (5) (the latter restricted to the case of Set-presheaves) by defining a notion of *a model of guarded recursive terms* and showing that *sheaves over complete Heyting algebras with a well-founded basis* proposed by [12] are instances of this notion. The difference between Definition 6.1 in [9] and our Definition 2.3 is that in the former a) the delay endofunctor $\rhd$ is also assumed to preserve finite limits. On other hand b) our equality (2.1) is only postulated in the case when Y is the terminal object, i.e., only non-parametrized fixpoint identity is assumed but c) the dagger in this less general version of (2.1) is assumed to be unique. Now, one can show that assumptions a) and c) imply our parametrized identity (2.1) *whenever the underlying category is cartesian closed*, in particular whenever $\mathscr{C}$ is a topos. Let us state both the definition and the result formally:

Definition 2.5 ([9]). A *model of guarded fixpoint terms* is a triple $(\mathscr{C}, \rhd, \ddagger)$, where

- $(\mathscr{C}, \rhd)$ satisfy our general Assumption 2.1, i.e., $\rhd : \mathscr{C} \to \mathscr{C}$ is a pointed endofunctor (with point $p : \mathsf{Id} \to \rhd$) and $\mathscr{C}$ has finite limits
- $\rhd$ preserves finite limits and
- $\ddagger$ is a family of operations $\ddagger_X : \mathscr{C}(\rhd X, X) \to \mathscr{C}(1, X)$ such that for every $f : \rhd X \to X$, $f^\ddagger$ is a unique morphism making the following square commute:

$$\begin{tikzcd} 1 \arrow[r, "f^\ddagger"] \arrow[d, "f^\ddagger"'] & X \\ X \arrow[r, "p_X"'] & \rhd X \arrow[u, "f"'] \end{tikzcd} \tag{2.2}$$

We write $\mathsf{can}^{-1}_{X,Y} : \rhd X \times \rhd Y \to \rhd(X \times Y)$ for the isomorphism provided by the assumption of limit preservation for the special case of product[3] of X and Y.

Proposition 2.6. *If $(\mathscr{C}, \rhd, \ddagger)$ is a model of guarded recursive terms and $\mathscr{C}$ is* cartesian closed *with*

$$\begin{aligned} &\mathsf{curry}^X_{Y,Z} &&: \mathscr{C}(X \times Y, Z) &&\to \mathscr{C}(X, Z^Y), \\ &\mathsf{uncurry}^X_{Y,Z} &&: \mathscr{C}(X, Z^Y) &&\to \mathscr{C}(X \times Y, Z), \\ &\mathsf{eval}_{Y,Z} &&: Y \times Z^Y &&\to Z, \end{aligned}$$

then the operator $\dagger_{X,Y} : \mathscr{C}(\rhd X \times Y, X) \to \mathscr{C}(Y, X)$ defined as

$$\mathsf{uncurry}^1_{Y,X}([\mathsf{curry}^{\rhd(X^Y)}_{Y,X}(f \cdot \langle (\rhd \mathsf{eval}_{Y,X}) \cdot \mathsf{can}^{-1}_{Y,X^Y} \cdot (p_Y \times \rhd(X^Y)), \pi_\ell \rangle)]^\ddagger)$$

is a guarded fixpoint operator on $(\mathscr{C}, \rhd)$.

[3] One can note here that for the purpose of stating and proving Proposition 2.6, the assumption of finite limit preservation in Definition 2.5 can be weakened to finite product preservation. We only keep the stronger assumption for full consistency with [9, Definition 6.1].

Obviously, we implicitly identified Y and $1 \times Y$ above. Note that the converse implication does not hold. Example 2.4.6 is a a guarded fixpoint category, but $(-)_\perp$ clearly fails to preserve even finite products and hence it does not yield a model of guarded recursive terms.

Also, while we do not have a counterexample at the moment, Proposition 2.6 is not likely to hold when the assumption that $\mathscr{C}$ is cartesian closed is removed: we believe there are examples of models of guarded recursive terms which are not guarded fixpoint categories. However, to apply Proposition 2.6, it is enough that $(\mathscr{C}, \rhd, \ddagger)$ is a *full subcategory* of a cartesian closed model of guarded recursive terms such that, moreover, the inclusion functor preserves products and $\rhd$.

Remark 2.7. Monads provide perhaps the most natural and well-known examples of pointed endofunctors. The reader may ask whether delay endofunctors in Example 2.4 happen to be monads. Clearly, the delay functors in (1), (2) and (6) are. In fact, while the first two ones are rather trivial monads, 6 is a paradigm example of a *fixpoint monad* of Crole and Pitts [11]. In (3), i.e. the CMS example, the type $\rhd\rhd A \to \rhd A$ is still inhabited (by any constant mapping), but one can easily show that monad laws cannot hold whatever candidate for monad multiplication is postulated. In the remaining (i.e., topos-theoretic) examples, monad laws fail more dramatically: $\rhd\rhd A \to \rhd A$ is not even always inhabited. The following section discusses perhaps the most interesting subclass of monads which happen to be delay endofunctors with unique dagger.

2.2 Completely Iterative Theories

In this subsection we will explain how categories with guarded fixpoint operator capture a classical setting in which guarded recursive definitions are studied—Elgot's (completely) iterative theories [13, 14]. The connection to guarded fixpoint operators is most easily seen if we consider monads in lieu of Lawvere theories, and so we follow the presentation of *(completely) iterative monads* in [20]. The motivating example for completely iterative monads are infinite trees on a signature, and we recall this now. Let Σ be a signature, i.e. a sequence $(\Sigma_n)_{n<\omega}$ of sets of operation symbols with prescribed arity n. A Σ-tree t on a set X of generators is a rooted and ordered (finite or infinite) tree whose nodes with $n > 0$ children are labelled by n-ary operation symbols from Σ and a leaf is labelled by a constant symbol from Σ_0 or by a generator from X. One considers systems of mutually recursive equations of the form

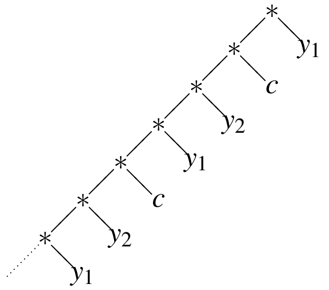
$$x_i \approx t_i(\vec{x}, \vec{y}) \qquad i \in I,$$

where $X = \{x_i \mid i \in I\}$ is a set of recursion variables and each t_i is a Σ-tree on $X + Y$ with Y a set of parameters (i.e. generators that do not occur on the left-hand side of a recursive equation). A system of recursive equations is *guarded* if none of the trees t_i is only a recursion variable $x \in X$. Every guarded system has a unique *solution*, which assigns to every recursion variable $x_i \in X$ a Σ-tree $t_i^\dagger(\vec{y})$ on Y such that $t_i^\dagger(\vec{y}) = t_i[\vec{t}^\dagger(\vec{y})/\vec{x}]$, i.e. t_i with each x_j replaced by $t_j^\dagger(\vec{y})$. For a concrete example, let Σ consist of a binary operation symbol $*$ and a constant symbol c, i.e. $\Sigma_0 = \{c\}$, $\Sigma_2 = \{*\}$ and $\Sigma_n = \emptyset$ else. Then the following system

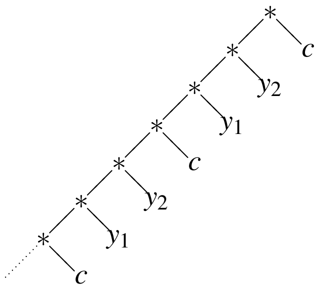
$$x_1 \approx x_2 * y_1 \qquad x_2 \approx (x_1 * y_2) * c,$$

where y_1 and y_2 are parameters, has the following unique solution:

$t_1^\dagger =$



and $t_2^\dagger =$



For any set X, let $T_\Sigma(X)$ be the set of Σ-trees on X. It has been realized by Badouel [6] that T_Σ is the object part of a monad. A system of equations is then nothing but a map

$$f : X \to T_\Sigma(X+Y)$$

and a solution is a map $f^\dagger : X \to T_\Sigma Y$ such that the following square commutes:

$$\begin{tikzcd}
X \arrow[r, "f^\dagger"] \arrow[d, "f"'] & T_\Sigma Y \\
T_\Sigma(X+Y) \arrow[r, "{[f^\dagger, \eta_Y]}"'] & T_\Sigma T_\Sigma Y \arrow[u, "\mu_Y"']
\end{tikzcd}$$

where η and μ are the unit and multiplication of the monad T_Σ, respectively.

It is clear that the notion of equation and solution can be formulated for every monad S. However, the notion of guardedness requires one to speak about *non-variables* in S. This is enabled by Elgot's notion of *ideal theory* [13], which for a finitary monad on Set is equivalent to the notion recalled in the following definition. We assume for the rest of this subsection that $\mathscr{A}$ is a category with finite coproducts such that coproduct injections are monomorphic.

Definition 2.8 ([2]). By an *ideal monad* on $\mathscr{A}$ is understood a six-tuple

$$(S, \eta, \mu, S', \sigma, \mu')$$

consisting of a monad (S, η, μ) on $\mathscr{A}$, a subfunctor $\sigma : S' \hookrightarrow S$ and a natural transformation $\mu' : S'S \to S'$ such that

(1) $S = S' + \mathsf{Id}$ with coproduct injections σ and η, and

(2) μ restricts to μ' along σ, i.e., the square below commutes:

$$\begin{tikzcd}
S'S \arrow[r, "\mu'"] \arrow[d, "\sigma S"'] & S' \arrow[d, "\sigma"] \\
SS \arrow[r, "\mu"'] & S
\end{tikzcd}$$

The subfunctor S' of an ideal monad S allows us to formulate the notion of a guarded equation system abstractly; this leads to the notion of completely iterative theory of Elgot et al. [14] for which we here present the formulation with monads from [20]:

Definition 2.9. Let $(S,\eta,\mu,S',\sigma,\mu')$ be an ideal monad on $\mathscr{A}$.

1. By an *equation morphism* is meant a morphism

$$f : X \to S(X+Y)$$

in $\mathscr{A}$, where X is an object ("of variables") and Y is an object ("of parameters").

2. By a *solution* of f is meant a morphism $f^\dagger : X \to SY$ for which the following square commutes:

$$\begin{tikzcd}
X \arrow[r, "f^\dagger"] \arrow[d, "f"'] & SY \\
S(X+Y) \arrow[r, "{S[f^\dagger,\eta_Y]}"'] & SSY \arrow[u, "\mu_Y"']
\end{tikzcd} \tag{2.3}$$

3. The equation morphism f is called *guarded* if it factors through the summand $S'(X+Y)+Y$ of $S(X+Y) = S'(X+Y)+X+Y$:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, dotted] & S(X+Y) \\
& S'(X+Y)+Y \arrow[u, "{[\sigma_{X+Y},\eta_{X+Y}\cdot \mathsf{inr}]}"']
\end{tikzcd}$$

4. The given ideal monad is called *completely iterative* if every guarded equation morphism has a unique solution.

Examples 2.10. We only briefly mention two examples of completely iterative monads. More can be found in [2, 20, 3].

(1) The monad T_Σ of Σ-trees is a completely iterative monad.

(2) A more general example is given by parametrized final coalgebras. Let $H : \mathscr{A} \to \mathscr{A}$ be an endofunctor such that for every object X of $\mathscr{A}$ a final coalgebra TX for $H(-)+X$ exists. Then T is the object assignment of a completely iterative monad; in fact, T is the free completely iterative monad on H (see [20]).

We will now explain how completely iterative monads are subsumed by the notion of categories with a guarded fixpoint operator. To this end we fix a completely iterative monad S. We will show that the dual of its Kleisli category $\mathscr{C} = (\mathscr{A}_S)^{op}$ is equipped with a guarded fixpoint operator. First notice, that since $\mathscr{A}_S$ has coproducts given by the coproducts in $\mathscr{A}$ we see that $\mathscr{C}$ has products. Next we need to obtain the endofunctor $\rhd$ on $\mathscr{C}$. This will be given as the dual of an extension of the subfunctor $S' : \mathscr{A} \to \mathscr{A}$ of S to the Kleisli category $\mathscr{A}_S$. Indeed, it is well-known that to have an extension of S' to $\mathscr{A}_S$ is equivalent to having a distributive law of the functor S' over the monad S (see Mulry [22]).

But it is easy to verify that the natural transformation

$$S'S \xrightarrow{\mu'} S' \xrightarrow{\eta S'} SS'$$

satisfies the two required laws and thus yields a distributive law. Moreover, the ensuing endofunctor $\rhd^{op} = S'$ on $\mathscr{A}_S$ is copointed, i.e. we have a natural transformation p from S' to $\mathsf{Id} : \mathscr{A}_S \to \mathscr{A}_S$; indeed, its components at X are given by the coproduct injections $\sigma_X : S'X \to SX$, and it is not difficult to verify that this is a natural transformation; thus, $\rhd$ is a pointed endofunctor on $\mathscr{C}$.

Now observe that a morphism $f : \rhd X \times Y \to X$ is equivalently a morphism

$$f : X \to S(S'X+Y)$$

in $\mathscr{A}$. We are ready to describe the guarded fixpoint operator on $\mathscr{C}$.

Construction 2.11. For any morphism $f : X \to S(S'X+Y)$ form the following morphism

$$\overline{f} = (X \xrightarrow{f} S(S'X+Y) \xrightarrow{S(\sigma_X+\eta_Y)} S(SX+SY) \xrightarrow{S\mathsf{can}} SS(X+Y) \xrightarrow{\mu_{X+Y}} S(X+Y)),$$

where $\mathsf{can} = [S\mathsf{inl}, S\mathsf{inr}] : SX+SY \to S(X+Y)$. It is not difficult to verify that $\overline{f}$ is a guarded equation morphism for S, and we define $f^\dagger : X \to SY$ to be the unique solution of $\overline{f}$.

Proposition 2.12. *For every f, $f^\dagger$ from Construction 2.11 is a unique morphism $Y \to X$ in $\mathscr{C}$ such that (2.1) commutes.*

In fact, to prove this proposition one shows that solutions of $\overline{f} : X \to S(X+Y)$ (i.e. morphisms $s : X \to SY$ such that (2.3) commutes) are in one-to-one correspondence with morphisms $Y \to X$ is $\mathscr{C}$ such that (2.1) commutes.

2.3 Properties of Guarded Fixpoint Operators

In this section we study properties of guarded fixpoint operators. Except for uniformity these properties are purely equational. They are generalizing analogous properties of iteration theories; more precisely, they would collapse to the original, unguarded counterparts when $\rhd$ is instantiated to the identity endofunctor (see Example 2.4(1)).

Definition 2.13. Let $(\mathscr{C}, \rhd, \dagger)$ be a guarded fixpoint category. We define the following properties of $\dagger$:

(1) Fixpoint Identity. For every $f : \rhd X \times Y \to X$ the diagram (2.1) commutes. This is built into the definition of guarded fixpoint categories and only mentioned here again for the sake of completeness.

(2) Parameter Identity. For every $f : \rhd X \times Y \to X$ and every $h : Z \to Y$ we have

$$Z \xrightarrow{h} Y \xrightarrow{f^\dagger} X = (\rhd X \times Z \xrightarrow{\rhd X \times h} \rhd X \times Y \xrightarrow{f} X)^\dagger.$$

(3) (Simplified) Composition Identity. Given $f : \rhd X \times Y \to Z$ and $g : Z \to X$ we have

$$(\rhd X \times Y \xrightarrow{f} Z \xrightarrow{g} X)^\dagger = (Y \xrightarrow{(f \cdot (\rhd g \times Y))^\dagger} Z \xrightarrow{g} X).$$

(4) Double Dagger Identity. For every $f : \rhd X \times \rhd X \times Y \to X$ we have

$$(Y \xrightarrow{f^{\dagger\dagger}} X) = (\rhd X \times Y \xrightarrow{\Delta \times Y} \rhd X \times \rhd X \times Y \xrightarrow{f} X)^\dagger.$$

(5) Uniformity. Given $f : \rhd X \times Y \to X$, $g : \rhd X' \times Y \to X'$ and $h : X \to X'$ we have

$$\begin{tikzcd} \rhd X \times Y \arrow[r, "f"] \arrow[dd, "\rhd h \times Y"'] & X \arrow[dd, "h"] & & & X \arrow[dd, "h"] \\ & & \Longrightarrow & Y \arrow[ur, "f^\dagger"] \arrow[dr, "g^\dagger"'] & \\ \rhd X' \times Y \arrow[r, "g"'] & X' & & & X' \end{tikzcd}$$

We call the first four properties (1)–(4) the *Conway* axioms.

Notice that the Conway axioms are equational properties while (5) is quasiequational (i.e. an implication between equations).

Next we shall show that in the presence of certain of the above properties the natural transformation $p : \mathsf{Id} \to \rhd$ is a derived structure. Let $(\mathscr{C}, \rhd)$ be equipped with an operator $\dagger$ *not necessarily satisfying (2.1)*. For every object X of $\mathscr{C}$ define $q_X : X \to \rhd X$ as follows: consider

$$f_X = (\,\rhd(\rhd X \times X) \times X \xrightarrow{\rhd\pi_r \times X} \rhd X \times X\,)$$

and form

$$q_X = (\,X \xrightarrow{f_X^\dagger} \rhd X \times X \xrightarrow{\pi_\ell} \rhd X\,).$$

Lemma 2.14. *Let $(\mathscr{C}, \rhd)$ be equipped with the operator $\dagger$. Then:*

1. *If $\dagger$ satisfies the parameter identity and uniformity, then $q : \mathsf{Id} \to \rhd$ is a natural transformation.*
2. *If $\dagger$ satisfies the fixpoint identity, then $q_X = p_X$ for all X.*

Definition 2.15. A guarded fixpoint category $(\mathscr{C}, \rhd, \dagger)$ satisfying the Conway axioms (i.e. fixpoint, parameter, composition and double dagger identities) is called a *guarded Conway category*.

If in addition uniformity is satisfied, we call $(\mathscr{C}, \rhd, \dagger)$ a *uniform guarded Conway category*.

And $(\mathscr{C}, \rhd, \dagger)$ is called a *unique guarded fixpoint category* if for every $f : \rhd X \times Y \to X$, $f^\dagger : Y \to X$ is the unique morphism such that (2.1) commutes. In this case, we can just write a pair $(\mathscr{C}, \rhd)$ rather than a triple $(\mathscr{C}, \rhd, \dagger)$.

The next theorem states that such a unique $\dagger$ satisfies all the properties in Definition 2.13.

Theorem 2.16. *If $(\mathscr{C}, \rhd)$ is a unique guarded fixpoint category, then it is a uniform guarded Conway category.*

Examples 2.17. (1) Several of our examples in 2.4 are unique guarded fixpoint categories and hence their unique $\dagger$ satisfies all the properties in Definition 2.13. This holds for Examples 2.4(2)–(6), and also for the example of completely iterative monads in Section 2.2.

(2) One can prove that Example 2.4(7), i.e., $\mathscr{C} = \mathsf{CPO}$ with the lifting functor $\rhd = (-)_\perp$ satisfies all the properties of Definition 2.13, i.e. $(\mathsf{CPO}, (-)_\perp)$ is a uniform guarded Conway category. But it is not a unique guarded fixpoint category: for let $X = \{0,1\}$ be the two-chain, $Y = 1$ the one element cpo and $f : X_\perp = X_\perp \times Y \to X$ be the map with $f(0) = f(\perp) = 0$ and $f(1) = 1$. Then both $0 : 1 \to X$ and $1 : 1 \to X$ make (2.1) commutative.

3 Guarded Trace Operators

In the case special case where $\rhd$ is the identity functor (see Example 2.4(1)), it is well-known that a fixpoint operator satisfying the Conway axioms is equivalent to a trace operator w.r.t. the product on $\mathscr{C}$ (see Hasegawa [16, 15]). In this section we present a similar result for a generalized notion of a guarded trace operator on $(\mathscr{C}, \rhd)$.

Remark 3.1. Recall that the notion of an (ordinary) trace operator was introduced by Joyal, Street and Verity [17] for symmetric monoidal categories. The applicability of the notion of trace to non-cartesian tensor products is in fact one of main reasons of its popularity. Our generalization can also be formulated for symmetric monoidal categories, see the remark preceding Construction 3.4 below. However, the main results in this section, i.e., Theorems 3.5 and 3.7 do not make any use of this added generality. Hence, we keep the Assumption 2.1 like in the remainder of the paper.

Definition 3.2. A (cartesian) *guarded trace operator* on $(\mathscr{C}, \rhd)$ is a natural family of operations

$$\mathsf{Tr}^X_{A,B} : \mathscr{C}(\rhd X \times A, X \times B) \to \mathscr{C}(A, B)$$

subject to the following three conditions:

1. **Vanishing.** (I) For every $f : \rhd 1 \times A \to B$ we have

$$\mathsf{Tr}^1_{A,B}(f) = (A \cong 1 \times A \xrightarrow{p_1 \times A} \rhd 1 \times A \xrightarrow{f} B).$$

   (II) For every $f : \rhd X \times \rhd Y \times A \to X \times Y \times B$ we have

$$\mathsf{Tr}^Y_{A,B}(\mathsf{Tr}^X_{\rhd Y \times A, Y \times A}(f)) = \mathsf{Tr}^{X \times Y}_{A,B}(\rhd(X \times Y) \times A \xrightarrow{\mathsf{can} \times A} \rhd X \times \rhd Y \times A \xrightarrow{f} X \times Y \times A).$$

2. **Superposing.** For every $f : \rhd X \times A \to X \times B$ we have

$$\mathsf{Tr}^X_{A \times C, B \times C}(f \times C) = \mathsf{Tr}^X_{A,B}(f) \times C.$$

3. **Yanking.** Consider the canonical isomorphism $c : \rhd X \times X \to X \times \rhd X$. Then we have

$$\mathsf{Tr}^X_{X, \rhd X}(c) = (X \xrightarrow{p_X} \rhd X).$$

If Tr is a (cartesian) guarded trace operator on $(\mathscr{C}, \rhd)$, $(\mathscr{C}, \rhd, \mathsf{Tr})$ is called a *guarded traced (cartesian) category*.

Of course, when $\rhd$ is taken to be the identity on $\mathscr{C}$ (as in Example 2.4(1)), our notion of guarded trace specializes to the notion of an ordinary trace operator (w.r.t. product) of Joyal, Street and Verity.

In addition, as in the case of ordinary trace operators naturality of Tr can equivalently be expressed by three more axioms:

4. **Left-tightening.** Given $f : \rhd X \times A \to X \times B$ and $g : A' \to A$ we have

$$\mathsf{Tr}^X_{A',B}(\rhd X \times A' \xrightarrow{\rhd X \times g} \rhd X \times A \xrightarrow{f} X \times B) = (A' \xrightarrow{g} A \xrightarrow{\mathsf{Tr}^X_{A,B}(f)} B).$$

5. **Right-tightening.** Given $f : \rhd X \times A \to X \times B$ and $g : B \to B'$ we have

$$\mathsf{Tr}^X_{A,B'}(\rhd X \times A \xrightarrow{f} X \times B \xrightarrow{X \times g} X \times B') = (A \xrightarrow{\mathsf{Tr}^X_{A,B}(f)} B \xrightarrow{g} B').$$

6. **Sliding.** Given $f : \rhd X \times A \to X' \times B$ and $g : X' \to X$ we have

$$\mathsf{Tr}^X_{A,B}(\rhd X \times A \xrightarrow{f} X' \times B \xrightarrow{g \times B} X \times B) = \mathsf{Tr}^{X'}_{A,B}(\rhd X' \times A \xrightarrow{\rhd g \times A} \rhd X \times A \xrightarrow{f} X' \times B).$$

Remark 3.3. The generalization for a symmetric monoidal category $(\mathscr{C}, \otimes, I, c)$ equipped with a pointed endofunctor $\rhd : \mathscr{C} \to \mathscr{C}$ requires the assumption that $\rhd$ is *comonoidal*, i.e., equipped with a morphism $m_I : \rhd I \to I$ and a natural transformation $m_{X,Y} : \rhd(X \times Y) \to \rhd X \times \rhd Y$ satisfying the usual coherence conditions. In fact, in the formulation of Vanishing (II) we used that in every category the product $\times$ is comonoidal via $m_{X,Y} = \mathsf{can}$.

Construction 3.4. 1. Let $(\mathscr{C}, \rhd, \mathsf{Tr})$ be a guarded traced category. Define a guarded fixpoint operator $\dagger_{\mathsf{Tr}} : \mathscr{C}(\rhd X \times A) \to \mathscr{C}(A, X)$ by

$$f^{\dagger_{\mathsf{Tr}}} = \mathsf{Tr}^{X}_{A,X}(\rhd X \times A \xrightarrow{\langle f, f\rangle} X \times X)) : A \to X.$$

2. Conversely, suppose $(\mathscr{C}, \rhd, \dagger)$ is a guarded fixpoint category. Define $\mathsf{Tr}_{\dagger}{}^{X}_{A,B} : \mathscr{C}(\rhd X \times A, X \times B) \to \mathscr{C}(A, B)$ by setting for every $f : \rhd X \times A \to X \times B$

$$\mathsf{Tr}_{\dagger}{}^{X}_{A,B}(f) = (A \xrightarrow{\langle (\pi_\ell \cdot f)^{\dagger}, A\rangle} X \times A \xrightarrow{\quad p_X \times A \quad} \rhd X \times A \xrightarrow{f} X \times B \xrightarrow{\pi_r} B).$$

The main result in this section states that the category $\mathscr{C}$ is guarded traced iff it is a guarded Conway category:

Theorem 3.5. *1. Whenever $(\mathscr{C}, \rhd, \mathsf{Tr})$ is a guarded traced category, $(\mathscr{C}, \rhd, \dagger_{\mathsf{Tr}})$ is a guarded Conway category. Furthermore, $\mathsf{Tr}_{\dagger_{\mathsf{Tr}}}$ is the original operator Tr.*

2. Whenever $(\mathscr{C}, \rhd, \dagger)$ is a guarded Conway category, $(\mathscr{C}, \rhd, \mathsf{Tr}_{\dagger})$ is guarded traced. Furthermore, $\dagger_{\mathsf{Tr}_{\dagger}}$ is the original operator $\dagger$.

The proof details are similar to the proof details for ordinary fixpoint operators and traced cartesian categories (see Hasegawa [15]). Here one has to stick $\rhd$ in "all the right places" in all the necessary verifications of the axioms for trace and dagger, respectively. However, some of proof steps, in particular the derivation of a guarded version of the so-called Bekič identity require some creativity; it is not a completely automatic adaptation.

Hasegawa related uniformity of trace to uniformity of dagger and we can do the same in the guarded setup. Recall that in iteration theories uniformity (called *functorial dagger implication*) plays an important role. On the one hand, this quasiequation implies the so-called *commutative identities*, an infinite set of equational axioms that are added to the Conway axioms in order to yield a complete axiomatization of fixpoint operators in domains. On the other hand, most examples of iteration theories actually satisfy uniformity, and so uniformity gives a convenient sufficient condition to verify that a given Conway theory is actually an iteration theory.

Definition 3.6. A guarded trace operator Tr is called uniform if for every morphism $f : \rhd X \times A \to X \times B$, $f' : \rhd X' \times A \to X' \times B$ and $h : X \to X'$ we have

$$\begin{tikzcd} \rhd X \times A \arrow[r, "f"] \arrow[d, "\rhd h \times A"'] & X \times B \arrow[d, "h \times B"] \\ \rhd X' \times A \arrow[r, "f'"'] & X' \times B \end{tikzcd} \qquad \Longrightarrow \qquad \mathsf{Tr}^{X}_{A,B}(f) = \mathsf{Tr}^{X'}_{A,B}(f') : A \to B.$$

Theorem 3.7. *1. Whenever $(\mathscr{C}, \rhd, \mathsf{Tr})$ is a uniform guarded traced category, $\dagger_{\mathsf{Tr}}$ is a uniform guarded Conway operator.*

2. Whenever $(\mathscr{C}, \rhd, \dagger)$ is a uniform guarded Conway category, $\mathsf{Tr}_{\dagger}$ is a uniform guarded trace operator.

Remark 3.8. Actually, Hasegawa proved a slightly stronger statement concerning uniformity then what we stated in Theorem 3.7; he showed that a Conway operator is uniform w.r.t. any fixed morphism $h : X \to X'$ (i.e. satisfies uniformity just for h) iff the corresponding trace operator is uniform w.r.t. this morphism h. The proof is somewhat more complicated and in our guarded setting we leave this as an exercise to the reader.

Finally, let us note that the bijective correspondence between guarded Conway operators and guarded trace operators established in Theorem 3.5 yields an isomorphism of the (2-)categories of (small) guarded Conway categories and guarded traced (cartesian) categories. The corresponding notions of morphisms are, of course, as expected:

Definition 3.9.

1. $F : (\mathscr{C}, \rhd^{\mathscr{C}}, \dagger) \to (\mathscr{D}, \rhd^{\mathscr{D}}, \ddagger)$ is a morphism of guarded Conway categories whenever $F : \mathscr{C} \to \mathscr{D}$ is a finite-product-preserving functor satisfying

$$\begin{tikzcd} \mathscr{C} \arrow[r, "\rhd^{\mathscr{C}}"] \arrow[d, "F"'] & \mathscr{C} \arrow[d, "F"] \\ \mathscr{D} \arrow[r, "\rhd^{\mathscr{D}}"'] & \mathscr{D} \end{tikzcd} \quad \text{and} \quad p^{\mathscr{D}}_{FX} = F(p^{\mathscr{C}}_{X}) : FX \to \rhd^{\mathscr{D}} FX = F(\rhd^{C} X), \tag{3.1}$$

and preserving dagger, i.e., for every $f : \rhd X \times A \to X$ we have

$$F(f^{\dagger}) = (\rhd^{\mathscr{D}} FX \times FA \cong F(\rhd^{\mathscr{C}} X \times A) \xrightarrow{Ff} FX)^{\ddagger}.$$

2. A morphism $F : (\mathscr{C}, \rhd^{\mathscr{C}}, \mathsf{Tr}_{\mathscr{C}}) \to (\mathscr{D}, \rhd^{\mathscr{D}}, \mathsf{Tr}_{\mathscr{D}})$ is a finite-product-preserving $F : \mathscr{C} \to \mathscr{D}$ satisfying (3.1) above and preserving the trace operation: for every $f : \rhd^{\mathscr{C}} X \times A \to X \times B$ in $\mathscr{C}$ we have

$$F(\mathsf{Tr}^{X}_{\mathscr{C}\, A,B}(f)) = \mathsf{Tr}^{FX}_{\mathscr{D}\, FA,FB}(\rhd^{\mathscr{D}} FX \times FA \cong F(\rhd^{\mathscr{C}} X \times A) \xrightarrow{Ff} F(X \times B) \cong FX \times FB).$$

Corollary 3.10. *The (2-)categories of guarded Conway categories and of guarded traced (cartesian) categories are isomorphic.*

4 Conclusions and Future Work

We have made the first steps in the study of equational properties of guarded fixpoint operators popular in the recent literature, e.g., [23, 24, 4, 7, 9, 19, 18, 9, 5]. We began with an extensive list of examples, including both those already discussed in the above references and some whose connection with the "later" modality has not seemed obvious so far—e.g., Example 2.4.6 or completely iterative theories in Section 2.2. Furthermore, we formulated the four Conway properties and uniformity in analogy to the respective properties in iteration theories and we showed them to be sound w.r.t. all models discussed in Section 2. In particular, Theorem 2.16 proved that our axioms hold in all categories with a unique guarded dagger. In Theorem 3.5, we have a generalization of a result by Hasegawa for ordinary fixpoint operators: we proved that to give a (uniform) guarded fixpoint operator satisfying the Conway axioms is equivalent to giving a (uniform) guarded trace operator on the same category.

Our paper can be considered as a work in progress report. Our aim is to eventually arrive at completeness results similar to the ones on iteration theories. We do not claim that the axioms we presented are complete. In the unguarded setting, completeness is obtained by adding to the Conway axioms an infinite set of equational axioms called the *commutative identities*, see [10, 25]. We did not consider those here, but we considered the quasi-equational property of uniformity which implies the commutative identities and is satisfied in most models of interest. Only further research can show whether this property can ensure completeness in the guarded setup or one needs to postulate stronger ones.

Other future work pertains to a syntactic type-theoretic presentation of the axioms we studied and a description of a classifying guarded Conway category.

Concerning further models of guarded fixpoint operators, it would be worthwhile to consider fixpoint monads of Crole and Pitts [11] more closely. These generalize our example of the category CPO with the lifting monad. One can prove that any fixpoint monad induces a guarded fixpoint operator satisfying parameter and simplified composition identities as well as uniformity. However, proving the double dagger identity in the general case is an open problem.

It would also be interesting to obtain examples of guarded traced monoidal categories which are not ordinary traced monoidal categories and which do not arise from guarded Conway categories. Traces w.r.t. a trace ideal as considered by Abramsky, Blute and Panangaden [1] might be a good starting point.

Acknowledgements We would like to acknowledge an inspiring discussion with Erwin R. Catesbeiana on (un-)productive (non-)termination. We would also like to thank in general William and Arthur for their very insistence on major modal undertones in modern modelling of this phenomenon.

References

[1] Samson Abramsky, Richard Blute & Prakash Panangaden (1999): *Nuclear and Trace Ideals in Tensored $*$-Categories*. *J. Pure Appl. Algebra* 143(1–3), pp. 3–47, doi:10.1016/S0022-4049(98)00106-6.

[2] Peter Aczel, Jiří Adámek, Stefan Milius & Jiří Velebil (2003): *Infinite Trees and Completely Iterative Theories: A Coalgebraic View*. *Theoret. Comput. Sci.* 300, pp. 1–45, doi:10.1016/S0304-3975(02)00728-4.

[3] Jiří Adámek & Stefan Milius (2006): *Terminal Coalgebras and Free Iterative Theories*. *Inform. and Comput.* 204, pp. 1139–1172, doi:10.1016/j.ic.2005.11.005.

[4] Andrew W. Appel, Paul-André Melliès, Christopher D. Richards & Jérôme Vouillon (2007): *A very modal model of a modern, major, general type system*. In Martin Hofmann & Matthias Felleisen, editors: *POPL*, ACM, pp. 109–122. Available at `http://doi.acm.org/10.1145/1190216.1190235`.

[5] Robert Atkey & Conor McBride (2013): *Productive Coprogramming with Guarded Recursion*. Accepted for ICFP.

[6] Eric Badouel (1989): *Terms and infinite trees as monads over a signature*. *Lecture Notes Comput. Sci.* 351, pp. 89–103, doi:10.1007/3-540-50939-9_126.

[7] Nick Benton & Nicolas Tabareau (2009): *Compiling functional types to relational specifications for low level imperative code*. In Andrew Kennedy & Amal Ahmed, editors: *TLDI*, ACM, pp. 3–14. Available at `http://doi.acm.org/10.1145/1481861.1481864`.

[8] Lars Birkedal & Rasmus E. Møgelberg (2013): *Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes*. In: *Proceedings of LICS*, pp. 213–222, doi:10.1109/LICS.2013.27.

[9] Lars Birkedal, Rasmus E. Møgelberg, Jan Schwinghammer & Kristian Støvring (2012): *First Steps in Synthetic Guarded Domain Theory: Step-Indexing in the Topos of Trees*. *Logical Methods in Computer Science* 8(4:1), pp. 1–45, doi:10.2168/LMCS-8(4:1)2012.

[10] Stephen L. Bloom & Zoltán Ésik (1993): *Iteration Theories: the equational logic of iterative processes*. EATCS Monographs on Theoretical Computer Science, Springer.

[11] Roy L. Crole & Andrew M. Pitts (1992): *New Foundations for Fixpoint Computations: FIX-Hyperdoctrines and FIX-Logic*. *Inform. and Comput.* 98(2), pp. 171–210, doi:10.1016/0890-5401(92)90018-B.

[12] Pietro Di Gianantonio & Marino Miculan (2004): *Unifying Recursive and Co-recursive Definitions in Sheaf Categories*. In Igor Walukiewicz, editor: *Foundations of Software Science and Computation Structures*, *Lecture Notes in Computer Science* 2987, Springer Berlin / Heidelberg, pp. 136–150. Available at `http://dx.doi.org/10.1007/978-3-540-24727-2_11`. 10.1007/978-3-540-24727-2_11.

[13] Calvin C. Elgot (1975): *Monadic Computation and Iterative Algebraic Theories*. In H. E. Rose & J. C. Sheperdson, editors: *Logic Colloquium '73*, 80, North-Holland Publishers, Amsterdam, pp. 175–230, doi:10.1007/978-1-4613-8177-8_6.

[14] Calvin C. Elgot, Stephen L. Bloom & Ralph Tindell (1978): *On the algebraic structure of rooted trees*. *J. Comput. System Sci.* 16, pp. 362–399, doi:10.1007/978-1-4613-8177-8_7.

[15] Masahito Hasegawa (1999): *Models of Sharing Graphs: A Categorical Semantics of let and letrec*. Distinguished Dissertation Series, Springer, doi:10.1007/978-1-4471-0865-8.

[16] Masihito Hasegawa (1997): *Recursion from Cyclic Sharing: Traced Monoidal Categories and Models of Cyclic Lambda Calculi*. In: *Proc. 3rd International Conference on Typed Lambda Calculi and Applications*, *Lecture Notes Comput. Sci.* 1210, Springer-Verlag, pp. 196–213, doi:10.1007/3-540-62688-3_37.

[17] André Joyal, Ross Street & Dominic Verity (1996): *Traced Monoidal Categories*. *Math. Proc. Cambridge Philos. Soc.* 119(3), pp. 447–468, doi:10.1017/S0305004100074338.

[18] Neelakantan R. Krishnaswami & Nick Benton (2011): *A semantic model for graphical user interfaces*. In Manuel M. T. Chakravarty, Zhenjiang Hu & Olivier Danvy, editors: *ICFP*, ACM, pp. 45–57. Available at `http://doi.acm.org/10.1145/2034773.2034782`.

[19] Neelakantan R. Krishnaswami & Nick Benton (2011): *Ultrametric Semantics of Reactive Programs*. In: *LICS*, IEEE Computer Society, IEEE Computer Society, pp. 257–266. Available at `http://dx.doi.org/10.1109/LICS.2011.38`.

[20] Stefan Milius (2005): *Completely Iterative Algebras and Completely Iterative Monads*. *Inform. and Comput.* 196, pp. 1–41, doi:10.1016/j.ic.2004.05.003.

[21] Robin Milner (1989): *Communication and Concurrency*. International Series in Computer Science, Prentice Hall.

[22] Philip S. Mulry (1994): *Lifting Theorems for Kleisli Categories*. In S. Brookes, M. Main, A. Melton, M. Mislove & D. Schmidt, editors: *Proc. Mathematical Foundations of Programming Semantics (MFPS'93)*, *Lecture Notes Comput. Sci.* 802, Springer, pp. 304–319, doi:10.1007/3-540-58027-1_15.

[23] Hiroshi Nakano (2000): *A Modality for Recursion*. In: *LICS*, IEEE Computer Society, pp. 255–266, doi:10.1109/LICS.2000.855774.

[24] Hiroshi Nakano (2001): *Fixed-Point Logic with the Approximation Modality and Its Kripke Completeness*. In Naoki Kobayashi & Benjamin C. Pierce, editors: *TACS*, *Lecture Notes in Computer Science* 2215, Springer, pp. 165–182, doi:10.1007/3-540-45500-0_8.

[25] Alex Simpson & Gordon D. Plotkin (2000): *Complete axioms for categorical fixed-point operators*. In: *Proc. 15th Symposium on Logic in Computer Science (LICS'00)*, IEEE Computer Society, pp. 30–41, doi:10.1109/LICS.2000.855753.

Łukasiewicz μ-calculus

Matteo Mio
CWI, Amsterdam (NL)
miomatteo@gmail.com

Alex Simpson
LFCS, School of Informatics
University of Edinburgh
Alex.Simpson@ed.ac.uk

The paper explores properties of *Łukasiewicz μ-calculus*, a version of the quantitative/probabilistic modal μ-calculus containing both weak and strong conjunctions and disjunctions from Łukasiewicz (fuzzy) logic. We show that this logic encodes the well-known probabilistic temporal logic **PCTL**. And we give a model-checking algorithm for computing the rational denotational value of a formula at any state in a finite rational probabilistic nondeterministic transition system.

1 Introduction

Among logics for expressing properties of nondeterministic (including concurrent) processes, represented as transition systems, Kozen's modal μ-calculus [15] plays a fundamental rôle. It subsumes other temporal logics of processes, such as **LTL**, **CTL** and **CTL***. It does not distinguish bisimilar processes, but separates (finite) non-bisimilar ones. More generally, by a remarkable result of Janin and Walukiewicz [14], it is exactly as expressive as the bisimulation-invariant fragment of monadic second-order logic. Furthermore, there is an intimate connection with parity games, which offers an intuitive reading of fixed-points, and underpins the existing technology for model-checking μ-calculus properties.

For many purposes, it is useful to add probability to the computational model, leading to probabilistic nondeterministic transition systems, cf. [23]. Among the different approaches that have been followed to developing analogues of the modal μ-calculus in this setting, the most significant is that introduced independently by Huth and Kwiatkowska [12] and by Morgan and McIver [22], under which a *quantitative* interpretation is given, with formulas denoting values in $[0,1]$. This quantitative setting permits several variations. In particular, three different quantitative extensions of conjunction from booleans to $[0,1]$ (with 0 as false and 1 as true) arise naturally [12]: minimum, $\min(x,y)$; multiplication, xy; and the strong conjunction (a.k.a. Łukasiewicz t-norm) from Łukasiewicz fuzzy logic, $\max(x+y-1,\,0)$. In each case, there is a dual operator giving a corresponding extension of disjunction: maximum, $\max(x,y)$; comultiplication, $x+y-xy$; and Łukasiewicz strong disjunction, $\min(x+y,\,1)$. The choice of min and max for conjunction and disjunction is particularly natural, since the corresponding μ-calculus, called qLμ in [18], has an interpretation in terms of 2-player *stochastic* parity games, which extends the usual parity-game interpretation of the ordinary modal μ-calculus. This allows the real number denoted by a formula to be understood as the *value* of the associated game [18, 20].

The present paper contributes to a programme of ongoing research, one of whose overall aims is to investigate the extent to which quantitative μ-calculi play as fundamental a rôle in the probabilistic setting as that of Kozen's μ-calculus in the nondeterministic setting. The logic qLμ, with min/max as conjunction/disjunction, is insufficiently expressive. For example, it cannot encode the standard probabilistic temporal logic **PCTL** of [2]. Nevertheless, richer calculi can be obtained by augmenting qLμ with the other alternatives for conjunction/disjunction, to be used in combination with max and min. Such extensions were investigated by the first author in [21, 19], where the game-theoretic interpretation was generalized to accommodate the new operations.

D. Baelde and A. Carayol (Eds.): Fixed Points
in Computer Science 2013 (FICS 2013)
EPTCS 126, 2013, pp. 87–104, doi:10.4204/EPTCS.126.7

In this paper, we focus on a calculus containing two different interpretations of conjunction and disjunction: min and max (written as $\sqcap$ and $\sqcup$) and the Łukasiewicz operations (written as $\odot$ and $\oplus$). In addition, as is natural in the quantitative setting, we include a basic operation for multiplying the value of a formula by a rational constant in $[0,1]$. Since these operations are all familiar from Łukasiewicz fuzzy logic (see, e.g., [11]), we call the resulting logic *Łukasiewicz μ-calculus* (Łμ).

As our first contribution, we show that the standard probabilistic temporal logic **PCTL** [2] can be encoded in Łμ. A similar translation was originally given in the first author's PhD thesis [19], where **PCTL** was translated into a quantitative μ-calculus containing all three pairs of quantitative conjunction/disjunction operations in combination. Here, we streamline the treatment by implementing the observation that the (co)multiplication operations are not required once the Łukasiewicz operations are in place. In fact, all that is needed is the encodability of certain *threshold modalities*, see Remark 3.6 below.

An advantage of the Łukasiewicz μ-calculus considered in the present paper is that it enjoys the property that the value of a formula in a finite rational model is rational, a property which does not hold when the (co)multiplication operations are included in the logic. As our second contribution, we exploit this property by giving a (quantitative) model-checking algorithm that computes the value of a Łμ formula at a state in a finite rational probabilistic nondeterministic transition system. The algorithm adapts the approximation-based approach to nested fixed-point calculation to our quantitative calculus.

One could combine our two contributions and obtain a new model-checking algorithm for **PCTL**. But this is not advisable since the complexity bounds we obtain for model-checking Łμ are abysmal. The positive messages of this paper are rather that **PCTL** fits into the conceptually appealing framework of quantitative μ-calculi, and that this framework is itself algorithmically approachable.

2 Technical background

Definition 2.1. Given a set S we denote with $\mathscr{D}(S)$ the set of *(discrete) probability distributions* on S defined as $\mathscr{D}(S) = \{d : S \to [0,1] \mid \sum_{s\in S} d(s) = 1\}$. We say that $d \in \mathscr{D}(S)$ is *rational* if $d(s)$ is a rational number, for all $s \in S$.

Definition 2.2. A *probabilistic nondeterministic transition system* (PNTS) is a pair $(S, \to)$ where S is a set of states and $\to \subseteq S \times \mathscr{D}(S)$ is the *accessibility* relation. We write $s \not\to$ if $\{d \mid s \to d\} = \emptyset$. A PNTS $(S, \to)$ is *finite rational* if S is finite and $\bigcup_{s\in S}\{d \mid s \to d\}$ is a finite set of rational probability distributions.

We now introduce the novel logic Łμ which extends the probabilistic (or quantitative) modal μ-calculus (qLμ) of [12, 22, 18, 5].

Definition 2.3. The logic Łμ is generated by the following grammar:

$$\phi ::= X \mid P \mid \overline{P} \mid q\phi \mid \phi \sqcup \phi \mid \phi \sqcap \phi \mid \phi \oplus \phi \mid \phi \odot \phi \mid \ \Diamond\phi \mid \ \Box\phi \mid \mu X.\phi \mid \nu X.\phi \ \ ,$$

where q ranges over rationals in $[0,1]$, X over a countable set `Var` of variables and P over a set `Prop` of propositional letters which come paired with associated complements $\overline{P}$. As a convention we denote with $\underline{1}$ the formula $\nu X.X$ and with $\underline{q}$ the formula $q\underline{1}$.

Thus, Łμ extends the syntax of the probabilistic modal μ-calculus by the new pair of connectives ($\odot$, $\oplus$), which we refer to as *Łukasiewicz conjunction* and *disjunction*, respectively, and a form of *scalar multiplication* ($q\phi$) by rationals numbers in $[0,1]$. For mild convenience in the encoding of **PCTL** below, we consider a version with unlabelled modalities and propositional letters. However, the approach of this paper easily adapts to a labeled version of Łμ.

Formulas are interpreted over PNTS's as we now describe.

Definition 2.4. Given a PNTS $(S,\rightarrow)$, an *interpretation* for the variables and propositional letters is a function $\rho : (\mathtt{Var} \uplus \mathtt{Prop}) \rightarrow (S \rightarrow [0,1])$ such that $\rho(\overline{P})(x) = 1 - \rho(P)(x)$. Given a function $f : S \rightarrow [0,1]$ and $X \in \mathtt{Var}$ we define the interpretation $\rho[f/X]$ as $\rho[f/X](X) = f$ and $\rho[f/X](Y) = \rho(Y)$, for $X \neq Y$.

Definition 2.5. The semantics of a Łμ formula ϕ interpreted over $(S,\rightarrow)$ with interpretation ρ is a function $[\![\phi]\!]_\rho : S \rightarrow [0,1]$ defined inductively on the structure of ϕ as follows:

$$\begin{array}{ll}
[\![X]\!]_\rho = \rho(X) & [\![q\phi]\!]_\rho(x) = q \cdot [\![\phi]\!]_\rho(x) \\
[\![P]\!]_\rho = \rho(P) & [\![\overline{P}]\!]_\rho = 1 - \rho(P) \\
[\![\phi \sqcup \psi]\!]_\rho(x) = \max\{[\![\phi]\!]_\rho(x), [\![\psi]\!]_\rho(x)\} & [\![\phi \sqcap \psi]\!]_\rho(x) = \min\{[\![\phi]\!]_\rho(x), [\![\psi]\!]_\rho(x)\} \\
[\![\phi \oplus \psi]\!]_\rho(x) = \min\{1, [\![\phi]\!]_\rho(x) + [\![\psi]\!]_\rho(x)\} & [\![\phi \odot \psi]\!]_\rho(x) = \max\{0, [\![\phi]\!]_\rho(x) + [\![\psi]\!]_\rho(x) - 1\} \\
[\![\Diamond\phi]\!]_\rho(x) = \bigsqcup_{x \rightarrow d} \Big(\sum_{y \in X} d(y) [\![\phi]\!]_\rho(y) \Big) & [\![\Box\phi]\!]_\rho(x) = \bigsqcap_{x \rightarrow d} \Big(\sum_{y \in X} d(y) [\![\phi]\!]_\rho(y) \Big) \\
[\![\mu X.\phi]\!] = \mathrm{lfp}\big(f \mapsto [\![\phi]\!]_{\rho[f/X]}\big) & [\![\mu X.\phi]\!] = \mathrm{gfp}\big(f \mapsto [\![\phi]\!]_{\rho[f/X]}\big)
\end{array}$$

It is straightforward to verify that the interpretation of every operator is monotone, thus the existence of least and greatest points in the last two clauses is guaranteed by the the Knaster-Tarski theorem.

As customary in fixed-point logics, we presented the logic Łμ in positive normal form. A negation operation $\mathtt{dual}(\phi)$ can be defined on *closed* formulas by replacing every connective with its dual and $(q\,\phi)$ with $((1-q)\,\phi)$. It is simple to verify that $[\![\mathtt{dual}(\phi)]\!]_\rho(x) = 1 - [\![\phi]\!]_\rho(x)$.

Next, we introduce the syntax and the semantics of the logic **PCTL** of [2]. We refer to [1] for an extensive presentation of this logic.

The notions of *paths*, *schedulers* and *Markov runs* in a PNTS are at the basis of the logic **PCTL**.

Definition 2.6. For a given PNTS $\mathscr{L} = (S,\rightarrow)$ the binary relation $\rightsquigarrow_{\mathscr{L}} \subseteq S \times S$ is defined as follows: $\rightsquigarrow_{\mathscr{L}} = \{(s,t) \mid \exists d.(s \rightarrow d \;\wedge\; d(t) > 0)\}$. Note that $s \not\rightarrow$ if and only if $s \not\rightsquigarrow$. We refer to $(S,\rightsquigarrow)$ as the *graph underlying* $\mathscr{L}$.

Definition 2.7. A *path* in a PNTS $\mathscr{L} = (S,\rightarrow)$ is an ordinary path in the graph $(S,\rightsquigarrow)$, i.e., a finite or infinite sequence $\{s_i\}_{i \in I}$ of states such that $s_i \rightsquigarrow s_{i+1}$, for all $i+1 \in I$. We say that a path is *maximal* if either it is infinite or it is finite and its last entry is a state s_n without successors, i.e., such that $s_n \not\rightsquigarrow$. We denote with $\mathrm{P}(\mathscr{L})$ the set of all maximal paths in $\mathscr{L}$. The set $\mathrm{P}(\mathscr{L})$ is endowed with the topology generated by the basic open sets $U_{\vec{s}} = \{\vec{r} \mid \vec{s} \sqsubseteq \vec{r}\}$ where $\vec{s}$ is a finite sequence of states and $\sqsubseteq$ denotes the prefix relation on sequences. The space $\mathrm{P}(\mathscr{L})$ is always 0-dimensional, i.e., the basic sets $U_{\vec{s}}$ are both open and closed and thus form a Boolean algebra. We denote with $\mathrm{P}(s)$ the open set $U_{\{s\}}$ of all maximal paths having s as first state.

Definition 2.8. A *scheduler* in a PNTS $(S,\rightarrow)$ is a partial function σ from non-empty finite sequences $s_0.\dots.s_n$ of states to probability distributions $d \in \mathscr{D}(S)$ such that $\sigma(s_0.\dots.s_n)$ is not defined if and only if $s_n \not\rightarrow$ and, if σ is defined at $s_0.\dots.s_n$ with $\sigma(s_0.\dots.s_n) = d$, then $s_n \rightarrow d$ holds. A pair (s,σ) is called a *Markov run* in $\mathscr{L}$ and denoted by M^s_σ. It is clear that each Markov run M^s_σ can be identified with a (generally) infinite Markov chain (having a tree structure) whose vertices are finite sequences of states and having $\{s\}$ as root.

Markov runs are useful as they naturally induce probability measures on the space $\mathrm{P}(\mathscr{L})$.

Definition 2.9. Let $\mathscr{L} = (S,\rightarrow)$ be a PNTS and M^s_σ a Markov run. We define the measure m^s_σ on $\mathrm{P}(\mathscr{L})$ as the unique (by Carathéodory extension theorem) measure specified by the following assignment of basic open sets:

$$m^s_\sigma\big(\mathrm{U}_{s_0.\dots.s_n}\big) = \prod_{i=0}^{n-1} d_i(s_{i+1})$$

where $d_i = \sigma(s_0.\dots.s_i)$ and $\prod \emptyset = 1$. It is simple to verify that m^s_σ is a probability measure, i.e., $m^s_\sigma(\mathrm{P}(\mathscr{L})) = 1$. We refer to m^s_σ as the probability measure on $\mathrm{P}(\mathscr{L})$ induced by the Markov run M^s_σ.

We are now ready to specify the syntax and semantics of **PCTL**.

Definition 2.10. Let the letter P range over a countable set of propositional symbols `Prop`. The class of **PCTL** *state-formulas* ϕ is generated by the following two-sorted grammar:

$$\phi ::= \text{ true} \mid P \mid \neg\phi \mid \phi \vee \phi \mid \exists\psi \mid \forall\psi \mid \mathbb{P}^{\exists}_{\rtimes q}\psi \mid \ \mathbb{P}^{\forall}_{\rtimes q}\psi$$

with $q \in \mathbb{Q} \cap [0,1]$ and $\rtimes \in \{>, \geq\}$, where *path-formulas* ψ are generated by the simple grammar: $\psi ::= \ \circ\phi \mid \phi_1 \mathscr{U} \phi_2$. Adopting standard terminology, we refer to the connectives $\circ$ and $\mathscr{U}$ as the *next* and *until* operators, respectively.

Definition 2.11. Given a PNTS $(S, \to)$, a ***PCTL-interpretation*** for the propositional letters is a function $\rho : \texttt{Prop} \to 2^S$, where 2^S denotes the powerset of S.

Definition 2.12. Given a PNTS $(S, \to)$ and a **PCTL**-interpretation ρ for the propositional letters, the semantics $\llparenthesis \phi \rrparenthesis_\rho$ of a **PCTL** state-formula ϕ is a subset of S (i.e., $\llparenthesis \phi \rrparenthesis_\rho : S \to \{0,1\}$) defined by induction on the structure of ϕ as follows:

- $\llparenthesis \text{true} \rrparenthesis_\rho = S$, $\llparenthesis P \rrparenthesis_\rho = \rho(P)$, $\llparenthesis \phi_1 \vee \phi_2 \rrparenthesis_\rho = \llparenthesis \phi_1 \rrparenthesis_\rho \cup \llparenthesis \phi_2 \rrparenthesis_\rho$, $\llparenthesis \neg\phi \rrparenthesis_\rho = S \setminus \llparenthesis \phi \rrparenthesis_\rho$,
- $\llparenthesis \exists\psi \rrparenthesis_\rho(s) = 1$ if and only there exists $\vec{s} \in \mathrm{P}(s)$ such that that $\vec{s} \in [\![\psi]\!]$
- $\llparenthesis \forall\psi \rrparenthesis_\rho(s) = 1$ if and only forall $\vec{s} \in \mathrm{P}(s)$ it holds that $\vec{s} \in \llparenthesis \psi \rrparenthesis_\rho(\vec{s})$
- $\llparenthesis \mathbb{P}^{\exists}_{\rtimes q}\psi \rrparenthesis_\rho(s) = 1$ if and only $\big(\bigsqcup_\sigma m^s_\sigma(\llparenthesis \psi \rrparenthesis_\rho)\big) \rtimes q$
- $\llparenthesis \mathbb{P}^{\forall}_{\rtimes q}\psi \rrparenthesis_\rho(s) = 1$ if and only $\big(\bigsqcap_\sigma m^s_\sigma(\llparenthesis \psi \rrparenthesis_\rho)\big) \rtimes q$

where σ ranges over schedulers and the semantics $\llparenthesis \psi \rrparenthesis_\rho$ of path formulas, defined as a subset of $\mathrm{P}(\mathscr{L})$ (i.e., as a map $\llparenthesis \psi \rrparenthesis_\rho : \mathrm{P}(\mathscr{L}) \to \{0,1\}$) is defined as:

- $\llparenthesis \circ\phi \rrparenthesis_\rho(\vec{s}) = 1$ if and only if $|\vec{s}| \geq 2$ (i.e., $\vec{s} = s_0.s_1.\dots$) and $s_1 \in \llparenthesis \phi \rrparenthesis_\rho$,
- $\llparenthesis \phi_1 \mathscr{U} \phi_2 \rrparenthesis_\rho(\vec{s}) = 1$ if and only if $\exists n.\big((s_n \in \llparenthesis \phi_2 \rrparenthesis_\rho) \wedge \forall m < n.(s_m \in \llparenthesis \phi_1 \rrparenthesis_\rho)\big)$,

It is simple to verify that, for all path-formulas ψ, the set $\llparenthesis \psi \rrparenthesis_\rho$ is Borel measurable [1]. Therefore the definition is well specified. Note how the logic **PCTL** can express probabilistic properties, by means of the connectives $\mathbb{P}^{\forall}_{\rtimes q}$ and $\mathbb{P}^{\exists}_{\rtimes q}$, as well as (qualitative) properties of the graph underlying the PNTS by means of the quantifiers $\forall$ and $\exists$.

3 Encoding of PCTL

We prove in this section how **PCTL** can be seen as a simple fragment of Łμ by means of an explicit encoding. We first introduce a few useful macro formulas in the logic Łμ which, crucially, are not expressible in the probabilistic μ-calculus (qLμ).

Definition 3.1. Let ϕ be a (possibly open) Łμ formula. We define:

• $\mathbb{P}_{>0}\phi = \mu X.(X \oplus \phi)$ • $\mathbb{P}_{=1}\phi = \nu X.(X \odot \phi)$ • $\mathbb{P}_{>q}\phi = \mathbb{P}_{>0}(\phi \odot \underline{1-q})$ • $\mathbb{P}_{\geq q}\phi = \mathbb{P}_{=1}(\phi \oplus \underline{1-q})$

for $q \in \mathbb{Q} \cap (0,1)$. We write $\mathbb{P}_{\rtimes q}\phi$, for $q \in \mathbb{Q} \cap [0,1]$, to denote one of the four cases.

The following proposition describes the denotational semantics of these macro formulas.

Proposition 3.2. *Let $(S, \to)$ be a PNTS, ϕ a Łμ formula and ρ an interpretation of the variables. Then it holds that:*

$$[\![\mathbb{P}_{\bowtie q}\phi]\!]_\rho(s) = \begin{cases} 1 & \textit{if } [\![\phi]\!]_\rho(s) \bowtie q \\ 0 & \textit{otherwise} \end{cases}$$

Proof. For the case $\mathbb{P}_{>0}\phi$, observe that the map $x \mapsto q \oplus x$, for a fixed $q \in [0,1]$, has 1 as unique fixed point when $q > 0$, and 0 as the least fixed point when $q = 0$. The result then follows trivially. Similarly for $\mathbb{P}_{=1}\phi$. The other cases are trivial. □

The following lemma is also useful.

Lemma 3.3. *Let $(S, \rightarrow)$ be a PNTS, ϕ a Łμ formula and ρ an interpretation of the variables. Then:*

- $[\![\mathbb{P}_{>0}(\Diamond X)]\!]_\rho(s) = 1$ *iff* $\exists t.\big(s \rightsquigarrow t \wedge \rho(X)(t) > 0\big)$
- $[\![\mathbb{P}_{=1}(\Box X)]\!]_\rho(s) = 1$ *iff* $\forall t.\big(s \rightsquigarrow t \rightarrow \rho(X)(t) = 1\big)$

Proof. Note that $[\![\Diamond X]\!]_\rho(s) > 0$ iff there exists $s \rightarrow d$ such that $\sum_{t\in S} d(t)\rho(X)(t) > 0$ holds. This is the case iff $d(t) > 0$ (i.e., $s \rightsquigarrow t$) and $\rho(X)(t) > 0$, for some $t \in S$. The result then follows by Proposition 3.2. The case for $\mathbb{P}_{=1}(\Box X)$ is similar. □

Remark 3.4. When considering $\{0,1\}$-valued interpretations for X, the macro formula $\mathbb{P}_{>0}\Diamond$ expresses the meaning of the diamond modality in classical modal logic with respect to the graph $(S, \rightsquigarrow)$ underlying the PNTS. Similarly, $\mathbb{P}_{=1}\Box$ corresponds to the the classical box modality.

We are now ready to define the encoding of **PCTL** into Łμ.

Definition 3.5. We define the encoding **E** from **PCTL** formulas to closed Łμ formulas (where $\boxdot\phi$ stands for the Łμ formula $\Box\phi \sqcap \Diamond\underline{1}$), by induction on the structure of the **PCTL** formulas ϕ as follows:

1. $\mathbf{E}(P) = P$,
2. $\mathbf{E}(\text{true}) = \underline{1}$,
3. $\mathbf{E}(\phi_1 \vee \phi_2) = \mathbf{E}(\phi_1) \sqcup \mathbf{E}(\phi_2)$,
4. $\mathbf{E}(\neg\phi) = \text{dual}(\mathbf{E}(\phi))$,
5. $\mathbf{E}(\exists(\circ\phi)) = \mathbb{P}_{>0}\big(\Diamond\mathbf{E}(\phi)\big)$,
6. $\mathbf{E}(\forall(\circ\phi)) = \mathbb{P}_{=1}\big(\boxdot\,\mathbf{E}(\phi)\big)$,
7. $\mathbf{E}(\exists(\phi_1\ \mathscr{U}\ \phi_2)) = \mu X.\Big(\mathbf{E}(\phi_2) \sqcup \big(\mathbf{E}(\phi_1) \sqcap \mathbb{P}_{>0}(\Diamond X)\big)\Big)$,
8. $\mathbf{E}(\forall(\phi_1\ \mathscr{U}\ \phi_2)) = \mu X.\Big(\mathbf{E}(\phi_2) \sqcup \big(\mathbf{E}(\phi_1) \sqcap \mathbb{P}_{=1}(\boxdot X)\big)\Big)$,
9. $\mathbf{E}(\mathbb{P}^{\exists}_{\bowtie q}(\circ\phi)) = \mathbb{P}_{\bowtie q}\big(\Diamond\mathbf{E}(\phi)\big)$,
10. $\mathbf{E}(\mathbb{P}^{\forall}_{\bowtie q}(\circ\phi)) = \mathbb{P}_{\bowtie q}\big(\boxdot\,\mathbf{E}(\phi)\big)$,
11. $\mathbf{E}(\mathbb{P}^{\exists}_{\bowtie q}(\phi_1 \mathscr{U} \phi_2)) = \mathbb{P}_{\bowtie q}\Big(\mu X.\Big(\mathbf{E}(\phi_2) \sqcup \big(\mathbf{E}(\phi_1) \sqcap \Diamond X\big)\Big)\Big)$,
12. $\mathbf{E}(\mathbb{P}^{\forall}_{\bowtie q}(\phi_1 \mathscr{U} \phi_2)) = \mathbb{P}_{\bowtie q}\Big(\mu X.\Big(\mathbf{E}(\phi_2) \sqcup \big(\mathbf{E}(\phi_1) \sqcap \boxdot X\big)\Big)\Big)$,

Note that Case 4 is well defined since $\mathbf{E}(\phi)$ is closed by construction.

Remark 3.6. The only occurrences of Łukasiewicz operators $\{\oplus, \odot\}$ and scalar multiplication $(q\phi)$ in encoded **PCTL** formulas appear in the formation of the macro formulas $\mathbb{P}_{\rtimes q}(\_)$ which we refer to as *threshold modalities*. Thus, **PCTL** can be also seen as a fragment of qLμ extended with threshold modalities as primitive operations. With the aid of these modalities the encoding is, manifestly, a straightforward adaption of the standard encoding of CTL into the modal μ-calculus (see, e.g., [24]).

We are now ready to prove the correctness theorem which holds for arbitrary models.

Theorem 3.7. *For every PNTS $(S, \rightarrow)$, **PCTL**-interpretation $\rho : \mathrm{Prop} \rightarrow (S \rightarrow \{0,1\})$ of the propositional letters and **PCTL** formula ϕ, the equality $\llparenthesis \phi \rrparenthesis_\rho(s) = \llbracket \mathbf{E}(\phi) \rrbracket_\rho(s)$ holds, for all $s \in S$.*

Proof (outline). The proof goes by induction on the complexity of ϕ. Cases 1–4 of Definition 3.5 are trivial. Case 5 follows directly from Lemma 3.3. Observing that $\llbracket \boxdot \phi \rrbracket_\rho(s) = 0$ if $s \not\rightarrow$ and $\llbracket \boxdot \phi \rrbracket_\rho(s) = \llbracket \Box \phi \rrbracket_\rho(s)$ otherwise, also Case 6 is a consequence of Lemma 3.3. Consider cases 7 and 8. The encoding is of the form $\mu X.(F \sqcup (G \sqcap H(X))$, where F and G (by induction hypothesis) and $H(X)$ (by Proposition 3.2) are all $\{0,1\}$-valued. Therefore the functor $f \mapsto \llbracket F \sqcup (G \sqcap H(X)) \rrbracket_{\rho[f/X]}$ maps $\{0,1\}$-valued functions to $\{0,1\}$-valued functions and has only $\{0,1\}$-valued fixed-points. It then follows by Remark 3.4 that the correctness of the encoding for these two cases can be proved with the standard technique used to prove the correctness of the encoding of CTL into Kozen's μ-calculus (see, e.g., [24]). Consider Case 9. It is immediate to verify that $\bigsqcup_\sigma \{m^s_\sigma(U)\}$, where $U = \llparenthesis \circ \phi \rrparenthesis_\rho = \bigcup \{U_{\{s.t\}} \mid t \in \llparenthesis \phi \rrparenthesis_\rho\}$, is equal (by induction hypothesis) to $\llbracket \Diamond \mathbf{E}(\phi) \rrbracket_\rho(s)$. The desired equality $\llparenthesis \mathbb{P}^{\exists}_{\rtimes q} \circ \phi \rrparenthesis_\rho = \llbracket \mathbb{P}_{\rtimes q} \Diamond \mathbf{E}(\phi) \rrbracket_\rho$ then follows by Proposition 3.2. Case 10 is similar. The two cases 11 and 12 are similar, thus we just consider case 11. Let $\phi = \mathbb{P}^{\exists}_{\rtimes q}(\psi)$ and $\psi = \phi_1 \mathscr{U} \phi_2$. We denote with Ψ the set of paths $\llparenthesis \psi \rrparenthesis_\rho$. Denote by $F(X)$ the formula $\mathbf{E}(\phi_2) \sqcup (\mathbf{E}(\phi_1) \sqcap \Diamond X)$. It is clearly sufficient to prove that the equality $\bigsqcup_\sigma \{m^s_\sigma(\Psi)\} = \llbracket \mu X.F(X)) \rrbracket_\rho(s)$ holds. Note that $\mu X.F(X)$ can be expressed as an equivalent qLμ formulas by substituting the closed subformulas $\mathbf{E}(\phi_1)$ and $\mathbf{E}(\phi_2)$ with two fresh atomic predicates P_i with interpretations $\rho(P_i) = \llbracket \mathbf{E}(\phi_i) \rrbracket$. The equality can then be proved by simple arguments based on the game-semantics of qLμ (see, e.g., [18] and [20]), similar to the ones used to prove that the Kozen's μ-calculus formula $\mu X.(P_2 \vee (P_1 \wedge \Diamond X))$ has the same denotation of the CTL formula $\exists(P_1 \mathscr{U} P_2)$ (see, e.g., [24]). $\Box$

4 Łukasiewicz μ-terms

The aim of the second half of the paper is to show how to compute the (rational) denotational value of a Łμ formula at any state in a finite rational probabilistic transition system. In this section, we build the main machinery for doing this, based on a system of fixed-point terms for defining monotone functions from $[0,1]^n$ to $[0,1]$. The syntax of *(Łukasiewicz) μ-terms* is specified by the grammar:

$$t ::= x \mid qt \mid t \sqcup t \mid t \sqcap t \mid t \oplus t \mid t \odot t \mid \mu x.t \mid \nu x.t$$

Again, q ranges over rationals in $[0,1]$. As expected, the μ and ν operators bind their variables. We write $t(x_1, \dots, x_n)$ to mean that all free variables of t are contained in $\{x_1, \dots, x_n\}$.

The *value* $t(\vec{r})$ (we eschew semantic brackets) of a μ-term $t(x_1, \dots, x_n)$ applied to a vector $(r_1, \dots, r_n) \in [0,1]^n$ is defined inductively in the obvious way, cf. Definition 2.5. (Indeed, μ-terms form a fragment of Łμ of formulas whose value is independent of the transition system in which they are interpreted.)

In Section 6, the model-checking task will be reduced to the problem of computing the value of μ-terms. The fundamental property that allows such values to be computed is that, for any μ-term $t(x_1, \dots, x_n)$ and vector of rationals $(q_1, \dots, q_n)$, the value of $t(\vec{q})$ is rational and can be computed from

t and q. One way of establishing this result is by a simple reduction to the first-order theory of rational linear arithmetic, which provides an indirect means of computing the value of $t(\vec{q})$. The current section presents a brief outline of this approach. After this, in Section 5, we provide an alternative direct algorithm for computing $t(\vec{q})$.

A *linear expression* in variables $x_1,\ldots,x_n$ is an expression

$$q_1x_1 + \cdots + q_nx_n + q$$

where $q_1,\ldots,q_n,q$ are real numbers. In the sequel, we only consider *rational* linear expressions, in which $q_1,\ldots,q_n,q$ are all rational, and we henceforth assume this property without mention. We write $e(x_1,\ldots,x_n)$ if e is a linear expression in $x_1,\ldots,x_n$, in which case, given real numbers $r_1,\ldots,r_n$, we write $e(\vec{r})$ for the value of the expression when the variables $\vec{x}$ take values $\vec{r}$. We also make use of the closure of linear expressions under substitution: given $e(x_1,\ldots,x_n)$ and $e_1(y_1,\ldots,y_m),\ldots,e_n(y_1,\ldots,y_m)$, we write $e(e_1,\ldots,e_n)$ for the evident substituted expression in variables $y_1,\ldots,y_m$ (which is defined formally by multiplying out and adding coefficients).

The first-order theory of *rational linear arithmetic* has linear expressions as terms, and strict and non-strict inequalities between linear expressions,

$$e_1 < e_1 \qquad e_1 \leq e_2 \ , \tag{1}$$

as atomic formulas. Equality can be expressed as the conjunction of two non-strict inequalities and the negation of an atomic formula can itself be expressed as an atomic formula. The truth of a first-order formula is given via its interpretation in the reals, or equivalently in the rationals since the inclusion of the latter in the former is an elementary embedding. The theory enjoys quantifier elimination [8].

Proposition 4.1. *For every Łukasiewicz μ-term $t(x_1,\ldots,x_n)$, its graph $\{(\vec{x},y) \in [0,1]^{n+1} \mid t(\vec{x}) = y\}$ is definable by a formula $F_t(x_1,\ldots,x_n,y)$ in the first-order theory of rational linear arithmetic, where F_t is computable from t.*

Proof. The proof is a straightforward induction on the structure of t. We consider two cases, in order to illustrate the simple manipulations used in the construction of F_t.

If t is $t_1 \oplus t_2$ then F_t is the formula

$$\exists z_1,z_2.\, F_{t_1}(\vec{x},z_1) \wedge F_{t_2}(\vec{x},z_2) \wedge ((z_1+z_2 \leq 1 \wedge z = z_1+z_2) \vee (1 \leq z_1+z_2 \wedge z = 1))$$

If t is $\mu x_{n+1}.t'$ then F_t is the formula

$$F_{t'}(x_1,\ldots,x_n,y,y) \wedge \forall z.\, F_{t'}(x_1,\ldots,x_n,z,z) \rightarrow y \leq z \ .$$

□

Proposition 4.1 provides the following method of computing the value $t(\vec{q})$ of μ-term $t(x_1,\ldots,x_n)$ at a rational vector $(q_1,\ldots,q_n) \in [0,1]^n$. First construct $F_t(x_1,\ldots,x_n,y)$. Next, perform quantifier elimination to obtain an equivalent quantifier-free formula $G_t(x_1,\ldots,x_n,y)$, and consider its instantiation $G_t(q_1,\ldots,q_n,y)$ at $\vec{q}$. (Alternatively, obtain an equivalent formula $G_t^{\vec{q}}(y)$ by performing quantifier elimination on $F_t(q_1,\ldots,q_n,y)$.) By performing obvious simplifications of atomic formulas in one variable, $G_t(q_1,\ldots,q_n,y)$ reduces to a boolean combination of inequalities each having one of the following forms

$$y \leq q \qquad y < q \qquad y \geq q \qquad y > q \ .$$

By the correctness of G_t there must be a unique rational satisfying the boolean combination of constraints, and this can be extracted in a straightforward way from $G_t(q_1,\dots,q_n,y)$.

We give a crude (but sufficient for our purposes) complexity analysis of the above procedure. In general, for a μ-term t of length u containing v fixed points, the length of F_t is bounded by $2^v uc$, for some constant c. The quantifier-elimination procedure in [8], when given a formula of length l as input produces a formula of length at most 2^{dl} as output, for some constant d, and takes time at most $2^{2^{d'l}}$. Thus the length of the formula $G_t(x_1,\dots,x_n,y)$ is bounded by $2^{2^v ucd}$, and the computation time for $t(\vec{q})$ is $\mathrm{O}\big(2^{2^{2^v ucd'}}\big)$, using a unit cost model for rational arithmetic.

5 A direct algorithm for evaluating μ-terms

Our direct approach to computing the values of μ-terms is based on a simple explicit representation of the functions defined by such terms. A *conditioned linear expression* is a pair, written $C \vdash e$, where e is a linear expression, and C is a finite set of strict and non-strict inequalities between linear expressions; i.e., each element of C has one of the forms in (1). We write $C(\vec{r})$ for the conjunction of the inequations obtained by instantiating $\vec{r}$ for $\vec{x}$ in C. Clearly, if $\vec{q}$ is a vector of rationals then it is decidable if $C(\vec{q})$ is true or false. The intended meaning of a conditioned linear expression $C \vdash e$ is that it denotes the value $e(\vec{r})$ when applied to a vector of reals $\vec{r}$ for which $C(\vec{r})$ is true, otherwise it is undefined. A basic property we exploit in the sequel is that every conditioning set $C(x_1,\dots,x_n)$ defines a convex subset $\{(r_1,\dots,r_n) \mid C(\vec{r})\}$ of $\mathbb{R}^n$.

Let $\mathscr{F}$ be a *system* (i.e., finite set) of conditioned linear expresssions in variables $x_1,\dots,x_n$. We say that $\mathscr{F}$ *represents* a function $f\colon [0,1]^n \to [0,1]$ if the following conditions hold:

1. For all $d_1,\dots,d_n \in [0,1]$, there exists a conditioned linear expression $(C \vdash e) \in \mathscr{F}$ such that $C(\vec{d})$ is true, and
2. for all $d_1,\dots,d_n \in [0,1]$, and every conditioned linear expression $(C \vdash e) \in \mathscr{F}$, if $C(\vec{d})$ is true then $e(\vec{d}) = f(\vec{d})$.

Note that, for two conditioned linear expressions $(C_1 \vdash e_1),(C_2 \vdash e_2) \in \mathscr{F}$, we do not require different conditioning sets C_1 and C_2 to be disjoint. However, e_1 and e_2 must agree on any overlap.

Obviously, the function represented by a system of conditioned linear expressions is unique, when it exists. But not every system represents a function. One could impose syntactic conditions on a system to ensure that it represents a function, but we shall not pursue this.

While conditioned linear expressions provide a syntax more directly tailored to expressing functions than general logical formulas, their expressivity in this regard coincides with rational linear arithmetic.

Proposition 5.1. *A function $f\colon [0,1]^n \to [0,1]$ is representable by a system of conditioned linear expressions if and only if its graph $\{(\vec{x},y) \in [0,1]^{n+1} \mid f(\vec{x}) = y\}$ is definable by a formula $F(x_1,\dots,x_n,y)$ in the first-order theory of rational linear arithmetic. Moreover, a defining formula and a representing system of conditioned linear equations can each be computed from the other.*

We believe this result to be folklore. The proof is a straightforward application of quantifier elimination.

Combining Propositions 4.1 and 5.1 we obtain:

Corollary 5.2. *For every Łukasiewicz μ-term $t(x_1,\dots,x_n)$, the function*

$$\vec{r} \mapsto t(\vec{r})\colon [0,1]^n \to [0,1]$$

is representable by a system of conditioned linear expressions in variables $x_1,\dots,x_n$. Furthermore a representing system can be computed from t.

The computation of a representing system for t via quantifier elimination, provided by the proofs of Propositions 4.1 and 5.1, is indirect. The goal of this section is to present an alternative algorithm for calculating the value $t(\vec{r})$ of a μ-term at rationals $r_1,\dots,r_n \in [0,1]$, which is directly based on manipulating conditioned linear expressions. Rather than computing an entire system of conditioned linear expressions representing t, the algorithm works locally to provide a single conditioned expression that applies to the input vector $\vec{r}$.

The algorithm takes, as input, a μ-term $t(x_1,\dots,x_n)$ and a vector of rationals $(r_1,\dots,r_n) \in [0,1]^n$, and returns a conditioned linear expression $C \vdash e$, in variables $x_1,\dots,x_n$, with the following two properties.

(P1) $C(\vec{r})$ is true.

(P2) For all $s_1,\dots,s_n \in \mathbb{R}$, if $C(\vec{s})$ is true then $s_1,\dots,s_n \in [0,1]$ and $e(\vec{s}) = t(\vec{s})$.

It follows that $e(\vec{r}) = t(\vec{r})$, so e can indeed be used to compute the value $t(\vec{r})$.

5.1 The algorithm

The algorithm takes, as input, a μ-term $t(x_1,\dots,x_n)$ and a vector of rationals $(r_1,\dots,r_n) \in [0,1]^n$, and returns a conditioned linear expression $C \vdash e$, in variables $x_1,\dots,x_n$, with the properties (P1) and (P2) above. For the purposes of the correctness proof in Section 5.3, it is convenient to consider the running of the algorithm in the more general case that $r_1,\dots,r_n$ are arbitrary real numbers in $[0,1]$. This more general algorithm can be understood as an algorithm in the Real RAM (a.k.a. BSS) model of computation [3]. When the input vector is rational, all real numbers encountered during execution of the algorithm are themselves rational, and so the general Real RAM algorithm specialises to a *bona fide* (Turing Machine) algorithm in this case. Moreover, even in the case of irrational inputs, all linear expressions constructed in the course of the algorithm are rational.

The algorithm works recursively on the structure of the term t. We present illustrative cases for terms $t_1 \oplus t_2$ and $\mu x_{n+1}.t'$. The latter is the critical case. The algorithm for $\nu x_{n+1}.t'$ is an obvious dualization.

If t is $t_1 \oplus t_2$ then recursively compute $C_1 \vdash e_1$ and $C_2 \vdash e_2$. If $e_1(\vec{r}) + e_2(\vec{r}) \leq 1$ then return

$$C_1, C_2, e_1 + e_2 \leq 1 \vdash e_1 + e_2 \ .$$

Otherwise, return

$$C_1, C_2, e_1 + e_2 \geq 1 \vdash 1 \ .$$

In the case that t is $\mu x_{n+1}.t'$, enter the following loop starting with $D = \emptyset$ and $d = 0$.

Loop: At the entry of the loop we have a finite set D of inequalities between linear expressions in $x_1,\dots,x_n$, and we have a linear expression $d(x_1,\dots,x_n)$. The loop invariant that applies is:

(I1) $D(\vec{r})$ is true; and

(I2) for all $\vec{s} \in [0,1]^n$, if $D(\vec{s})$ then $d(\vec{s}) \leq (\mu x_{n+1}.t')(\vec{s})$.

We think of D as constraints propagated from earlier iterations of the loop, and of d as the current approximation to the least fixed point subject to the constraints.

Recursively compute $t'(x_1,\dots,x_{n+1})$ at $(\vec{r}, d(\vec{r}))$ as $C \vdash e$, where e has the form:

$$q_1 x_1 + \dots + q_n x_n + q_{n+1} x_{n+1} + q \ . \qquad (2)$$

In the case that $q_{n+1} \neq 1$, define the linear expression:

$$f := \frac{1}{1-q_{n+1}} \left(q_1 x_1 + \cdots + q_n x_n + q \right) \ . \tag{3}$$

Test if $C(\vec{r}, f(\vec{r}))$ is true. If it is, exit the loop and return:

$$D \cup C(x_1,\ldots,x_n,d(x_1,\ldots,x_n)) \cup C(x_1,\ldots,x_n, f(x_1,\ldots,x_n)) \vdash f \tag{4}$$

as the result of the algorithm for $\mu x.t'$ at $\vec{r}$. Otherwise, if $C(\vec{r}, f(\vec{r}))$ is false, define $N(x_1,\ldots,x_n)$ to be the negation of the inequality $e_1(x_1,\ldots,x_n, f(x_1,\ldots x_n)) \lhd e_2(x_1,\ldots,x_n, f(x_1,\ldots x_n))$ (using $\lhd$ to stand for either $<$ or $\leq$), where $e_1(x_1,\ldots,x_{n+1}) \lhd e_2(x_1,\ldots,x_{n+1})$ is a chosen inequality in C for which $e_1(\vec{r}, f(\vec{r})) \lhd e_2(\vec{r}, f(\vec{r}))$ is false, and go to **find next approximation** below.

In the case that $q_{n+1} = 1$, test the equality $q_1 r_1 + \cdots + q_n r_n + q = 0$. If true, exit the loop with result:

$$D \cup C(x_1,\ldots,x_n,d(x_1,\ldots,x_n)) \cup \{q_1 x_1 + \cdots + q_n x_n + q = 0\} \vdash d \ . \tag{5}$$

If instead $q_1 r_1 + \cdots + q_n r_n + q \neq 0$, choose $N(x_1,\ldots,x_n)$ to be whichever of the inequalities

$$q_1 x_1 + \cdots + q_n x_n + q \ < \ 0 \qquad 0 \ < \ q_1 x_1 + \cdots + q_n x_n + q$$

is true for $\vec{r}$, and proceed with **find next approximation** below.

Find next approximation: Arrange the inequalities in C so they have the following structure.

$$C' \cup \{x_{n+1} > a_i\}_{1\leq i \leq l'} \cup \{x_{n+1} \geq a_i\}_{l'<i\leq l} \cup \{x_{n+1} \leq b_i\}_{1\leq i\leq m'} \cup \{x_{n+1} < b_i\}_{m'<i\leq m} \tag{6}$$

such that the only variables in the inequalities C', and linear expressions a_i, b_i are $x_1,\ldots,x_n$. Choose j with $1 \leq j \leq m$ such that $b_j(\vec{r}) \leq b_i(\vec{r})$ for all i with $1 \leq i \leq m$. Then go back to **loop**, taking

$$D \cup C(x_1,\ldots,x_n,d(x_1,\ldots,x_n)) \cup \{N(x_1,\ldots,x_n)\} \cup \{b_j \leq b_i \mid 1 \leq i \leq m\} \qquad e(\vec{x}, b_j(\vec{x})) \tag{7}$$

to replace D and d respectively.

5.2 A simple example

Consider the Łμ term $t = \mu x.(\mathbb{P}_{\geq\frac{1}{2}} x \sqcup \frac{1}{2})$, where $\mathbb{P}_{\geq\frac{1}{2}} x$ is the macro formula as in Definition 3.1, that is $\mathbb{P}_{\geq\frac{1}{2}} x = \mathbb{P}_{=1}(x \oplus \frac{1}{2}) = \nu y.(y \odot (x \oplus \frac{1}{2}))$. Thus,

$$t = \mu x.\Big(\nu y.\big(y \odot (x \oplus \frac{1}{2})\big) \sqcup \frac{1}{2} \Big)$$

Here, $t'(x) = \nu y.\big(y \odot (x \oplus \frac{1}{2})\big) \sqcup \frac{1}{2}$ is a discontinuous function, and the value of t is 1.

We omit giving a detailed simulation of the algorithm on the subexpression $t'(x)$ at $x = r$. The result it produces, however, is $\{0 \leq x < \frac{1}{2}\} \vdash \frac{1}{2}$ if $r < \frac{1}{2}$, and $\{\frac{1}{2} \leq x \leq 1\} \vdash 1$ if $r \geq \frac{1}{2}$.

We run the algorithm on input $\mu x.t'(x)$. Set $D = \emptyset$ and $d = 0$. Calculating $t'(x)$ at $x = 0$ we obtain $C \vdash e$ as $\{0 \leq x < \frac{1}{2}\} \vdash \frac{1}{2}$. We now need to calculate $f := \frac{1}{1-0}(\frac{1}{2}) = \frac{1}{2}$. The constraint $C(\frac{1}{2})$ does not hold. Thus we need to improve the approximation $d = 0$. Since $e = \frac{1}{2}$ is constant, the next approximation is $\frac{1}{2}$. The new set of constraints is still the emptyset. Thus we iterate the algorithm with $D = \emptyset$ and $d = \frac{1}{2}$. Calculating $t'(x)$ at $x = \frac{1}{2}$ produces $C \vdash e$ as $\{\frac{1}{2} \leq x \leq 1\} \vdash 1$. Compute $f := \frac{1}{1-0}(1) = 1$. Since $C(1)$ holds, the algorithm terminates with $\emptyset \vdash 1$, as desired.

5.3 Correctness of the algorithm

Theorem 5.3. *Let $t(x_1,\dots,x_n)$ be any Łukasiewicz μ-term. Then, for every input vector $(r_1,\dots,r_n) \in [0,1]^n$, the above (Real RAM) algorithm terminates with a conditioned linear expression $C_{\vec{r}} \vdash e_{\vec{r}}$ satisfying properties (P1) and (P2). Moreover, the set of all possible resulting conditioned linear expressions*

$$\{C_{\vec{r}} \vdash e_{\vec{r}} \mid \vec{r} \in [0,1]^n\} \tag{8}$$

is finite, and thus provides a representing system for the function $t\colon [0,1]^n \to [0,1]$.

Before the proof it is convenient to introduce some terminology associated with the properties stated in the theorem. For a μ-term t, we call the cardinality of the set (8) of possible results, $C_{\vec{r}} \vdash e_{\vec{r}}$, the *basis size*, and we call the maximum number of inequalities in any $C_{\vec{r}}$ the *condition size*.

Proof. By induction on the structure of t. We verify the critical case when t is $\mu x_{n+1}.t'$.

We show first that the loop invariants (I1), (I2) guarantee that any result returned via (4) or (5) satisfies (P1) and (P2). By induction hypothesis, the recursive computation of $t'(x_1,\dots,x_{n+1})$ at $(\vec{r},d(\vec{r}))$ as $C \vdash e$, where e has the form $q_1 x_1 + \dots + q_n x_n + q_{n+1} x_{n+1} + q$ as in (2), satisfies: $C(\vec{r},d(\vec{r}))$; and, for all $s_1,\dots,s_{n+1} \in \mathbb{R}$, if $C(s_1,\dots,s_{n+1})$ then $\vec{s} \in [0,1]^n$ and $t'(s_1,\dots,s_{n+1}) = e(s_1,\dots,s_{n+1})$.

In the case that $q_{n+1} \neq 1$, the linear expression f, defined in (3), maps any $s_1,\dots,s_n \in \mathbb{R}$ to the unique solution $f(\vec{s})$ to the equation $x_{n+1} = e(s_1,\dots,s_n,x_{n+1})$ in $\mathbb{R}$. Suppose that $D(\vec{s})$ holds. Then, by loop invariant (I2), $d(\vec{s}) \leq (\mu x_{n+1}.t')(\vec{s})$. Suppose also that $C(\vec{s}, f(\vec{s}))$. Then $t'(\vec{s}, f(\vec{s})) = e(\vec{s}, f(\vec{s})) = f(\vec{s})$, i.e., $f(\vec{s})$ is a fixed point of $x_{n+1} \mapsto t'(\vec{s},x_{n+1})$; whence, $(\mu x_{n+1}.t')(\vec{s}) \leq f(\vec{s})$. Suppose, finally, that $C(\vec{s},d(\vec{s}))$ also holds. Then, because both $C(\vec{s},d(\vec{s}))$ and $C(\vec{s}, f(\vec{s}))$, and $d(\vec{s}) \leq (\mu x_{n+1}.t')(\vec{s}) \leq f(\vec{s})$, we have, by the convexity of constraints, that $t'(\vec{s},s_{n+1}) = e(\vec{s},s_{n+1})$ for all $s_{n+1} \in [d(\vec{s}), f(\vec{s})]$. So $f(\vec{s})$ is the unique fixed-point of $x_{n+1} \mapsto t'(\vec{s},x_{n+1})$ on $[d(\vec{s}), f(\vec{s})]$. Since, $d(\vec{s}) \leq (\mu x_{n+1}.t')(\vec{s})$, we have $f(\vec{s}) = (\mu x_{n+1}.t')(\vec{s})$. This argument justifies that the conditioned linear expression of (4) satisfies (P2). It satisfies (P1) just if $C(\vec{r}, f(\vec{r}))$, which is exactly the condition under which (4) is returned as the result.

In the case that $q_{n+1} = 1$ then, for any $s_1,\dots,s_n \in \mathbb{R}$, the equation $x_{n+1} = e(s_1,\dots,s_n,x_{n+1})$ has a solution if and only if $q_1 s_1 + \dots + q_n s_n + q = 0$, in which case any $x_{n+1} \in \mathbb{R}$ is a solution. Suppose that $q_1 s_1 + \dots + q_n s_n + q = 0$ and $C(\vec{s},d(\vec{s}))$ both hold. Then $t'(s_1,\dots,s_n,d(\vec{s})) = e(\vec{s},d(\vec{s})) = d(\vec{s})$, so $d(\vec{x})$ is a fixed point of $x_{n+1} \mapsto t'(\vec{s},x_{n+1})$. If also $D(\vec{s})$ holds then, by loop invariant (I2), $d(\vec{x}) = (\mu x_{n+1}.t')(\vec{s})$. We have justified that the conditioned linear expression of (5) satisfies (P2). It satisfies (P1) just if $q_1 r_1 + \dots + q_n r_n + q = 0$, which is exactly the condition under which (5) is returned as the result.

Next we show that the loop invariants are preserved through the computation. Properties (I1) and (I2) are trivially satisfied by the initial values $D = \emptyset$ and $d = 0$. We must show that they are preserved when D and d are modified via (7), which happens when execution passes to **find next approximation**. In this subroutine, the inequalities in C are first arranged as in (6) where, as $C(\vec{r},d(\vec{r}))$, we must have $m \geq 1$, as otherwise $C(\vec{r},s)$ would hold for all real $s \geq d(\vec{r})$, contradicting that $C(\vec{r},s)$ implies $s \in [0,1]$. (Similarly, $l \geq 1$.) Thus there indeed exists j with $1 \leq j \leq m$ such that $b_j(\vec{r}) \leq b_i(\vec{r})$ for all i with $1 \leq i \leq m$. It is immediate that the constraints in the modified D of (7) are true for $\vec{r}$. Thus (I1) is preserved. To show (I2), suppose $s_1,\dots,s_n$ satisfy the constraints, i.e.,

$$D(\vec{s}) \qquad C(\vec{s},d(\vec{s})) \qquad N(\vec{s}) \qquad \{b_j(\vec{s}) \leq b_i(\vec{s}) \mid 1 \leq i \leq m\} \ .$$

Defining $r' = (\mu x_{n+1}.t')(\vec{s})$, by (I2) for D,d we have $d(\vec{s}) \leq r'$. We must show that $e(\vec{s},b_j(\vec{s})) \leq r'$. By the definition of $N(x_1,\dots,x_n)$, in either the $q_{n+1} \neq 1$ or $q_{n+1} = 1$ case, $N(\vec{s})$ implies that $C(\vec{s}, r')$ does not hold. Because $C(\vec{s},d(\vec{s}))$ and by the choice of j, it holds that $C(\vec{s},s)$, for all $s \in [0,1]$ such that $s = d(\vec{s})$ or

$d(\vec{s}) < s < b_j(\vec{s})$. Since $C(\vec{s}, r')$ is false and $d(\vec{s}) \leq r'$, it follows from the convexity of the conditioning set C that, for every s with $s = d(\vec{s})$ or $d(\vec{s}) < s < b_j(\vec{s})$, we have $s < r'$. Whence, since r' is the least prefixed point for $x_{n+1} \mapsto t'(\vec{s}, x_{n+1})$, also $s < t'(\vec{s}, s) \leq r'$, i.e.,

$$s < e(\vec{s}, s) \leq r' \quad . \tag{9}$$

Thus, $e(\vec{s}, b_j(\vec{s})) = \sup\{e(\vec{s}, s) \mid s = d(\vec{s}) \text{ or } d(\vec{s}) \leq s < b_j(\vec{s})\} \leq r'$. Thus, $e(\vec{s}, b_j(\vec{s})) \leq r'$, i.e., it is an approximation to the fixed point. Moreover, it is a good new approximation to choose in the sense that:

$$d(\vec{s}) < e(\vec{s}, b_j(\vec{s})) \text{ and } \text{not}\, C(\vec{s}, e(\vec{s}, b_j(\vec{s}))) \quad . \tag{10}$$

The former holds because $d(\vec{s}) < e(\vec{s}, d(\vec{s}))$, by (9), and $d(\vec{s}) \leq b_j(\vec{s})$. The latter because if $C(\vec{s}, e(\vec{s}, b_j(\vec{s})))$ then, in particular, $e(\vec{s}, b_j(\vec{s})) \leq b_j(\vec{s})$, so $b_j(\vec{s}) = e(\vec{s}, b_j(\vec{s})) = r'$, contradicting that not $C(\vec{s}, r')$.

To show termination, by induction hypothesis, collecting all possible results of running the algorithm on t' produces a representing system for $t' \colon [0,1]^{n+1} \to [0,1]$:

$$C_1 \vdash e_1 \quad \ldots \quad C_{k'} \vdash e_{k'} \quad , \tag{11}$$

where k' is the basis size of t'. We now analyse the execution of the algorithm for $\mu x_{n+1}.t'$ on a given input vector $(r_1, \ldots, r_n)$. On iteration number i, the loop is entered with constraints D_i and approximation d_i (where $D_1 = \emptyset$ and $d_1 = 0$), after which the recursive call to the algorithm for t' yields one of the conditioned linear expressions, $C_{k_i} \vdash e_{k_i}$, from (11) above, such that $C_{k_i}(\vec{r}, d_i(\vec{r}))$ holds. Then, depending on conditions involving only $C_{k_i} \vdash e_{k_i}$ and $\vec{r}$, either a result is returned, or D_{i+1} and d_{i+1} are constructed for the loop to be repeated. By (10), at iteration $i+1$ of the loop, we have $d_{i+1}(\vec{r}) > d_i(\vec{r})$ and also $C_{k_i}(\vec{r}, d_{i+1}(\vec{r}))$ is false. Since each conditioning set is convex, it follows that no C_j can occur twice in the list $C_{k_1}, C_{k_2}, \ldots$. Hence the algorithm must exit the loop after at most k' iterations. Therefore, the computation for $\mu x.t'$ at $\vec{r}$ terminates.

It remains to show that the algorithm for $\mu x.t'$ produces only finitely many conditioned linear expressions $C_{\vec{r}} \vdash e_{\vec{r}}$. The crucial observation is that the vector $\vec{r}$ is used only to determine the control flow of the algorithm, i.e., which branches of conditional statements are followed, the choices made in selecting N and b_j in (7), and the order in which the different $C_j \vdash e_j$, from (11) are visited (given by the sequence $k_1, k_2, \ldots$ of values taken by j). Using this, if l' is the condition size of t', then a loose upper bound is that the number of possible results $C_{\vec{r}} \vdash e_{\vec{r}}$ for the algorithm for $\mu x_{n+1}.t'$ is at most $(k'(l')^2)^{k'}$, and the number of inequalities in $C_{\vec{r}}$ is at most $2k'l'$. □

The above proof gives a truly abysmal complexity bound for the algorithm. Let the basis and condition size for the term $t'(x_1, \ldots, x_{n+1})$ be k' and l' respectively. Then, as in the proof, the basis and condition size for $\mu x_{n+1}.t'$ are respectively bounded by:

$$k \leq (k'(l')^2)^{k'} \text{ and } l \leq 2k'l' \quad .$$

Using these bounds, the basis and condition size have non-elementary growth in the number of fixed points in a term t.

5.4 Comparison

According to the crude complexity analyses we have given, the evaluation of Łukasiewicz μ-terms via rational linear arithmetic is (in having doubly- and triply-exponential space and time complexity bounds) preferable to the (non-elementary space and hence time) evaluation via the direct algorithm. Nevertheless, we expect the direct algorithm to work better than this in practice. Indeed, a main motivating factor in the design of the direct algorithm is that the algorithm for $\mu x_{n+1}.t'$ only explores as much of the basis set for t' as it needs to, and does so in an order that is tightly constrained by the monotone improvements made to the approximating d expressions along the way. In contrast, the crude complexity analysis is based on a worst-case scenario in which the algorithm is assumed to visit the entire basis for t', and, moreover, to do so, for different input vectors $\vec{r}$, in every possible order for visiting the different basis sets. Perhaps better bounds can be obtained by a more careful analysis of the algorithm.

6 Model checking

Let ϕ be a closed Łμ formula and $(S, \rightarrow)$ a finite rational PNTS. We wish to compute the value $[\![\phi]\!](s)$ at any given state $s \in S$. We do this by effectively producing a closed μ-term $t_s(\phi)$, with the property that $t_s(\phi) = [\![\phi]\!](s)$, whence the rational value of $[\![\phi]\!](s)$ can be calculated by the algorithm in Section 5.

We assume, without loss of generality, that all fixed-point operators in ϕ bind distinct variables. Let $X_1, \dots, X_m$ be the variables appearing in ϕ. We write $\sigma_i X_i.\, \psi_i$ for the unique subformula of ϕ in which X_i is bound. The strict (i.e., irreflexive) *domination* relation $X_i \rhd X_j$ between variables is defined to mean that $\sigma_j X_j.\, \psi_j$ occurs as a subformula in ψ_i.

Suppose $|S| = n$. For each $s \in S$, we translate ϕ to a μ-term $t_s(\phi)$ containing at most mn variables $x_{i,s'}$, where $1 \leq i \leq m$ and $s' \in S$. The translation is defined using a more general function t_s^Γ, defined on subformulas of ϕ, where $\Gamma \subseteq \{1, \dots, m\} \times S$ is an auxiliary component keeping track of the states at which variables have previously been encountered. Given Γ and $(i,s) \in \{1, \dots, m\} \times S$, we define:

$$\Gamma \rhd (i,s) \;=\; (\Gamma \cup \{(i,s)\}) \backslash \{(j,s') \in \Gamma \mid X_i \rhd X_j\} \;\; .$$

This operation is used in the definition below to ‘reset’ subordinate fixed-point variables whenever a new variable that dominates them is declared.

$$\begin{aligned}
t_s^\Gamma(X_i) &= \begin{cases} x_{i,s} & \text{if } (i,s) \in \Gamma \\ \sigma_i x_{i,s}.\, t_s^{\Gamma \rhd (i,s)}(\psi_i) & \text{otherwise} \end{cases} \\
t_s^\Gamma(P) &= \underline{\rho(P)(s)} \\
t_s^\Gamma(\overline{P}) &= \underline{1 - \rho(P)(s)} \\
t_s^\Gamma(q\phi) &= q t_s^\Gamma(\phi) \\
t_s^\Gamma(\phi_1 \bullet \phi_2) &= t_s^\Gamma(\phi_1) \bullet t_s^\Gamma(\phi_2) \qquad \bullet \in \{\sqcup, \sqcap, \oplus, \odot\} \\
t_s^\Gamma(\Diamond\phi) &= \bigsqcup_{s \rightarrow d} \bigoplus_{s' \in S} d(s')\, t_{s'}^\Gamma(\phi) \\
t_s^\Gamma(\Box\phi) &= \bigsqcap_{s \rightarrow d} \bigoplus_{s' \in S} d(s')\, t_{s'}^\Gamma(\phi) \\
t_s^\Gamma(\sigma_i X_i.\, \psi_i) &= \sigma_i x_{i,s}.\, t_s^{\Gamma \cup \{(i,s)\}}(\psi_i)
\end{aligned}$$

This is well defined because changing from Γ to $\Gamma \rhd (i,s)$ or to $\Gamma \cup \{(i,s)\}$ strictly increases the function

$$i \mapsto |\{(i,s) \mid (i,s) \in \Gamma\}| \colon \{1, \dots, m\} \rightarrow \{0, \dots, n\}$$

under the lexicographic order on functions relative to $\rhd$.

Proposition 6.1. *For any closed Łμ formula ϕ, finite PNTS $(S,\rightarrow)$ and $s \in S$, it holds that $[\![\phi]\!](s) = t_s^{\emptyset}(\phi)$.*

We omit the laborious proof. It is reminiscent of the reduction of modal μ-calculus model checking to a system of nested boolean fixed-point equations in Section 4 of [17].

7 Related and future work

The first encodings of probabilistic temporal logics in a probabilistic version of the modal μ-calculus were given in [4], where a version **PCTL***, tailored to processes exhibiting probabilistic but not nondeterministic choice, was translated into a non-quantitative probabilisitic variant of the μ-calculus, which included explicit (probabilistic) path quantifiers but disallowed fixed-point alternation.

In their original paper on quantitative μ-calculi [12], Huth and Kwiatkowska attempted a model checking algorithm for alternation-free formulas in the version of Łμ with $\oplus$ and $\odot$ but without $\sqcap$, $\sqcup$ and scalar multiplication. Subsequently, several authors have addressed the problem of computing (sometimes approximating) fixed points for monotone functions combining linear (sometimes polynomial) expressions with min and max operations; see [10] for a summary. However, such work has focused on (efficiently) finding outermost (simultaneous) fixed-points for systems of equations whose underlying monotone functions are continuous. The nested fixed points considered in the present paper give rise to the complication of non-continuous functions, as the example of Section 5.2 demonstrates.

As future work, it is planned to run an experimental comparison of the direct algorithm against the reduction to linear arithmetic. As suggested in Section 5.4, we expect the direct algorithm to work better in practice than the non-elementary upper bound on its complexity, given by our crude analysis, suggests. Furthermore, as a natural generalization of the approximation approach to computing fixed points, the direct algorithm should be amenable to optimizations such as the simultaneous solution of adjacent fixed points of the same kind, and the reuse of previous approximations when applicable due to monotonicity considerations. Unlike the black-box reduction to linear arithmetic, based on quantifier elimination, the linear-constraint-based approach of the direct algorithm should also offer a flexible machinery helpful in the design of optimized procedures for calculating values of particular subclasses of Łμ-terms. An important example is given by the fragment of Łμ capable of encoding **PCTL** (see Remark 3.6).

Our results on Łμ are a contribution towards the development of a robust theory of fixed-point probabilistic logics. The simplicity of the proposed encoding of **PCTL** (see Remark 3.6 above) suggests that the direction we are following is promising. In a follow-up paper, by the first author, it will be shown that the process equivalence characterised by Łukasiewicz μ-calculus is the standard notion of *probabilistic bisimilarity* [23]. Thus the quantitative approach to probabilistic μ-calculi may be considered equally suitable as a mechanism for characterising process equivalence as the non-quantitative μ-calculi advocated for this purpose in [4] and [7].

Further research will have to explore the relations between quantitative μ-calculi such as Łμ and other established frameworks for verification and design of probabilistic systems. Important examples include the *abstract probabilistic automata* of [6], the compositional *assume-guarantee* techniques of [16, 9] and the recent *p-automata* of [13]. In particular, with respect to the latter formalism, we note that the acceptance condition of p-automata is specified in terms of stochastic games whose configurations may have preseeded threshold values whose action closely resembles that of the threshold modalities considered in this work (Definition 3.1). Exploring the relations between p-automata games and Łμ-games [19] could shed light on some underlying fundamental ideas.

Acknowledgements

We thank Kousha Etessami, Grant Passmore, Colin Stirling and the anonymous reviewers for helpful comments and for pointers to the literature.

The first author carried out this work during the tenure of an ERCIM "Alain Bensoussan" Fellowship, supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

References

[1] Christel Baier & Joost Pieter Katoen (2008): *Principles of Model Checking*. The MIT Press.

[2] Andrea Bianco & Luca de Alfaro (1995): *Model Checking of Probabilistic and Nondeterministic Systems*. In: *Foundations of Software Technology and Theoretical Computer Science*, *Lecture Notes in Computer Science* 1026, Springer-Verlag, pp. 499–513, doi:10.1007/3-540-60692-0_70.

[3] Lenore Blum, Mike Shub & Steve Smale (1989): *On a Theory of Computation and Complexity over the Real Numbers: NP-completeness, Recursive Functions and Universal Machines*. *Bulletin of the AMS* 21(1), doi:10.1109/SFCS.1988.21955.

[4] Rance Cleaveland, S. Purushothaman Iyer & Muralidhar Narasimha (1999): *Probabilistic Temporal Logics via the Modal mu-Calculus*. In: *Foundations of Software Science and Computation Structures*, doi:10.1007/3-540-49019-1_20.

[5] Luca de Alfaro & Rupak Majumdar (2004): *Quantitative Solution of omega-Regular Games*. *Journal of Computer and System Sciences, Volume 68, Issue 2*, pp. 374 – 397, doi:10.1016/j.jcss.2003.07.009.

[6] Benoit Delahaye, Joost Pieter Katoen, Kim Larsen, Axel Legay, Mikkel Pedersen, Falak Sher & Andrzej Wasowski (2011): *Abstract Probabilistic Automata*. In: *Proc. of 12th VMCAI*, doi:10.1007/978-3-642-18275-4_23.

[7] Yuxin Deng & Rob van Glabbeek (2010): *Characterising Probabilistic Processes Logically*. In: *Logic for programming, artificial intelligence and reasoning*, *Lecture Notes in Computer Science* 6397, doi:10.1007/978-3-642-16242-8_20.

[8] Jeanne Ferrante & Charles Rackoff (1975): *A Decision Procedure for the First Order Theory of Real Addition with Order*. *SIAM Journal of Computing* 4(1), pp. 69–76, doi:10.1137/0204006.

[9] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, David Parker & Hongyang Qu (2011): *Quantitative multi-Objective Verification for Probabilistic Systems*. In: *Proc. of 14th TACAS*, doi:10.1007/978-3-642-19835-9_11.

[10] Thomas Martin Galwitza & Helmut Seidl (2011): *Solving Systems of Rational Equations through Strategy Iteration*. *ACM Trabnsactions on Programming Languages and Systems* 33(3), doi:10.1145/1961204.1961207.

[11] Petr Hájek (2001): *Metamathematics of Fuzzy Logic*. Springer.

[12] Michael Huth & Marta Kwiatkowska (1997): *Quantitative Analysis and Model Checking*. In: *Proceeding of the 12th Annual IEEE Symposium on Logic in Computer Science*.

[13] Michael Huth, Nir Piterman & Daniel Wagner (2012): *p-Automata: New Foundations for discrete-time Probabilistic Verification*. *Perform. Eval.* 69(7-8), doi:10.1016/j.peva.2012.05.005.

[14] David Janin & Igor Walukiewicz (1996): *On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic*. *Lecture Notes in Computer Science* 1119, pp. 263–277, doi:10.1007/3-540-61604-7_60.

[15] D. Kozen (1983): *Results on the Propositional mu-Calculus*. In: *Theoretical Computer Science*, pp. 333–354, doi:10.1016/0304-3975(82)90125-6.

[16] Marta Kwiatkowska, Gethin Norman, David Parker & Hongyang Qu (2010): *Assume-Guarantee Verification for Probabilistic Systems*. In: *Proceedings of 16th TACAS*, doi:10.1007/978-3-642-12002-2_3.

[17] Angelika Mader (1995): *Modal μ-Calculus, Model Checking and Gauß Elimination*. In: *Tools and Algorithms for the Construction and Analysis of Systems (TACAS), LNCS* 1019, pp. 72–88, doi:10.1007/3-540-60630-0_4.

[18] Annabelle McIver & Carroll Morgan (2007): *Results on the Quantitative μ-Calculus qMμ*. *ACM Transactions on Computational Logic* 8(1), doi:10.1145/1182613.1182616.

[19] Matteo Mio (2012): *Game Semantics for Probabilistic μ-Calculi*. Ph.D. thesis, School of Informatics, University of Edinburgh. Permanent URL: http://hdl.handle.net/1842/6223.

[20] Matteo Mio (2012): *On The Equivalence of Denotational and Game Semantics for the Probabilistic μ-Calculus*. *Logical Methods in Computer Science* 8(2), doi:10.2168/LMCS-8(2:7)2012.

[21] Matteo Mio (2012): *Probabilistic Modal μ-Calculus with Independent Product*. *Logical Methods in Computer Science* 8(4), doi:10.2168/LMCS-8(4:18)2012.

[22] Carroll Morgan & Annabelle McIver (1997): *A Probabilistic Temporal Calculus Based on Expectations*. In: *In Lindsay Groves and Steve Reeves, editors, Proc. Formal Methods*, Springer Verlag.

[23] Roberto Segala (1995): *Modeling and Verification of Randomized Distributed Real-Time Systems*. Ph.D. thesis, Laboratory for Computer Science, M.I.T.

[24] Colin Stirling (2001): *Modal and Temporal Logics for Processes*. Springer, doi:10.1007/3-540-60915-6_5.

A Appendix: some omitted proof details

We add detail to the outlined proof of Theorem 3.7, by supplying the omited argument for the equality

$$\bigsqcup_{\sigma}\{m^s_\sigma(\Psi)\} = [\![\mu X.F(X))]\!]_\rho(s) \quad ,$$

which appears as case 11. Although game semantics provides the most intuitive justification, we instead give a direct denotational proof, in order to avoid introducing game-theoretic machinery.

Expanded proof of Theorem 3.7. Case 11 ($\leq$). We first show that

$$\bigsqcup_{\sigma}\{m^s_\sigma(\Psi)\} \leq [\![\mu X.F(X))]\!]_\rho(s) \tag{12}$$

Define $\Psi_k = \{s_0.s_1.s_2 \dots \mid s_0 = s \text{ and } \exists n \leq k.\big(s_n \in (\!|\phi_2|\!)_\rho \wedge \forall m < n.(s_m \in (\!|\phi_1|\!)_\rho)\big)\}$. Clearly $\Psi = \bigcup_k \Psi_k$. Suppose Inequality 12 does not hold. Then there exists some k and scheduler σ such that

$$m^s_\sigma(\Psi_k) > [\![\mu X.F(X)]\!]_\rho(s) \tag{13}$$

We prove that this is not possible by induction on k. In the $k=0$ case, since we are assuming $m^s_\sigma(\Psi_0) > 0$, it holds that $s \in (\!|\phi_2|\!)_\rho$. By inductive hypothesis on ϕ_2, we know that $[\![\mathbf{E}(\phi_2)]\!](s) = 1$ and this implies that $\mu X.F(X) = 1$, which is a contradiction with the assumed strict inequality 13. Consider the case $k+1$. Note that if $s \in (\!|\phi_2|\!)_\rho$ then, $[\![\mu X.F(X)]\!]_\rho(s) = 1$ as before, contradicting Inequality 13. So assume $s \notin (\!|\phi_2|\!)_\rho$. Since we are assuming $m^s_\sigma(\Psi_{k+1}) > 0$ it must be the case that $s \in (\!|\phi_1|\!)_\rho$. Similarly, $m^s_\sigma(\Psi_{k+1}) > 0$ and $s \notin (\!|\phi_2|\!)_\rho$ imply that $s \not\to$ does not hold. This means (see Definition 2.8) that $\sigma(\{s\})$ is defined. Let $d = \sigma(\{s\})$ and observe that $m^s_\sigma(\Psi_{k+1}) = \sum_{t \in S} d(t) m^t_{\sigma'}(\Psi_k)$, where $\sigma'(s_0, s_1, \dots, s_n) = \sigma(s, s_0, s_1, \dots, s_n)$.

By induction on k we know that the inequality $m^t_{\sigma'}(\Psi_k) \leq [\![\mu X.F(X)]\!]_\rho(t)$ holds for every $t \in S$. Thus, by definition of the semantics of $\Diamond$, we obtain $m^s_\sigma(\Psi_k) \leq [\![\Diamond\big(\mu X.F(X)\big)]\!]_\rho$. Recall that we previously assumed $s \not\in (\!|\, \phi_2 \,|\!)_\rho$ and $s \in (\!|\, \phi_1 \,|\!)_\rho$. Hence the equality

$$[\![\Diamond\big(\mu X.F(X)\big)]\!]_\rho(s) = [\![\mathbf{E}(\phi_2) \sqcup (\mathbf{E}(\phi_1) \sqcap \big(\Diamond \mu X.F(X)\big))]\!]_\rho(s)$$

holds. The formula on the right is just the unfolding $F(\mu X.F(X))$ of $\mu X.F(X)$. This implies the desired contradiction.

Case $11(\geq)$. We now prove that also the inequality

$$\bigsqcup_\sigma \{m^s_\sigma(\psi)\} \geq [\![\mu X.F(X)]\!]_\rho(s) \tag{14}$$

holds. By Knaster-Tarski theorem, $[\![\mu X.F(X)]\!]_\rho = \bigsqcup_\alpha [\![F(X)]\!]^\alpha_\rho$, where α ranges over the ordinals and $[\![F(X)]\!]^\alpha_{\rho^\alpha}$ with $\rho^\alpha = \rho[\bigsqcup_{\beta<\alpha} [\![F(X)]\!]_{\rho^\beta}/X]$. We prove Inequality 14 by showing, by transfinite induction, that for every ordinal α and $\varepsilon > 0$, the inequality

$$\bigsqcup_\sigma \{m^s_\sigma(\psi)\} > [\![\mu X.F(X)]\!]_{\rho^\alpha}(s) - \varepsilon \tag{15}$$

holds, for all $s \in S$. The case for $\alpha = 0$ is immediate since $[\![F]\!]_{\rho^0}(s) > 0$ if and only if $[\![\mathbf{E}(\phi_2)]\!]_\rho(s) = 1$ and this implies $\bigsqcup_\sigma \{m^s_\sigma(\psi)\} = 1$. Consider $\alpha = \beta + 1$. If $[\![\mathbf{E}(\phi_2)]\!]_\rho(s) = 1$ then Inequality 14 holds as above. Thus assume $[\![\phi_2]\!]_\rho(s) = 0$. Note that $[\![F]\!]_{\rho^0}(s) > 0$ only if $s \in [\![\mathbf{E}(\phi_1)]\!]$. Thus assume $[\![\mathbf{E}(\phi_1)]\!]^\beta_\rho(s) = 1$. Under these assumption, $[\![F(X)]\!]_{\rho^\alpha} = [\![\Diamond F(X)]\!]_{\rho^\beta}$ as it is immediate to verify. By definition of the semantics of $\Diamond$ we have:

$$[\![\Diamond F(X)]\!]_{\rho^\beta}(s) = \bigsqcup_{s \to d} \big(\sum_{t \in S} d(t) [\![F(X)]\!]_{\rho^\beta}(t)\big)$$

By induction hypothesis on β we know that for every ε,

$$[\![\Diamond F(X)]\!]_{\rho^\beta}(s) < \bigsqcup_{s \to d} \big(\sum_{t \in S} d(t) \Big(\bigsqcup_\sigma \{m^t_\sigma(\psi)\} + \varepsilon\Big)\big)$$

For each $s \to d$ and σ define σ^d as $\sigma^d(\{s\}) = d$ and $\sigma^d(s.t_0.\dots) = \sigma(t_0\dots)$. A simple argument shows that

$$\bigsqcup_{s \to d} \big(\sum_{t \in S} d(t) \Big(\bigsqcup_\sigma \{m^t_\sigma(\psi)\} + \varepsilon\Big)\big) = \bigsqcup_{\sigma^d} \{m^s_{\sigma^d}(\psi)\} + \varepsilon$$

and this conclude the proof for the case $\alpha = \beta + 1$. Lastly, the case for α a limit ordinal follows straightforwardly from the inductive hypothesis on $\beta < \alpha$. □

Proof of Proposition 5.1. Suppose we have a system of k conditioned linear expressions representing f. Each conditioned expression $C \vdash e$ is captured by the implication $(\bigwedge C) \to y = e$, so the whole system translates into a conjunction of k such implications. To this conjunction, one need only add the range constraints $0 \leq z$ and $z \leq 1$ for each variable z, as further conjuncts. In this way, the graph is easily expressed as a quantifier free formula. (Since the implications are equivalent to disjunctions of atomic formulas, the resulting formula is naturally in conjunctive normal form.)

Conversely, suppose $F(x_1, \dots, x_n, y)$ defines the graph of f. By quantifier elimination, we can assume that F is quantifier free and in disjunctive normal form. Then F is a disjunction of conjunctions, where each conjunction, K, can be easily rewritten in the form

$$\Big(\bigwedge C\Big) \wedge \Bigg(\bigwedge_{1 \leq i \leq h} y > a_i\Bigg) \wedge \Bigg(\bigwedge_{1 \leq i \leq k} y \geq b_i\Bigg) \wedge \Bigg(\bigwedge_{1 \leq i \leq l} y \leq c_i\Bigg) \wedge \Bigg(\bigwedge_{1 \leq i \leq m} y < d_i\Bigg) \ , \tag{16}$$

such that the only variables in the finite set of atomic formulas C, and linear expressions a_i, b_i, c_i, d_i are $x_1, \ldots, x_n$. Since F is the graph of a function, for all reals $r_1, \ldots, r_n$, there is at most one s such that $K(\vec{r}, s)$ holds, and, if it does, then all of $r_1, \ldots, r_n, s$ are in $[0,1]$. Given such an s, we therefore have:

$$\max\{a_i(\vec{r}) \mid 1 \le i \le h\} < \max\{b_i(\vec{r}) \mid 1 \le i \le k\} = s = \min\{c_i(\vec{r}) \mid 1 \le i \le l\} < \min\{d_i(\vec{r}) \mid 1 \le i \le m\} \quad .$$

A system of conditioned linear expressions for f is thus obtained as follows. For each conjunct K in F, written in the form of (16) above, and each j with $1 \le j \le k$, include the conditioned linear expression:

$$C, \{b_j > a_i\}_{1 \le i \le h}, \{b_j \ge b_i\}_{1 \le i \le k}, \{b_j \le c_i\}_{1 \le i \le l}, \{b_j < d_i\}_{1 \le i \le m}, \vdash b_j \quad .$$

□

We supplement the proof of Theorem 5.3 with more detail on the bounds on basis and condition size.

Expanded proof of Theorem 5.3. We analyse the control flow in the algorithm for $\mu x_{n+1}.t'$ on a given input vector $(r_1, \ldots, r_n)$. On iteration number i, the loop is entered with constraints D_i and approximation d_i, after which the recursive call to the algorithm for t' yields one of the conditioned linear expressions, $C_{k_i} \vdash e_{k_i}$. Suppose that C_{k_i} and D_i contain u and v inequalities respectively. If the loop is exited producing (4) as result then the resulting $C_{\vec{r}}$ has $2u+v$ inequalities. If it is exited producing (5) as result then $C_{\vec{r}}$ has $u+v+2$ inequalities (where $u+v+2 \le 2u+v$ because C_{k_i} has to enforce the range constraint $0 \le x_{n+1} \le 1$). Otherwise, the algorithm repeats the loop, entering iteration $i+1$ with D_{i+1}, given by (7), having at most $2u+v$ inequalities (N contributes 1 inequality, and there are at most $u-1$ inequalities $b_j \le b_i$ in (7) since $l \ge 1$).

Therefore, if l' is now maximum number of inequalities occurring in any C_j from (11) (i.e., if it is the condition size for t') the algorithm for $\mu x_{n+1}.t'$ at $\vec{r}$, which runs for at most k' iterations, results in $C_{\vec{r}}$ containing at most $2k'l'$ inequalities.

To bound the number of results $C_{\vec{r}} \vdash e_{\vec{r}}$, we count the possible control flows of the algorithm. At iteration i, the algorithm uses $C_{k_i} \vdash e_{k_i}$ from (11), using which it might terminate with either (4) or (5), or it might repeat the loop, entering iteration $i+1$ with D_{i+1}, given by (7), which can arise from C_{k_i} in a number of ways determined by the possible pairs of choices for N and b_j in (7). In the case that the variable vector $(x_1, \ldots, x_n)$ is empty (i.e., the term $\mu x_{n+1}.t'$ is closed) the constraints in D are redundant (they are simply true inequalities between rathionals) and so can be discarded. In the case that $n \ge 1$, there are at least 2 inequalities in C giving range constraints on x_1, so there are at most l' choices for N ($l'-2$ choices in the case that $q_{n+1} \ne 1$, and 2 in the case $q_{n+1} = 1$). Irrespective of n, there are at most $l'-1$ choices for b_j (taking n into account this can be improved to $l'-2n-1$). Therefore, the execution of the algorithm, is determined by the sequence:

$$k_1, u_1, k_2, u_2, \ldots, k_m, v$$

where: $m \le k'$ is the number of loop iterations performed; each u_i, where $1 \le u_i \le l'(l'-1)$, represents the choice of N and b_j used in the construction of D_{i+1} (7), and v is 1 or 2 according to whether the resulting $C_{\vec{r}} \vdash e_{\vec{r}}$ is returned via (4) or (5). Since each number k_i is distinct, the number of different such sequences is bounded by:

$$2 \sum_{m=1}^{k'} \frac{k'!}{(k'-m)!} (l'(l'-1))^{m-1} \;\le\; (k'(l')^2)^{k'} \quad , \tag{17}$$

where the right-hand-side gives a somewhat loose upper bound. Therefore, the number of possible results $C_{\vec{r}} \vdash e_{\vec{r}}$ for the algorithm for $\mu x_{n+1}.t'$ is at most $(k'(l')^2)^{k'}$. □