

Roberto Bruni, Ugo Montanari

Models of Computation

– Monograph –

April 11, 2016

DRAFT

Springer

Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two facilities, which we may call intuition and ingenuity.

*Alan Turing*¹

¹ The purpose of ordinal logics (from Systems of Logic Based on Ordinals), Proceedings of the London Mathematical Society, series 2, vol. 45, 1939.

Contents

Part I Preliminaries

1	Introduction	3
1.1	Structure and Meaning	3
1.1.1	Syntax, Types and Pragmatics	4
1.1.2	Semantics	4
1.1.3	Mathematical Models of Computation	6
1.2	A Taste of Semantics Methods: Numerical Expressions	9
1.3	Applications of Semantics	17
1.4	Key Topics and Techniques	20
1.4.1	Induction and Recursion	20
1.4.2	Semantic Domains	21
1.4.3	Bisimulation	23
1.4.4	Temporal and Modal Logics	25
1.4.5	Probabilistic Systems	25
1.5	Chapters Contents and Reading Guide	26
1.6	Further Reading	28
	References	30
2	Preliminaries	33
2.1	Notation	33
2.1.1	Basic Notation	33
2.1.2	Signatures and Terms	34
2.1.3	Substitutions	35
2.1.4	Unification Problem	35
2.2	Inference Rules and Logical Systems	37
2.3	Logic Programming	45
	Problems	47

Part II IMP: a simple imperative language

3	Operational Semantics of IMP	53
3.1	Syntax of IMP	53
3.1.1	Arithmetic Expressions	54
3.1.2	Boolean Expressions	54
3.1.3	Commands	55
3.1.4	Abstract Syntax	55
3.2	Operational Semantics of IMP	56
3.2.1	Memory State	56
3.2.2	Inference Rules	57
3.2.3	Examples	62
3.3	Abstract Semantics: Equivalence of Expressions and Commands ...	66
3.3.1	Examples: Simple Equivalence Proofs	67
3.3.2	Examples: Parametric Equivalence Proofs	69
3.3.3	Examples: Inequality Proofs	71
3.3.4	Examples: Diverging Computations	73
	Problems	75
4	Induction and Recursion	79
4.1	Noether Principle of Well-founded Induction	79
4.1.1	Well-founded Relations	79
4.1.2	Noether Induction	85
4.1.3	Weak Mathematical Induction	86
4.1.4	Strong Mathematical Induction	87
4.1.5	Structural Induction	87
4.1.6	Induction on Derivations	90
4.1.7	Rule Induction	91
4.2	Well-founded Recursion	95
	Problems	100
5	Partial Orders and Fixpoints	105
5.1	Orders and Continuous Functions	105
5.1.1	Orders	106
5.1.2	Hasse Diagrams	108
5.1.3	Chains	111
5.1.4	Complete Partial Orders	113
5.2	Continuity and Fixpoints	116
5.2.1	Monotone and Continuous Functions	116
5.2.2	Fixpoints	118
5.3	Immediate Consequence Operator	122
5.3.1	The Operator \hat{R}	122
5.3.2	Fixpoint of \hat{R}	123
	Problems	126

6	Denotational Semantics of IMP	129
6.1	λ -Notation	129
6.1.1	λ -Notation: Main Ideas	130
6.1.2	Alpha-Conversion, Beta-Rule and Capture-Avoiding Substitution	133
6.2	Denotational Semantics of IMP	135
6.2.1	Denotational Semantics of Arithmetic Expressions: The Function \mathcal{A}	136
6.2.2	Denotational Semantics of Boolean Expressions: The Function \mathcal{B}	137
6.2.3	Denotational Semantics of Commands: The Function \mathcal{C}	138
6.3	Equivalence Between Operational and Denotational Semantics	143
6.3.1	Equivalence Proofs For Expressions	143
6.3.2	Equivalence Proof for Commands	144
6.4	Computational Induction	151
	Problems	154
 Part III HOFL: a higher-order functional language		
7	Operational Semantics of HOFL	159
7.1	Syntax of HOFL	159
7.1.1	Typed Terms	160
7.1.2	Typability and Typechecking	162
7.2	Operational Semantics of HOFL	166
	Problems	173
8	Domain Theory	177
8.1	The Flat Domain of Integer Numbers \mathbb{Z}_\perp	177
8.2	Cartesian Product of Two Domains	178
8.3	Functional Domains	180
8.4	Lifting	183
8.5	Function's Continuity Theorems	185
8.6	Apply, Curry and Fix	188
	Problems	192
9	HOFL Denotational Semantics	193
9.1	HOFL Semantic Domains	193
9.2	HOFL Evaluation Function	194
9.2.1	Constants	194
9.2.2	Variables	194
9.2.3	Binary Operators	195
9.2.4	Conditional	195
9.2.5	Pairing	196
9.2.6	Projections	196
9.2.7	Lambda Abstraction	197
9.2.8	Function Application	197

9.2.9	Recursion	197
9.3	Continuity of Meta-language's Functions	199
9.4	Substitution Lemma	201
	Problems	202
10	Equivalence between HOFL denotational and operational semantics ..	205
10.1	Completeness	206
10.2	Equivalence (on Convergence)	209
10.3	Operational and Denotational Equivalences of Terms	211
10.4	A Simpler Denotational Semantics	212
	Problems	213
Part IV Concurrent Systems		
11	CCS, the Calculus for Communicating Systems	219
11.1	Syntax of CCS	224
11.2	Operational Semantics of CCS	225
11.2.1	Action Prefix	226
11.2.2	Restriction	226
11.2.3	Relabelling	226
11.2.4	Choice	227
11.2.5	Parallel Composition	227
11.2.6	Recursion	228
11.2.7	CCS with Value Passing	231
11.2.8	Recursive Declarations and the Recursion Operator	232
11.3	Abstract Semantics of CCS	234
11.3.1	Graph Isomorphism	234
11.3.2	Trace Equivalence	236
11.3.3	Bisimilarity	237
11.4	Compositionality	243
11.4.1	Bisimilarity is Preserved by Choice	244
11.5	A Logical View to Bisimilarity: Hennessy-Milner Logic	245
11.6	Axioms for Strong Bisimilarity	248
11.7	Weak Semantics of CCS	250
11.7.1	Weak Bisimilarity	250
11.7.2	Weak Observational Congruence	252
11.7.3	Dynamic Bisimilarity	253
	Problems	254
12	Temporal Logic and μ-Calculus	259
12.1	Temporal Logic	259
12.1.1	Linear Temporal Logic	260
12.1.2	Computation Tree Logic	262
12.2	μ -Calculus	264
12.3	Model Checking	267
	Problems	268

13	π-Calculus	271
13.1	Name Mobility	271
13.2	Syntax of the π -calculus	274
13.3	Operational Semantics of the π -calculus	276
13.3.1	Action Prefix	277
13.3.2	Choice	278
13.3.3	Name Matching	278
13.3.4	Parallel Composition	278
13.3.5	Restriction	279
13.3.6	Scope Extrusion	279
13.3.7	Replication	279
13.3.8	A Sample Derivation	280
13.4	Structural Equivalence of π -calculus	281
13.4.1	Reduction semantics	281
13.5	Abstract Semantics of the π -calculus	282
13.5.1	Strong Early Ground Bisimulations	283
13.5.2	Strong Late Ground Bisimulations	284
13.5.3	Strong Full Bisimilarities	285
13.5.4	Weak Early and Late Ground Bisimulations	286
	Problems	287

Part V Probabilistic Systems

14	Measure Theory and Markov Chains	291
14.1	Probabilistic and Stochastic Systems	291
14.2	Measure Theory	292
14.2.1	σ -field	292
14.2.2	Constructing a σ -field	293
14.2.3	Continuous Random Variables	295
14.2.4	Stochastic Processes	299
14.3	Markov Chains	299
14.3.1	Discrete and Continuous Time Markov Chain	300
14.3.2	DTMC as LTS	301
14.3.3	DTMC Steady State Distribution	303
14.3.4	CTMC as LTS	305
14.3.5	Embedded DTMC of a CTMC	306
14.3.6	CTMC Bisimilarity	306
14.3.7	DTMC Bisimilarity	308
	Problems	309
15	Markov Chains with Actions and Non-determinism	313
15.1	Discrete Markov Chains With Actions	313
15.1.1	Reactive DTMC	314
15.1.2	DTMC With Non-determinism	316
	Problems	319

16 PEPA - Performance Evaluation Process Algebra	321
16.1 From Qualitative to Quantitative Analysis	321
16.2 CSP	322
16.2.1 Syntax of CSP	322
16.2.2 Operational Semantics of CSP	323
16.3 PEPA	324
16.3.1 Syntax of PEPA	324
16.3.2 Operational Semantics of PEPA	326
Problems	331
Glossary	335
Solutions	337
Index	351

DRAFT

Acronyms

\sim	operational equivalence in IMP (see Definition 3.3)
\equiv_{den}	denotational equivalence in HOFL (see Definition 10.4)
\equiv_{op}	operational equivalence in HOFL (see Definition 10.3)
\approx	CCS strong bisimilarity (see Definition 11.5)
$\approx\approx$	CCS weak bisimilarity (see Definition 11.16)
$\approx\approx\approx$	CCS weak observational congruence (see Section 11.7.2)
\approx_d	CCS dynamic bisimilarity (see Definition 11.17)
$\overset{\circ}{\sim}_E$	π -calculus early bisimilarity (see Definition 13.3)
$\overset{\circ}{\sim}_L$	π -calculus late bisimilarity (see Definition 13.4)
\sim_E	π -calculus strong early full bisimilarity (see Section 13.5.3)
\sim_L	π -calculus strong late full bisimilarity (see Section 13.5.3)
$\overset{\bullet}{\sim}_E$	π -calculus weak early bisimilarity (see Section 13.5.4)
$\overset{\bullet}{\sim}_L$	π -calculus weak late bisimilarity (see Section 13.5.4)
\mathcal{A}	interpretation function for the denotational semantics of IMP arithmetic expressions (see Section 6.2.1)
<i>ack</i>	Ackermann function (see Example 4.18)
<i>Aexp</i>	set of IMP arithmetic expressions (see Chapter 3)
\mathcal{B}	interpretation function for the denotational semantics of IMP boolean expressions (see Section 6.2.2)
<i>Bexp</i>	set of IMP boolean expressions (see Chapter 3)
\mathbb{B}	set of booleans
\mathcal{C}	interpretation function for the denotational semantics of IMP commands (see Section 6.2.3)
CCS	Calculus of Communicating Systems (see Chapter 11)
<i>Com</i>	set of IMP commands (see Chapter 3)
CPO	Complete Partial Order (see Definition 5.11)
CPO_{\perp}	Complete Partial Order with bottom (see Definition 5.12)
CSP	Communicating Sequential Processes (see Section 16.2)
CTL	Computation Tree Logic (see Section 12.1.2)
CTMC	Continuous Time Markov Chain (see Definition 14.15)

DTMC	Discrete Time Markov Chain (see Definition 14.14)
<i>Env</i>	set of HOFL environments (see Chapter 9)
fix	(least) fixpoint (see Definition 5.2.2)
FIX	(greatest) fixpoint
gcd	greatest common divisor
HML	Hennessy-Milner modal Logic (see Section 11.5)
HM-Logic	Hennessy-Milner modal Logic (see Section 11.5)
HOFL	A Higher-Order Functional Language (see Chapter 7)
IMP	A simple IMPerative language (see Chapter 3)
<i>int</i>	integer type in HOFL (see Definition 7.2)
Loc	set of locations (see Chapter 3)
LTL	Linear Temporal Logic (see Section 12.1.1)
LTS	Labelled Transition System (see Definition 11.2)
lub	least upper bound (see Definition 5.7)
\mathbb{N}	set of natural numbers
\mathcal{P}	set of closed CCS processes (see Definition 11.1)
PEPA	Performance Evaluation Process Algebra (see Chapter 16)
Pf	set of partial functions on natural numbers (see Example 5.13)
PI	set of partial injective functions on natural numbers (see Problem 5.12)
PO	Partial Order (see Definition 5.1)
PTS	Probabilistic Transition System (see Section 14.3.2)
\mathbb{R}	set of real numbers
\mathcal{T}	set of HOFL types (see Definition 7.2)
Tf	set of total functions from \mathbb{N} to \mathbb{N}_+ (see Example 5.14)
<i>Var</i>	set of HOFL variables (see Chapter 7)
\mathbb{Z}	set of integers

Part III
HOFL: a higher-order functional language

DRAFT

This part focuses on models for sequential computations that are associated to HOFL, a higher-order declarative language that follows the functional style. Chapter 7 presents the syntax, typing and operational semantics of HOFL, while Chapter 9 defines its denotational semantics. The two are related in Chapter 10. Chapter 8 extends the theory presented in Chapter 5 to allow the construction of more complex domains, as needed by the type-constructors available in HOFL.

DRAFT

Chapter 8

Domain Theory

Order, unity and continuity are human inventions just as truly as catalogues and encyclopedias. (Bertrand Russell)

Abstract As done for IMP we would like to introduce the denotational semantics of HOFL, for which we need to develop a proper domain theory that is more sophisticated than the one presented in Chapter 5. In order to define the denotational semantics of IMP we have shown that the semantic domain of commands, for which we need to apply fixpoint theorem, has the required properties. The situation is more complicated for HOFL, because HOFL provides constructors for infinitely many term types, so there are infinitely many domains to be considered. We will handle this problem by showing, using structural induction, that the type constructors of HOFL correspond to domains which are equipped with adequate CPO_{\perp} structures and that we can define useful continuous functions between them.

8.1 The Flat Domain of Integer Numbers \mathbb{Z}_{\perp}

The first domain we introduce is very simple: it consists of all the integers numbers together with a distinguished bottom element. It relies on a flat order in the sense of Example 5.5.

Definition 8.1 (\mathbb{Z}_{\perp}). We define the CPO with bottom $\mathbb{Z}_{\perp} = (\mathbb{Z} \cup \{\perp\}, \sqsubseteq)$ as follows:

- \mathbb{Z} is the set of integer numbers;
- \perp is a distinguished bottom element that we add to the purpose;
- $\forall x \in \mathbb{Z} \cup \{\perp\}. \perp \sqsubseteq x$ and $x \sqsubseteq x$

It is immediate to check that \mathbb{Z}_{\perp} is a CPO with bottom, where \perp is the bottom element and each chain has a lub because chains are all finite: they either contain 1 or 2 different elements.

Remark 8.1. Since in this chapter we present several different domains, each coming with its proper order relation and bottom element, we find it useful to annotate them with the name of the domain as a subscript to avoid ambiguities. For example, we can write $\perp_{\mathbb{Z}_{\perp}}$ to make explicit that we are referring to the bottom element of the

domain \mathbb{Z}_\perp . Also note that the subscript \perp we attach to the name of the domain \mathbb{Z} is just a tag and it should not be confused with the name of the bottom element itself: it is the standard way to indicate that the domain \mathbb{Z} is enriched with a bottom element (e.g., we could have used a different notation like $\underline{\mathbb{Z}}$ to the same purpose).

8.2 Cartesian Product of Two Domains

Given two CPO $_\perp$ s we can combine them to obtain another CPO $_\perp$ whose elements are pairs formed with one element from each CPO $_\perp$.

Definition 8.2. Let:

$$\mathcal{D} = (D, \sqsubseteq_D) \quad \mathcal{E} = (E, \sqsubseteq_E)$$

be two CPO $_\perp$ s. Now we define their Cartesian product domain

$$\mathcal{D} \times \mathcal{E} = (D \times E, \sqsubseteq_{D \times E})$$

1. whose elements are the pairs of elements from D and E ; and
2. whose order $\sqsubseteq_{D \times E}$ is defined as follows:¹

$$\forall d_0, d_1 \in D, \forall e_0, e_1 \in E. (d_0, e_0) \sqsubseteq_{D \times E} (d_1, e_1) \Leftrightarrow d_0 \sqsubseteq_D d_1 \wedge e_0 \sqsubseteq_E e_1$$

Proposition 8.1. $(D \times E, \sqsubseteq_{D \times E})$ is a partial order with bottom.

Proof. We need to show that the relation $\sqsubseteq_{D \times E}$ is reflexive, antisymmetric and transitive:

reflexivity: since \sqsubseteq_D and \sqsubseteq_E are reflexive we have $\forall e \in E. e \sqsubseteq_E e$ and $\forall d \in D. d \sqsubseteq_D d$ so by definition of $\sqsubseteq_{D \times E}$ we have

$$\forall d \in D \forall e \in E. (d, e) \sqsubseteq_{D \times E} (d, e).$$

antisymmetry: let us assume $(d_0, e_0) \sqsubseteq_{D \times E} (d_1, e_1)$ and $(d_1, e_1) \sqsubseteq_{D \times E} (d_0, e_0)$ so by definition of $\sqsubseteq_{D \times E}$ we have $d_0 \sqsubseteq_D d_1$ (using the first relation) and $d_1 \sqsubseteq_D d_0$ (by using the second relation) so it must be $d_0 = d_1$ and similarly $e_0 = e_1$, hence $(d_0, e_0) = (d_1, e_1)$.

transitivity: let us assume $(d_0, e_0) \sqsubseteq_{D \times E} (d_1, e_1)$ and $(d_1, e_1) \sqsubseteq_{D \times E} (d_2, e_2)$. By definition of $\sqsubseteq_{D \times E}$ we have $d_0 \sqsubseteq_D d_1$, $d_1 \sqsubseteq_D d_2$, $e_0 \sqsubseteq_E e_1$ and $e_1 \sqsubseteq_E e_2$. By transitivity of \sqsubseteq_D and \sqsubseteq_E we have $d_0 \sqsubseteq_D d_2$ and $e_0 \sqsubseteq_E e_2$. By definition of $\sqsubseteq_{D \times E}$ we get $(d_0, e_0) \sqsubseteq_{D \times E} (d_2, e_2)$.

Finally, we show that there is a bottom element. Let $\perp_{D \times E} = (\perp_D, \perp_E)$. In fact $\forall d \in D, e \in E. \perp_D \sqsubseteq_D d \wedge \perp_E \sqsubseteq_E e$, thus $(\perp_D, \perp_E) \sqsubseteq_{D \times E} (d, e)$. \square

It remains to show the completeness of $\mathcal{D} \times \mathcal{E}$.

¹ Note that the order is different from the lexicographic one considered in Example 4.9.

Theorem 8.1 (Completeness of $\mathcal{D} \times \mathcal{E}$). *The PO $\mathcal{D} \times \mathcal{E}$ defined above is complete.*

Proof. We prove that for each chain $(d_i, e_i)_{i \in \mathbb{N}}$ it holds:

$$\bigsqcup_{i \in \mathbb{N}} (d_i, e_i) = \left(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i \right)$$

Obviously $(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i)$ is an upper bound, indeed for each $j \in \mathbb{N}$ we have $d_j \sqsubseteq_D \bigsqcup_{i \in \mathbb{N}} d_i$ and $e_j \sqsubseteq_E \bigsqcup_{i \in \mathbb{N}} e_i$ so by definition of $\sqsubseteq_{D \times E}$ it holds $(d_j, e_j) \sqsubseteq_{D \times E} (\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i)$.

Moreover $(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i)$ is also the least upper bound. Indeed, let (d, e) be an upper bound of $\{(d_i, e_i)\}_{i \in \mathbb{N}}$, since $\bigsqcup_{i \in \mathbb{N}} d_i$ is the lub of $\{d_i\}_{i \in \mathbb{N}}$ we have $\bigsqcup_{i \in \mathbb{N}} d_i \sqsubseteq_D d$, furthermore we have that $\bigsqcup_{i \in \mathbb{N}} e_i$ is the lub of $\{e_i\}_{i \in \mathbb{N}}$ then $\bigsqcup_{i \in \mathbb{N}} e_i \sqsubseteq_E e$. So by definition of $\sqsubseteq_{D \times E}$ we have $(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i) \sqsubseteq_{D \times E} (d, e)$. Thus $(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i)$ is the least upper bound. \square

We can now define suitable projection operators over $\mathcal{D} \times \mathcal{E}$.

Definition 8.3 (Projection operators π_1 and π_2). Let $(d, e) \in D \times E$ be a pair, we define the left and right projection functions $\pi_1 : D \times E \rightarrow D$ and $\pi_2 : D \times E \rightarrow E$ as follows.

$$\pi_1((d, e)) \stackrel{\text{def}}{=} d \quad \text{and} \quad \pi_2((d, e)) \stackrel{\text{def}}{=} e.$$

Recall that in order to use a function in domain theory we have to show that it is continuous; this ensures that the function respects the domain structure (i.e., the function preserves the order and limits) and so we can calculate its fixpoints to solve recursive equations. So we have to prove that each function which we use on $\mathcal{D} \times \mathcal{E}$ is continuous. The proof that projections are monotone is immediate and left as an exercise (see Problem 8.1).

Theorem 8.2 (Continuity of π_1 and π_2). *Let π_1 and π_2 be the projection functions in Definition 8.3 and let $\{(d_i, e_i)\}_{i \in \mathbb{N}}$ be a chain of elements in $\mathcal{D} \times \mathcal{E}$, then:*

$$\pi_1 \left(\bigsqcup_{i \in \mathbb{N}} (d_i, e_i) \right) = \bigsqcup_{i \in \mathbb{N}} \pi_1((d_i, e_i)) \quad \pi_2 \left(\bigsqcup_{i \in \mathbb{N}} (d_i, e_i) \right) = \bigsqcup_{i \in \mathbb{N}} \pi_2((d_i, e_i))$$

Proof. Let us prove the first statement:

$$\begin{aligned} \pi_1 \left(\bigsqcup_{i \in \mathbb{N}} (d_i, e_i) \right) &= \pi_1 \left(\left(\bigsqcup_{i \in \mathbb{N}} d_i, \bigsqcup_{i \in \mathbb{N}} e_i \right) \right) \quad (\text{by definition of limit in } D \times E) \\ &= \bigsqcup_{i \in \mathbb{N}} d_i \quad (\text{by definition of projection}) \\ &= \bigsqcup_{i \in \mathbb{N}} \pi_1((d_i, e_i)) \quad (\text{by definition of projection}). \end{aligned}$$

For the second statement the proof is completely analogous. \square

8.3 Functional Domains

Let (D, \sqsubseteq_D) and (E, \sqsubseteq_E) be two CPOs. In the following we denote by $D \rightarrow E \stackrel{\text{def}}{=} \{f \mid f : D \rightarrow E\}$ the set of all functions from D to E (where the order relations are not important), while we denote by $[D \rightarrow E] \subseteq D \rightarrow E$ the set of all continuous functions from D to E (i.e., $[D \rightarrow E]$ contains just the functions that preserve order and limits). As for Cartesian product, we can define a suitable order on the set $[D \rightarrow E]$ to get a CPO_\perp . Note that as usual we require the continuity of the functions to preserve the applicability of fixpoint theory.

Definition 8.4. Let us consider the CPO_\perp s:

$$\mathcal{D} = (D, \sqsubseteq_D) \quad \mathcal{E} = (E, \sqsubseteq_E)$$

We define an order on the set of continuous functions from D to E as follows:

$$[\mathcal{D} \rightarrow \mathcal{E}] = ([D \rightarrow E], \sqsubseteq_{[D \rightarrow E]})$$

where:

1. $[D \rightarrow E] = \{f \mid f : D \rightarrow E, f \text{ is continuous}\}$
2. $f \sqsubseteq_{[D \rightarrow E]} g \Leftrightarrow \forall d \in D. f(d) \sqsubseteq_E g(d)$

We leave as an exercise the proof that $[\mathcal{D} \rightarrow \mathcal{E}]$ is a PO with bottom, namely that the relation $\sqsubseteq_{[D \rightarrow E]}$ is reflexive, antisymmetric, transitive and that the function $\perp_{[D \rightarrow E]} : D \rightarrow E$ defined by letting, for any $d \in D$:

$$\perp_{[D \rightarrow E]}(d) \stackrel{\text{def}}{=} \perp_E$$

is continuous and that it is also the bottom element of $[\mathcal{D} \rightarrow \mathcal{E}]$ (see Problem 8.2).

We show that the PO $[\mathcal{D} \rightarrow \mathcal{E}]$ is complete. In order to simplify the proof we introduce first the following lemmas.

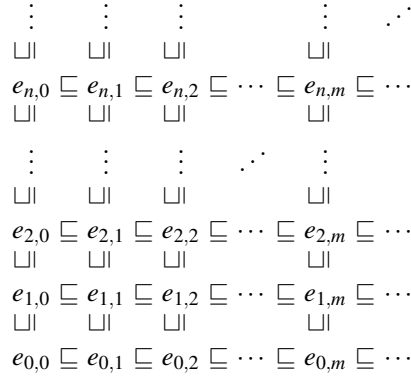
Lemma 8.1 (Switch Lemma). *Let (E, \sqsubseteq_E) be a CPO whose elements are of the form $e_{n,m}$ with $n, m \in \mathbb{N}$. If \sqsubseteq_E is such that:*

$$e_{n,m} \sqsubseteq_E e_{n',m'} \text{ if } n \leq n' \text{ and } m \leq m'$$

then, it holds:

$$\bigsqcup_{n,m \in \mathbb{N}} e_{n,m} = \bigsqcup_{n \in \mathbb{N}} \left(\bigsqcup_{m \in \mathbb{N}} e_{n,m} \right) = \bigsqcup_{m \in \mathbb{N}} \left(\bigsqcup_{n \in \mathbb{N}} e_{n,m} \right) = \bigsqcup_{k \in \mathbb{N}} e_{k,k}$$

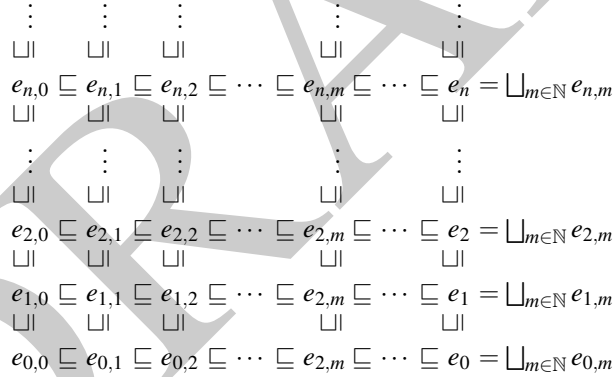
Proof. The relation between the elements of E can be summarized as follows:



We show that all the following sets have the same upper bounds:

$$\{e_{n,m}\}_{n,m \in \mathbb{N}} \quad \left\{ \bigsqcup_{m \in \mathbb{N}} e_{n,m} \right\}_{n \in \mathbb{N}} \quad \left\{ \bigsqcup_{n \in \mathbb{N}} e_{n,m} \right\}_{m \in \mathbb{N}} \quad \{e_{k,k}\}_{k \in \mathbb{N}}$$

- Let us consider the first two sets. For any $n \in \mathbb{N}$, let $e_n = \bigsqcup_{j \in \mathbb{N}} e_{n,j}$. This amounts to consider each row of the above diagram and compute the least upper bound for the elements in the same row. Clearly, $e_{n_1} \sqsubseteq e_{n_2}$ when $n_1 \leq n_2$ because for any $j \in \mathbb{N}$ an upper bound of $e_{n_2,j}$ is also an upper bound of $e_{n_1,j}$.



Let e be an upper bound of $\{e_i\}_{i \in \mathbb{N}}$, we want to show that e is an upper bound for $\{e_{n,m}\}_{n,m \in \mathbb{N}}$. Take any $n, m \in \mathbb{N}$. Then

$$e_{n,m} \sqsubseteq \bigsqcup_{j \in \mathbb{N}} e_{n,j} = e_n \sqsubseteq e$$

since $e_{n,m}$ is an element of the chain $\{e_{n,j}\}_{j \in \mathbb{N}}$ whose limit is $e_n = \bigsqcup_{j \in \mathbb{N}} e_{n,j}$. Thus e is an upper bound for $\{e_{n,m}\}_{n,m \in \mathbb{N}}$.

Vice versa, let e be an upper bound of $\{e_{i,j}\}_{i,j \in \mathbb{N}}$ and consider $e_n = \bigsqcup_{m \in \mathbb{N}} e_{n,m}$ for some n . Since $\{e_{n,m}\}_{m \in \mathbb{N}} \subseteq \{e_{i,j}\}_{i,j \in \mathbb{N}}$, obviously e is an upper bound for $\{e_{n,m}\}_{m \in \mathbb{N}}$ and therefore $e_n \sqsubseteq e$, because e_n is the lub of $\{e_{n,m}\}_{m \in \mathbb{N}}$.

- The correspondence between the sets of upper bounds of $\{e_{n,m}\}_{n,m \in \mathbb{N}}$ and $\{\bigsqcup_{n \in \mathbb{N}} e_{n,m}\}_{m \in \mathbb{N}}$ can be proved analogously.
- Finally, let us consider the sets $\{e_{n,m}\}_{n,m \in \mathbb{N}}$ and $\{e_{k,k}\}_{k \in \mathbb{N}}$ and show that they have the same set of upper bounds.

Taken any $n, m \in \mathbb{N}$ the element, let $k = \max\{n, m\}$. We have

$$e_{n,m} \sqsubseteq e_{n,k} \sqsubseteq e_{k,k}$$

thus any upper bound of $\{e_{k,k}\}_{k \in \mathbb{N}}$ is also an upper bound of $\{e_{n,m}\}_{n,m \in \mathbb{N}}$.

Vice versa, it is immediate to check that $\{e_{k,k}\}_{k \in \mathbb{N}}$ is a subset of $\{e_{n,m}\}_{n,m \in \mathbb{N}}$ so any upper bound of $\{e_{n,m}\}_{n,m \in \mathbb{N}}$ is also an upper bound of $\{e_{k,k}\}_{k \in \mathbb{N}}$.

We conclude by noting that the set of upper bounds $\{e_{n,m}\}_{n,m \in \mathbb{N}}$ has a least element. In fact, $\{\bigsqcup_{m \in \mathbb{N}} e_{n,m}\}_{n \in \mathbb{N}}$ is a chain, and it has a lub because E is a CPO. \square

Lemma 8.2. *Let $\{f_n\}_{n \in \mathbb{N}}$ be a chain of functions² in $\mathcal{D} \rightarrow \mathcal{E}$. Then the lub $\bigsqcup_{n \in \mathbb{N}} f_n$ exists and it is defined as:*

$$\left(\bigsqcup_{n \in \mathbb{N}} f_n \right) (d) = \bigsqcup_{n \in \mathbb{N}} (f_n(d))$$

Proof. The function

$$h \stackrel{\text{def}}{=} \lambda d. \bigsqcup_{n \in \mathbb{N}} (f_n(d))$$

is clearly an upper bound for $\{f_n\}_{n \in \mathbb{N}}$ since for every $k \in \mathbb{N}$ and $d \in D$ we have $f_k(d) \sqsubseteq_E \bigsqcup_{n \in \mathbb{N}} f_n(d)$.

The function h is also the lub of $\{f_n\}_{n \in \mathbb{N}}$. In fact, given any other upper bound g , i.e., such that $f_n \sqsubseteq_{D \rightarrow E} g$ for any $n \in \mathbb{N}$, we have that for any $d \in D$ the element $g(d)$ is an upper bound of the chain $\{f_n(d)\}_{n \in \mathbb{N}}$ and therefore $\bigsqcup_{n \in \mathbb{N}} (f_n(d)) \sqsubseteq_E g(d)$. \square

Lemma 8.3. *Let $\{f_n\}_{n \in \mathbb{N}}$ be a chain of continuous functions in $[\mathcal{D} \rightarrow \mathcal{E}]$ and let $\{d_n\}_{n \in \mathbb{N}}$ be a chain of \mathcal{D} . Then, the function*

$$h \stackrel{\text{def}}{=} \lambda d. \bigsqcup_{n \in \mathbb{N}} (f_n(d))$$

is continuous, namely

$$h \left(\bigsqcup_{m \in \mathbb{N}} d_m \right) = \bigsqcup_{m \in \mathbb{N}} h(d_m)$$

Furthermore, h is the lub of $\{f_n\}_{n \in \mathbb{N}}$ not only in $\mathcal{D} \rightarrow \mathcal{E}$ as stated by Lemma 8.2, but also in $[\mathcal{D} \rightarrow \mathcal{E}]$.

² Note that the f_n are not necessarily continuous, because we select $\mathcal{D} \rightarrow \mathcal{E}$ and not $[\mathcal{D} \rightarrow \mathcal{E}]$.

Proof.

$$\begin{aligned}
h\left(\bigsqcup_{m \in \mathbb{N}} d_m\right) &= \bigsqcup_{n \in \mathbb{N}} \left(f_n\left(\bigsqcup_{m \in \mathbb{N}} d_m\right)\right) && \text{(by definition of } h\text{)} \\
&= \bigsqcup_{n \in \mathbb{N}} \left(\bigsqcup_{m \in \mathbb{N}} (f_n(d_m))\right) && \text{(by continuity of } f_n\text{)} \\
&= \bigsqcup_{m \in \mathbb{N}} \left(\bigsqcup_{n \in \mathbb{N}} (f_n(d_m))\right) && \text{by Lemma 8.1 (switch lemma)} \\
&= \bigsqcup_{m \in \mathbb{N}} h(d_m) && \text{(by definition of } h\text{)}
\end{aligned}$$

The upper bounds of $\{f_n\}_{n \in \mathbb{N}}$ in the PO $\mathcal{D} \rightarrow \mathcal{E}$ are a larger set than those in $[\mathcal{D} \rightarrow \mathcal{E}]$, thus if h is the lub in $\mathcal{D} \rightarrow \mathcal{E}$, it is also the lub in $[\mathcal{D} \rightarrow \mathcal{E}]$. \square

Theorem 8.3 ($[\mathcal{D} \rightarrow \mathcal{E}]$ is a CPO_\perp). *The PO $[\mathcal{D} \rightarrow \mathcal{E}]$ is a CPO_\perp*

Proof. The statement follows immediately from the previous lemmas. \square

8.4 Lifting

In IMP we introduced a lifting operator (see Definition 6.9) on functions $f: \Sigma \rightarrow \Sigma_\perp$ to derive a function $f^*: \Sigma_\perp \rightarrow \Sigma_\perp$ defined over the lifted domain Σ_\perp , and thus able to handle the argument \perp_{Σ_\perp} . In the semantics of HOFL we need the same operator in a more general fashion: we need to apply the lifting operator to any domain, not just Σ .

Definition 8.5 (Lifted domain). Let $\mathcal{D} = (D, \sqsubseteq_D)$ be a CPO and let \perp be an element not in D . We define the lifted domain $\mathcal{D}_\perp = (D_\perp, \sqsubseteq_{D_\perp})$ as follows:

- $D_\perp \stackrel{\text{def}}{=} \{\perp\} \uplus D = \{(0, \perp)\} \cup \{1\} \times D$
- $\perp_{D_\perp} \stackrel{\text{def}}{=} (0, \perp)$
- $\forall x \in D_\perp. \perp_{D_\perp} \sqsubseteq_{D_\perp} x$
- $\forall d_1, d_2 \in D. d_1 \sqsubseteq_D d_2 \Rightarrow (1, d_1) \sqsubseteq_{D_\perp} (1, d_2)$

We leave it as an exercise to show that \mathcal{D}_\perp is a CPO_\perp (see Problem 8.3).

We define a lifting function $[\cdot]: D \rightarrow D_\perp$ by letting, for any $d \in D$:

$$[d] \stackrel{\text{def}}{=} (1, d)$$

As it was the case for Σ in the IMP semantics, when we add a bottom element to a domain \mathcal{D} we would like to extend the continuous functions in $[D \rightarrow E]$ to continuous functions in $[D_\perp \rightarrow E]$. The function defining the extension should itself be continuous.

Definition 8.6 (Lifting). Let \mathcal{D} be a CPO and let \mathcal{E} be a CPO_{\perp} . We define the lifting operator $(\cdot)^* : [D \rightarrow E] \rightarrow [D_{\perp} \rightarrow E]$ as follows:

$$\forall f \in [D \rightarrow E]. f^*(x) \stackrel{\text{def}}{=} \begin{cases} \perp_E & \text{if } x = \perp_{D_{\perp}} \\ f(d) & \text{if } x = \lfloor d \rfloor \end{cases}$$

We need to prove that the definition is well-given and that the lifting operator is continuous.

Theorem 8.4. Let \mathcal{D}, \mathcal{E} be two CPOs.

1. If $f : D \rightarrow E$ is continuous, then f^* is continuous.
2. The operator $(\cdot)^*$ is continuous.

Proof. We prove the two statements separately.

1. We need to prove that if $f \in [D \rightarrow E]$, then $f^* \in [D_{\perp} \rightarrow E]$. Let $\{x_n\}_{n \in \mathbb{N}}$ be a chain in \mathcal{D}_{\perp} . We have to prove $f^*(\bigsqcup_{n \in \mathbb{N}} x_n) = \bigsqcup_{n \in \mathbb{N}} f^*(x_n)$.
If $\forall n \in \mathbb{N}. x_n = \perp_{D_{\perp}}$, then this is obvious.
Otherwise, for some $k \in \mathbb{N}$ there must exist a set of elements $\{d_{n+k}\}_{n \in \mathbb{N}}$ in D such that for all $m \geq k$ we have $x_m = \lfloor d_m \rfloor$ and also $\bigsqcup_{n \in \mathbb{N}} x_n = \bigsqcup_{n \in \mathbb{N}} x_{n+k} = \lfloor \bigsqcup_{n \in \mathbb{N}} d_{n+k} \rfloor$ (by prefix independence of the limit, Lemma 5.1). Then:

$$\begin{aligned} f^*\left(\bigsqcup_{n \in \mathbb{N}} x_n\right) &= f^*\left(\bigsqcup_{n \in \mathbb{N}} \lfloor d_{n+k} \rfloor\right) && \text{by the above argument} \\ &= f\left(\bigsqcup_{n \in \mathbb{N}} d_{n+k}\right) && \text{by definition of lifting} \\ &= \bigsqcup_{n \in \mathbb{N}} f(d_{n+k}) && \text{by continuity of } f \\ &= \bigsqcup_{n \in \mathbb{N}} f^*(\lfloor d_{n+k} \rfloor) && \text{by definition of lifting} \\ &= \bigsqcup_{n \in \mathbb{N}} f^*(x_{n+k}) && \text{by definition of } x_{n+k} \\ &= \bigsqcup_{n \in \mathbb{N}} f^*(x_n) && \text{by Lemma 5.1} \end{aligned}$$

2. We leave the proof that $(\cdot)^*$ is monotone as an exercise (see Problem 8.4).
Let $\{f_i\}_{i \in \mathbb{N}}$ be a chain of functions in $[\mathcal{D} \rightarrow \mathcal{E}]$. We will prove that for all $x \in D_{\perp}$:

$$\left(\bigsqcup_{i \in \mathbb{N}} f_i\right)^*(x) = \left(\bigsqcup_{i \in \mathbb{N}} f_i^*\right)(x)$$

if $x = \perp_{D_{\perp}}$ both sides of the equation simplify to \perp_E . So let us assume $x = \lfloor d \rfloor$ for some $d \in D$ we have:

$$\begin{aligned}
\left(\bigsqcup_{i \in \mathbb{N}} f_i\right)^* (\lfloor d \rfloor) &= \left(\bigsqcup_{i \in \mathbb{N}} f_i\right) (d) && \text{by definition of lifting} \\
&= \bigsqcup_{i \in \mathbb{N}} (f_i(d)) && \text{by def. of lub in a functional domain} \\
&= \bigsqcup_{i \in \mathbb{N}} (f_i^*(\lfloor d \rfloor)) && \text{by definition of lifting} \\
&= \left(\bigsqcup_{i \in \mathbb{N}} f_i^*\right) (\lfloor d \rfloor) && \text{by def. of lub in a functional domain}
\end{aligned}$$

□

8.5 Function's Continuity Theorems

In this section we show some theorems which allow to prove the continuity of some functions. We start proving that the composition of two continuous functions is continuous.

Theorem 8.5 (Continuity of composition). *Let $f \in [D \rightarrow E]$ and $g \in [E \rightarrow F]$. Their composition*

$$f; g = g \circ f \stackrel{\text{def}}{=} \lambda d. g(f(d)) : D \rightarrow F$$

is a continuous function, i.e., $g \circ f \in [D \rightarrow F]$.

Proof. The statement is just a rephrasing of Theorem 5.5. □

Now we consider a function whose outcome is a pair of values. So the function has a single CPO as domain but it returns a result over a product of CPOs.

$$f : D \rightarrow E_1 \times E_2$$

For this type of functions we introduce a theorem which allows to prove the continuity of f in a convenient way. We will consider f as the pairing of two simpler functions $g_1 : D \rightarrow E_1$ and $g_2 : D \rightarrow E_2$, such that $f(d) = (g_1(d), g_2(d))$ for any $d \in D$. Then we can prove the continuity of f from the continuity of g_1 and g_2 (and vice versa).

Theorem 8.6. *Let $f : D \rightarrow E_1 \times E_2$ be a function over CPOs and let*

$$g_1 \stackrel{\text{def}}{=} f; \pi_1 : D \rightarrow E_1 \quad g_2 \stackrel{\text{def}}{=} f; \pi_2 : D \rightarrow E_2$$

where $f; \pi_i = \lambda x. \pi_i(f(x))$ is the composition of f and π_i for $i = 1, 2$. Then: f is continuous if and only if g_1 and g_2 are continuous.

Proof. Notice that we have

$$\forall d \in D. f(d) = (g_1(d), g_2(d))$$

We prove the two implications separately.

- \Rightarrow) Since f is continuous, by Theorem 8.5 (continuity of composition) and Theorem 8.2 (continuity of projections), also g_1 and g_2 are continuous, because they are obtained as the composition of continuous functions.
- \Leftarrow) We assume the continuity of g_1 and g_2 and we want to prove that f is continuous. Let $\{d_i\}_{i \in \mathbb{N}}$ be a chain in D . We want to prove:

$$f\left(\bigsqcup_{i \in \mathbb{N}} d_i\right) = \bigsqcup_{i \in \mathbb{N}} f(d_i)$$

So we have:

$$\begin{aligned} f\left(\bigsqcup_{i \in \mathbb{N}} d_i\right) &= \left(g_1\left(\bigsqcup_{i \in \mathbb{N}} d_i\right), g_2\left(\bigsqcup_{i \in \mathbb{N}} d_i\right)\right) && \text{(by definition of } g_1, g_2) \\ &= \left(\bigsqcup_{i \in \mathbb{N}} g_1(d_i), \bigsqcup_{i \in \mathbb{N}} g_2(d_i)\right) && \text{(by continuity of } g_1 \text{ and } g_2) \\ &= \bigsqcup_{i \in \mathbb{N}} (g_1(d_i), g_2(d_i)) && \text{(by definition of lub of pairs)} \\ &= \bigsqcup_{i \in \mathbb{N}} f(d_i) && \text{(by definition of } g_1, g_2) \end{aligned}$$

□

Now let us consider the case of a function $f : D_1 \times D_2 \rightarrow E$ over CPOs which takes a pair of arguments in D_1 and D_2 and then returns an element of E . The following theorem allows us to study the continuity of f by analysing each parameter separately.

Theorem 8.7. *Let $f : D_1 \times D_2 \rightarrow E$ be a function over CPOs. Then f is continuous if and only if all the functions in the following two classes are continuous:*

1. $\forall d' \in D_1. f_{d'} : D_2 \rightarrow E$ is defined as $f_{d'} \stackrel{\text{def}}{=} \lambda y. f(d', y)$;
2. $\forall d'' \in D_2. f_{d''} : D_1 \rightarrow E$ is defined as $f_{d''} \stackrel{\text{def}}{=} \lambda x. f(x, d'')$.

Proof. We prove the two implications separately:

- \Rightarrow) If f is continuous then for all $d' \in D_1, d'' \in D_2$ the functions $f_{d'}$ and $f_{d''}$ are continuous, since we are considering only certain chains (where one element of the pair is fixed). For example, let us fix $d' \in D_1$ and consider a chain $\{d''_i\}_{i \in \mathbb{N}}$ in D_2 . Then we prove that $f_{d'}$ is continuous as follows:

$$\begin{aligned}
f_{d'} \left(\bigsqcup_{i \in \mathbb{N}} d_i'' \right) &= f \left(d', \bigsqcup_{i \in \mathbb{N}} d_i'' \right) && \text{(by definition of } f_{d'} \text{)} \\
&= f \left(\bigsqcup_{i \in \mathbb{N}} (d', d_i'') \right) && \text{(by definition of lub)} \\
&= \bigsqcup_{i \in \mathbb{N}} f(d', d_i'') && \text{(by continuity of } f \text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} f_{d'}(d_i'') && \text{(by definition of } f_{d'} \text{)}
\end{aligned}$$

Similarly, if we fix $d'' \in D_2$ and take a chain $\{d_i'\}_{i \in \mathbb{N}}$ in D_1 we have $f_{d''}(\bigsqcup_{i \in \mathbb{N}} d_i') = \bigsqcup_{i \in \mathbb{N}} f_{d''}(d_i')$.

\Leftarrow) In the opposite direction, assume that $f_{d'}$ and $f_{d''}$ are continuous for all elements $d' \in D_1$ and $d'' \in D_2$. We want to prove that f is continuous. Take a chain $\{(d'_k, d''_k)\}_{k \in \mathbb{N}}$. By definition of lub on pairs, we have

$$\bigsqcup_{k \in \mathbb{N}} (d'_k, d''_k) = \left(\bigsqcup_{i \in \mathbb{N}} d'_i, \bigsqcup_{j \in \mathbb{N}} d''_j \right)$$

Let $d'' \stackrel{\text{def}}{=} \bigsqcup_{j \in \mathbb{N}} d''_j$. It follows:

$$\begin{aligned}
f\left(\bigsqcup_{k \in \mathbb{N}} (d'_k, d''_k)\right) &= f\left(\bigsqcup_{i \in \mathbb{N}} d'_i, \bigsqcup_{j \in \mathbb{N}} d''_j\right) && \text{(by definition of lub on pairs)} \\
&= f\left(\bigsqcup_{i \in \mathbb{N}} d'_i, d''\right) && \text{(by definition of } d''\text{)} \\
&= f_{d''}\left(\bigsqcup_{i \in \mathbb{N}} d'_i\right) && \text{(by definition of } f_{d''}\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} f_{d''}(d'_i) && \text{(by continuity of } f_{d''}\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} f(d'_i, d'') && \text{(by definition of } f_{d''}\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} f_{d'_i}(d'') && \text{(by definition of } f_{d'_i}\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} f_{d'_i}\left(\bigsqcup_{j \in \mathbb{N}} d''_j\right) && \text{(by definition of } d''\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} \bigsqcup_{j \in \mathbb{N}} f_{d'_i}(d''_j) && \text{(by continuity of } f_{d'_i}\text{)} \\
&= \bigsqcup_{i \in \mathbb{N}} \bigsqcup_{j \in \mathbb{N}} f(d'_i, d''_j) && \text{(by definition of } f_{d'_i}\text{)} \\
&= \bigsqcup_{k \in \mathbb{N}} f(d'_k, d''_k) && \text{(by Lemma 8.1 (switch lemma))}
\end{aligned}$$

□

8.6 Apply, Curry and Fix

As done for IMP we will use the λ -notation as meta-language for the denotational semantics of HOFL. In Section 8.2 we have already defined two new continuous functions for our meta-language (π_1 and π_2). In this section we introduce some additional functions that will form the kernel of our meta-language.

Definition 8.7 (Apply). Let D and E be two CPOs. We define a function `apply` : $[D \rightarrow E] \times D \rightarrow E$ as follows:

$$\text{apply}(f, d) \stackrel{\text{def}}{=} f(d)$$

The function `apply` represents the application of a function in our meta-language: it takes a continuous function $f : D \rightarrow E$ and an element $d \in D$ and then returns $f(d)$ as

a result. We leave it as an exercise to prove that `apply` is monotone (see Problem 8.5). We prove that it is also continuous.

Theorem 8.8 (Continuity of `apply`). *Let $\text{apply} : [D \rightarrow E] \times D \rightarrow E$ be the function defined above and let $\{(f_n, d_n)\}_{n \in \mathbb{N}}$ be a chain in the $\text{CPO}_\perp [D \rightarrow E] \times D$ then:*

$$\text{apply} \left(\bigsqcup_{n \in \mathbb{N}} (f_n, d_n) \right) = \bigsqcup_{n \in \mathbb{N}} \text{apply}(f_n, d_n)$$

Proof. By Theorem 8.7 we can prove the continuity on each parameter separately.

- Let us fix $d \in D$ and take a chain $\{f_n\}_{n \in \mathbb{N}}$ in $[D \rightarrow E]$. We have:

$$\begin{aligned} \text{apply} \left(\left(\bigsqcup_{n \in \mathbb{N}} f_n \right), d \right) &= \left(\bigsqcup_{n \in \mathbb{N}} f_n \right) (d) && \text{(by definition)} \\ &= \bigsqcup_{n \in \mathbb{N}} (f_n(d)) && \text{(by definition of lub of functions)} \\ &= \bigsqcup_{n \in \mathbb{N}} \text{apply}(f_n, d) && \text{(by definition)} \end{aligned}$$

- Now we fix $f \in [D \rightarrow E]$ and take a chain $\{d_n\}_{n \in \mathbb{N}}$ in D . We have:

$$\begin{aligned} \text{apply} \left(f, \bigsqcup_{n \in \mathbb{N}} d_n \right) &= f \left(\bigsqcup_{n \in \mathbb{N}} d_n \right) && \text{(by definition)} \\ &= \bigsqcup_{n \in \mathbb{N}} f(d_n) && \text{(by continuity of } f) \\ &= \bigsqcup_{n \in \mathbb{N}} \text{apply}(f, d_n) && \text{(by definition)} \end{aligned}$$

So `apply` is a continuous function. □

Currying is the name of a technique for transforming a function that takes a pair (or, more generally, a tuple) of arguments into a function that takes each argument separately but computes the same result.

Definition 8.8 (Curry and un-curry). We define the function

$$\text{curry} : (D \times E \rightarrow F) \rightarrow (D \rightarrow E \rightarrow F)$$

by letting, for any $g : D \times E \rightarrow F$, $d \in D$ and $e \in E$:

$$\text{curry } g \ d \ e \stackrel{\text{def}}{=} g(d, e)$$

And we define the function

$$\text{un-curry} : (D \rightarrow E \rightarrow F) \rightarrow (D \times E \rightarrow F)$$

by letting, for any $h : D \rightarrow E \rightarrow F$, $d \in D$ and $e \in E$:

$$\text{un-curry } h(d, e) \stackrel{\text{def}}{=} h d e$$

Theorem 8.9 (Continuity of curry). *Let D, E, F be CPOs and $g : D \times E \rightarrow F$ be a continuous function. Then $(\text{curry } g) : D \rightarrow (E \rightarrow F)$ is a continuous function, namely given any chain $\{d_i\}_{i \in \mathbb{N}}$ in D :*

$$(\text{curry } g) \left(\bigsqcup_{i \in \mathbb{N}} d_i \right) = \bigsqcup_{i \in \mathbb{N}} (\text{curry } g)(d_i).$$

Proof. Let us note that since g is continuous, by Theorem 8.7, g is continuous separately on each argument. Then let us take $e \in E$ we have:

$$\begin{aligned} (\text{curry } g) \left(\bigsqcup_{i \in \mathbb{N}} d_i \right) (e) &= g \left(\left(\bigsqcup_{i \in \mathbb{N}} d_i \right), e \right) && \text{(by definition of curry } g) \\ &= \bigsqcup_{i \in \mathbb{N}} g(d_i, e) && \text{(by continuity of } g) \\ &= \bigsqcup_{i \in \mathbb{N}} ((\text{curry } g)(d_i)(e)) && \text{(by definition of curry } g) \end{aligned}$$

□

To define the denotational semantics of recursive definitions we need to provide a fixpoint operator. So it seems useful to introduce fix in our meta-language.

Definition 8.9 (Fix). Let D be a CPO_\perp . We define $\text{fix} : [D \rightarrow D] \rightarrow D$ as:

$$\text{fix} \stackrel{\text{def}}{=} \bigsqcup_{i \in \mathbb{N}} \lambda f. f^i(\perp_D)$$

Note that, since $\{\lambda f. f^i(\perp_D)\}_{i \in \mathbb{N}}$ is a chain of functions and $[D \rightarrow D] \rightarrow D$ is complete, we are guaranteed that the lub $\bigsqcup_{i \in \mathbb{N}} \lambda f. f^i(\perp_D)$ exists.

Theorem 8.10 (Continuity of fix). *The function $\text{fix} : [D \rightarrow D] \rightarrow D$ is continuous, namely $\text{fix} \in [[D \rightarrow D] \rightarrow D]$.*

Proof. We know that $[[D \rightarrow D] \rightarrow D]$ is complete, thus if for all $i \in \mathbb{N}$ the function $\lambda f. f^i(\perp_D)$ is continuous, then $\text{fix} = \bigsqcup_{i \in \mathbb{N}} \lambda f. f^i(\perp_D)$ is also continuous. We prove that $\forall i \in \mathbb{N}$. $\lambda f. f^i(\perp_D)$ is continuous by mathematical induction on i .

Base case: $\lambda f. f^0(\perp_D) = \lambda f. \perp_D$ is a constant, and thus continuous, function.

Inductive case: Let us assume that $g \stackrel{\text{def}}{=} \lambda f. f^i(\perp_D)$ is continuous, i.e., that given a chain $\{f_n\}_{n \in \mathbb{N}}$ in $[D \rightarrow D]$ we have $g(\bigsqcup_{n \in \mathbb{N}} f_n) = \bigsqcup_{n \in \mathbb{N}} g(f_n)$, and let us prove that $h \stackrel{\text{def}}{=} \lambda f. f^{i+1}(\perp_D)$ is continuous, namely that $h(\bigsqcup_{n \in \mathbb{N}} f_n) = \bigsqcup_{n \in \mathbb{N}} h(f_n)$. In fact we have:

$$\begin{aligned}
h\left(\bigsqcup_{n \in \mathbb{N}} f_n\right) &= \left(\bigsqcup_{n \in \mathbb{N}} f_n\right)^{i+1} (\perp_D) && \text{(by def. of } h\text{)} \\
&= \left(\bigsqcup_{n \in \mathbb{N}} f_n\right) \left(\left(\bigsqcup_{n \in \mathbb{N}} f_n\right)^i (\perp_D)\right) && \text{(by def. of } (\cdot)^{i+1}\text{)} \\
&= \left(\bigsqcup_{n \in \mathbb{N}} f_n\right) \left(g\left(\bigsqcup_{n \in \mathbb{N}} f_n\right)\right) && \text{(by def. of } g\text{)} \\
&= \left(\bigsqcup_{n \in \mathbb{N}} f_n\right) \left(\bigsqcup_{n \in \mathbb{N}} g(f_n)\right) && \text{(by ind. hyp.)} \\
&= \left(\bigsqcup_{n \in \mathbb{N}} f_n\right) \left(\bigsqcup_{n \in \mathbb{N}} f_n^i (\perp_D)\right) && \text{(by def of } g\text{)} \\
&= \bigsqcup_{n \in \mathbb{N}} \left(f_n \left(\bigsqcup_{m \in \mathbb{N}} f_m^i (\perp_D)\right)\right) && \text{(by def. of lub)} \\
&= \bigsqcup_{n \in \mathbb{N}} \bigsqcup_{m \in \mathbb{N}} f_n (f_m^i (\perp_D)) && \text{(by cont. of } f_n\text{)} \\
&= \bigsqcup_{k \in \mathbb{N}} f_k (f_k^i (\perp_D)) && \text{(by Lemma 8.1)} \\
&= \bigsqcup_{k \in \mathbb{N}} f_k^{i+1} (\perp_D) && \text{(by def. of } (\cdot)^{i+1}\text{)} \\
&= \bigsqcup_{n \in \mathbb{N}} h(f_n) && \text{(by def. of } h\text{)}
\end{aligned}$$

□

Finally we introduce the **let** operator, whose role is that of binding a name x to a de-lifted expression. Note that the continuity of the **let** operator directly follows from the continuity of the lifting operator.

Definition 8.10 (Let operator). Let \mathcal{E} be a CPO_{\perp} and $\lambda x. e$ a function in $[D \rightarrow E]$. We define the *let* operator as follows, where $d' \in D_{\perp}$:

$$\mathbf{let} \ x \leftarrow d'. e \stackrel{\text{def}}{=} \underbrace{\underbrace{(\lambda x. e)^*}_{D \rightarrow E} \left(\underbrace{d'}_{D_{\perp}} \right)}_{D_{\perp} \rightarrow E} = \begin{cases} \perp_E & \text{if } d' = \perp_{D_{\perp}} \\ e [d/x] & \text{if } d' = [d] \text{ for some } d \in D \end{cases}$$

Intuitively, taken $d' \in D_{\perp}$, if $d' = \perp$ then **let** $x \leftarrow d'. e$ returns \perp_E , otherwise it means that $d' = [d]$ for some $d \in D$ and thus it returns $e [d/x]$, as if $\lambda x. e$ was applied to d , i.e., $d' = [d]$ is de-lifted so that $\lambda x. e$ can be used.

Problems

8.1. Prove that the projection functions in Definition 8.3 are monotone.

8.2. Prove that the domain $[\mathcal{D} \rightarrow \mathcal{E}]$ from Definition 8.4 is a CPO_\perp .

8.3. Prove that the lifted domain \mathcal{D}_\perp from Definition 8.5 is a CPO_\perp .

8.4. Complete the proof of Theorem 8.4 for what is concerned with the monotonicity of the lifting function $(\cdot)^*$.

8.5. Prove that the function $\text{apply} : [D \rightarrow E] \times D \rightarrow E$ introduced in Definition 8.7 is monotone.

8.6. Let D be a CPO and $f : D \rightarrow D$ be a continuous function. Prove that the set of fixpoints of f is itself a CPO (under the order inherited from D).

8.7. Let D and E be two CPO_\perp s. A function $f : D \rightarrow E$ is called *strict* if $f(\perp_D) = \perp_E$. Prove that the set of strict functions from D to E is a CPO_\perp under the usual order.

8.8. Let D and E be two CPOs. Prove that the following two definitions of the order between continuous functions $f, g : D \rightarrow E$ are equivalent.

1. $f \sqsubseteq g \Leftrightarrow \forall d \in D. f(d) \sqsubseteq_E g(d)$.
2. $f \preceq g \Leftrightarrow \forall d_1, d_2 \in D. (d_1 \sqsubseteq_D d_2 \Rightarrow f(d_1) \sqsubseteq_E g(d_2))$

8.9. Let $\mathcal{D} = (D, \sqsubseteq_D)$ and $\mathcal{E} = (E, \sqsubseteq_E)$ be two CPOs. Their sum $\mathcal{D} + \mathcal{E}$ has:

1. The set of elements

$$\{\perp_{D+E}\} \cup D \uplus E = \{\perp_{D+E}\} \cup (\{0\} \times D) \cup (\{1\} \times E)$$

2. The order relation \sqsubseteq_{D+E} defined by letting:

- $(0, d_1) \sqsubseteq_{D+E} (0, d_2)$ if $d_1 \sqsubseteq_D d_2$;
- $(1, e_1) \sqsubseteq_{D+E} (1, e_2)$ if $e_1 \sqsubseteq_E e_2$;
- $\forall x \in \{\perp_{D+E}\} \cup D \uplus E. \perp_{D+E} \sqsubseteq_{D+E} x$.

Prove that $\mathcal{D} + \mathcal{E}$ is a CPO_\perp .

8.10. Prove that un-curry is continuous and inverse to curry (see Definition 8.8).