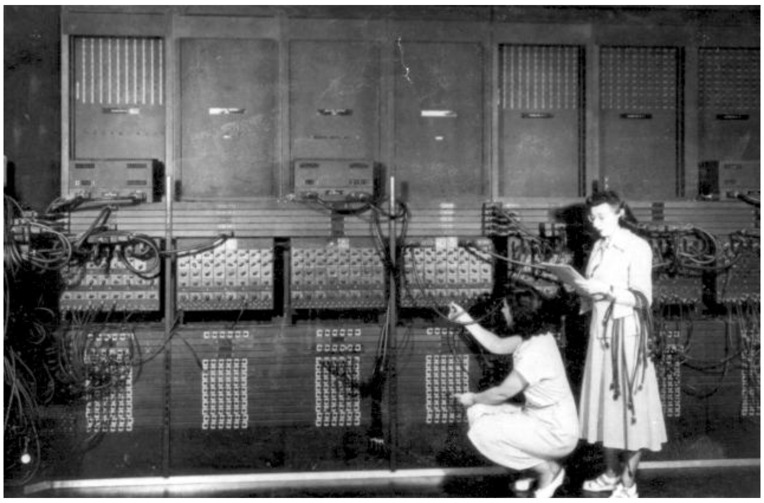
Dialectica Interpretations
A Categorical Analysis

Bodil Biering



Ph.D. Dissertation
Programming, Logic, and Semantics Group
IT University of Copenhagen

Photo on front page: *Two women wiring the right side of the ENIAC with a new program, in the "pre-von Neumann" days. "U.S. Army Photo" from the archives of the ARL Technical Library. Standing: Ester Gerston Crouching: Gloria Ruth Gorden*. Source: http://ftp.arl.army.mil/ftp/historic-computers/

Abstract

The work presented in this thesis is a contribution to the area of type theory and semantics for programming languages in that we develop and study new models for type theories and programming logics. It is also a contribution to the area of logic in computer science, in that our categorical analysis provides us with new insights into functional interpretations. Functional interpretations have proved highly effective in the area of proof mining, which is the enterprise of extracting constructive (computable) contents from non-constructive proofs.

When Gödel published his functional interpretation in the journal *Dialectica*, hence the name "Dialectica Interpretation", in 1958, it was as a contribution to Hilbert's program. The Dialectica interpretation reduces consistency of Heyting arithmetic (and combined with the double negation translation, also Peano arithmetic) to consistency of Gödel's system T, a quantifier-free theory of computable finite-type functionals. Nowadays, fifty years later, the interest in Gödel's functional interpretation as well as other functional interpretations (e.g. Kleene's number realizability, Kreisel's modified realizability, the Diller-Nahm interpretation) is much less philosophical, and much more oriented toward (theoretical) computer science.

The Dialectica interpretations are remarkable syntactic constructions, We use these constructions to develop new mathematical structures such as the *Dialectica categories*, the *Dialectica-* and *Diller-Nahm triposes*, and the *Dialectica-* and *Diller-Nahm toposes*. The benefits work in both directions. The mathematical structures created from the functional interpretations provides us with new models for type theories and programming logics. And from studying the mathematical structures we also gain new insights into the syntactical constructions, in particular we present a new Dialectica variant for higher typed Heyting arithmetic: the *Copenhagen interpretation*, which is a product of the categorical analysis of the original Dialectica interpretation.

Preface

This Ph.D. dissertation is a collection of papers, preprints and technical reports that document my research at the IT-University (2004–2008) and during my stay at Cambridge University (Spring, 2006). Each of the papers, preprints or technical reports is preceded by a short declaration that summarizes the current status (published, accepted or submitted), contributions of the authors if more than one author, main results and relation to other work. To help the reader, each declaration also contains a short list of references for background material. It is the overall assumption that the reader is familiar with the basics of category theory, logic, and categorical logic.
I would like to start with a few words on the broader context into which this work fits. Let me to quote from [HHI$^+$01]:

> Just as in natural sciences, mathematics has been highly effective in computer science. In particular, several areas of mathematics, including linear algebra, number theory, probability theory, graph theory, and combinatorics, have been instrumental in the development of computer science. Unlike the natural sciences, however, computer science has also benefited from an extensive and continuous interaction with logic. As a matter of fact, logic has turned out to be significantly more effective in computer science than it has been in mathematics. This is quite remarkable, especially since much of the impetus for the development of logic during the past one hundred years came from mathematics.
>
> Indeed, let us recall that to a large extent mathematical logic was developed in an attempt to confront the crisis in the foundations of mathematics that emerged around the turn of the 20th Century. Between 1900 and 1930, this development was spearheaded by Hilbert's Program, whose main aim was to formalize all of mathematics and establish that mathematics is *complete* and *decidable*. Informally, completeness means that all "true" mathematical statements can be "proved", whereas decidability means that there is a mechanical rule to determine whether a given mathematical statement is "true" or "false". Hilbert firmly believed that these ambitious goals could be achieved. Nonetheless, Hilbert's Program was dealt devastating blows during the 1930's. Indeed, the standard first-order axioms of arithmetic were shown to be incomplete by Gödel in his celebrated 1931 paper [Göd06]. [...]
>
> [L]ogic has permeated through computer science during the past thirty years much more than it has through mathematics during the past hundred years. Indeed, at present concepts and methods of logic occupy a central place in computer science, insomuch that logic has been called "the calculus of computer science" [MW85].

And from the same source we get a nice description of type theory in programming language research:

> In the 1980's and 1990's the study of programming languages was revolutionized by a remarkable confluence of ideas from mathematical and philosophical logic and theoretical computer science. Type theory emerged as a unifying conceptual framework for the design, analysis, and implementation of programming languages. Type theory helps to clarify subtle concepts such as data abstraction, polymorphism, and inheritance. It provides a foundation for developing logics of program behavior that are essential for reasoning about programs. It suggests new techniques for implementing compilers that improve the efficiency and integrity of generated code.

The work presented in this thesis is a contribution to the area of type theory and semantics for programming languages in that we develop and study new models for type theories and programming logics. It is also a contribution to the area of logic in computer science, in that our categorical analysis provides us with new insights into functional interpretations. Functional interpretations have proved highly effective in the area of proof mining. Proof mining is the enterprise of extracting constructive (computable) contents from non-constructive proofs, see e.g. the survey papers [KO03], [Koh].

When Gödel finally[1] published his functional interpretation in the journal *Dialectica*, hence the name "Dialectica Interpretation", in 1958, it was as a contribution to Hilbert's program. The Dialectica interpretation reduces consistency of Heyting arithmetic (and combined with the double negation translation, also Peano arithmetic) to consistency of Gödel's system T, a quantifier-free theory of computable finite-type functionals. Nowadays, fifty years later, the interest in Gödel's functional interpretation as well as other functional interpretations (e.g. Kleene's number realizability, Kreisel's modified realizability, the Diller-Nahm interpretation) is much less philosophical, and much more oriented toward (theoretical) computer science.

The Effective Topos [Hyl82] is a categorical structure which is built from Kleene's number realizability. The benefits of this structure falls into two: new insights related to the source of origin - in this case Kleene's number realizability, and new categorical models for type theory and logic. In the case of the Effective topos these benefits include a higher order version of Kleene's number realizability, and a whole range of models for dependent type theory. For a text book presentation see [Jac99].

Thus inspired we turn to another, and somewhat more complex functional interpretation - the Dialectica interpretation, to attack it with the powerful, modern tool that Gödel and his contemporaries would have wished they had: category theory. I believe the first attack was carried out by Valeria de Paiva, who defined and studied the Dialectica categories in [dP91, dP89] when she was a student of Martin Hyland. Given sufficient conditions on the base category $\mathbb{C}$, the Dialectica category $\mathrm{Dial}(\mathbb{C})$ is symmetric monoidal closed with products and weak coproducts. No Cartesian closed structure was found. In her thesis [dP91], de Paiva also makes the connection between the Diller-Nahm interpretation [DN74] and the Dialectica interpretation via a Girardian comonad ! (i.e., it satisfies $!(A \times B) \cong !A \otimes !B$), achieving a class of categorical models of Girard's linear logic with modality. Valeria de Paiva also introduced the Girard categories, and showed that they are symmetric, monoidal closed and have finite products and coproduct. The Kleisli category for the comonad ! corresponds to the Diller-Nahm interpretation. This category is Cartesian closed because the comonad is Girardian.

The Dialectica categories by de Paiva were studied for the subobject fibration only, and in [Hyl02] this was taken a step further to preordered fibrations. There are (at least) three different approaches to obtaining Cartesian closed Dialectica categories. The natural structure of the category $\mathrm{Dial}(p)$ is symmetric monoidal closed with finite products. One way to obtain Cartesian closure is by adding structure that will make $\otimes$ a product, that is, making sure we get projections and diagonals for $\otimes$. This approach has been studied briefly in [Hyl02]. Another way of obtaining Cartesian closed Dialectica categories is by altering the definition slightly to get variations like the Diller-Nahm Dialectica category (see [dP91]). We show in Chapter 3 that there are several variants constructed in the same manner as the Diller-Nahm category, that is, by a Girardian comonad on a Dialectica category, or on a Girard category. The third approach that one might think of is to add enough structure to define an exponent (without making $\otimes = \times$). The paper in Chapter 4 is devoted to studying this approach.

Combining the ideas of the Effective Topos with that of Dialectica categories, we get the Dialectica topos (and tripos). The Dialectica tripos was first described by Lars Birkedal, following ideas of Martin Hyland, in an unpublished note. This note was later merged with another unpublished note by Thomas Streicher, and after thorough revision and addition of material, this resulted in the preprint in Chapter 2.

The exponent construction, originally defined for the Dialectica tripos, is examined in the paper in Chapter 4. The analysis there shows that what we are really looking at, is a variant of the Dialectica categories, namely the Kleisli category of a certain non-Girardian comonad on a Dialectica category. This Dialectica-Kleisli category is weakly Cartesian closed. Thus, both the preordered reflection and the Cauchy completion are Cartesian closed variants of Dialectica categories.

[1]"The ideas in this paper date back at least as far as 1941, since Gödel lectured at that time on his interpretation at Princeton and Yale." From [Göd90]

Outline: In Chapter 2 we present four new triposes reflecting as much as possible of the Dialectica interpretation, which we call the *Dialectica tripos* and denote by d. The resulting topos is denoted by $\mathbf{Dia}$. From the tripos d we get a closed subtripos, the *modified Dialectica tripos*, denoted by d_m, and the resulting topos is denoted by $\mathbf{Dia}_m$. We also define a tripos reflecting as much as possible of the Diller-Nahm interpretation, which we call the *Diller-Nahm tripos*, denoted dn. The resulting topos is denoted by $\mathbf{DN}$. From the tripos dn we get a closed subtripos, the *modified Diller-Nahm tripos*, denoted by dn_m, and the resulting topos is denoted by $\mathbf{DN}_m$. The modified versions are in closer correspondence with the standard interpretations of Dialectica, respectively Diller-Nahm, since "modified" corresponds to having non-empty types. We give an account of the first order logic of the toposes and find that first order logic of dn_m corresponds to the Diller-Nahm interpretation, and that first order logic of $\mathbf{Dia}_m$ does not correspond to the Dialectica interpretation, but instead to a variant of Dialectica, which we call the Copenhagen interpretation. This is perhaps not so surprising when we recall that the Dialectica interpretation assumes that atomic formulas are decidable, and that there is no such restriction for the tripos. Though we have some nice results regarding the decidable fragment of predicates over the natural numbers in $\mathbf{Dia}_m$, we argue that it is not possible to interpret first order logic with decidable atomic formulas in this fragment. Hence we do not find a correspondence between first order logic with decidable atomic formulas in $\mathbf{Dia}_m$ and the Dialectica interpretation. The tripos setting allows us to reveal many new relations in the form of geometric morphisms to other functional interpretations, which are also represented by triposes. The relations can be summed up in diagrams:

At the tripos level, we get the following diagram of fibred adjunctions, where, however, only some give rise to geometric morphisms:

$$\begin{tikzcd}
 & \mathsf{dn}_m \arrow[rr, shift left, "!"] \arrow[d, shift left, "q_*"] & & \mathsf{d}_m \arrow[ll, shift left, "id"] \arrow[rr, shift left, "F"] \arrow[d, shift left, "H_*"] \arrow[dll, shift left, "p_*"] & & \mathbf{Set} \arrow[ll, shift left, "\nabla"] \\
 & \mathsf{m} \arrow[u, shift left, "q^*"] \arrow[rr, shift left, "v_*"] \arrow[urr, shift left, "p^*"] \arrow[dl, shift left, "\Gamma"] & & \mathsf{e} \arrow[u, shift left, "H^*"] \arrow[ll, shift left, "v^*"] \arrow[rr, shift left, "i_*"] \arrow[dl, shift left, "\Gamma"] \arrow[dr, shift left, "j^*"] & & \mathsf{dn} \arrow[ll, shift left, "i^*"] \\
\mathbf{Set} \arrow[ur, shift left, "\Delta"] \arrow[rr, "\cong"] & & \mathbf{Set} \arrow[ur, shift left, "\Delta"] & & \mathsf{d} \arrow[ul, shift left, "j_*"] &
\end{tikzcd}$$

Here H, v, and q all are connected geometric morphisms, so they lift to surjective geometric morphisms on the induced toposes, and i and j are open geometric inclusions, so they lift to open geometric inclusions. The left adjoints of the adjunctions $! \dashv id$, $\nabla \dashv F$, and $p^* \dashv p_*$ are full and faithful. At the topos level we get the following geometric morphisms

$$\begin{tikzcd}
 & \mathbf{DN}_m \arrow[d, "q"] & & \mathbf{Dia}_m \arrow[d, "H"] & \\
 & \mathbf{Mod} \arrow[rr, "v"] & & \mathbf{Eff} \arrow[rr, hook, "i"] \arrow[dr, hook, "j"] & & \mathbf{DN} \\
\mathbf{Mod}_{\neg\neg} \arrow[ur, hook] \arrow[rr, "\cong"] & & \mathbf{Eff}_{\neg\neg} \arrow[ur, hook] & & \mathbf{Dia} &
\end{tikzcd}$$

with i, j open inclusions.

In Chapter 3 we develop a uniform way to relate the triposes, namely via comonads on a Girard category. Independently, a similar unifying framework is presented in [Oli08], but in a syntactic setting. If the comonad is Girardian, then we have a result stating that the Kleisli category is a tripos. In Chapter 3, we have collected the results in the following neat table, which we will explain properly there:

G			G_m		
comonad	Kleisli	$L(p \wedge q) \equiv Lp \otimes Lq$	comonad	Kleisli	$K(p \wedge q) \equiv Kp \otimes Kq$
L_{d}	d	No	$K_{\mathsf{d}_m} = L_{\mathsf{d}}$	d_m	No
L_s	$\mathbf{Set}(-,2)$	Yes	$K_s = L_s$	$\mathbf{Set}(-,2)$	Yes
L_{dn}	dn	Yes	$K_{\mathsf{dn}_m} = L_{\mathsf{dn}}$	dn_m	Yes
L_{e}	e	Yes	K_{e}	e	Yes
L_{e_2}	e_2	Yes	$K_{\mathsf{m}} = L_{\mathsf{e}_2}$	m	Yes

The Girard category together with a Girardian comonad yields a model of linear logic with modality, so we get a collection of realizability models for linear logic with modality.

The construction from a fibration $p : \mathcal{E} \to \mathcal{T}$ to its Dialectica category $\mathrm{Dial}(p)$ contains an implicit soundness proof. Informally, if for example $\mathrm{Dial}(p)$ (as a preorder) carries the structure of a Heyting algebra, then $\mathbf{HA} \vdash \phi$ implies $\mathcal{L}(p) \vdash \phi^D$, where $\mathcal{L}(p)$ is the internal logic of the fibration p and ϕ^D is the Dialectica interpreted formula. With the intention of making the ideas available outside the categories community, this soundness proof has been made explicit in Chapter 5, giving rise to the Copenhagen interpretation. The Copenhagen interpretation is a generalization of Dialectica which is not limited to decidable atomic formulas, thus it soundly interprets higher typed Heyting arithmetic, $\mathbf{HA}^\omega$, whereas the original Dialectica interpretation only interprets first order Heyting arithmetic, $\mathbf{HA}$. Generalizing the Dialectica interpretation to higher types has been one of the aims for our research. With the Copenhagen interpretation, we reach this goal, though it is perhaps not as simple as one would have liked. Very recently, we have discovered a much simpler variant, which we believe also interprets $\mathbf{HA}^\omega$. This new variant will be presented in a future paper, but we give a rough sketch of the idea in an appendix of the paper in Chapter 5. The Copenhagen interpretation is the direct result of the categorical analysis of the Dialectica and Diller-Nahm interpretations in [dP89, Hyl02], and the papers in Chapters 2 and 4. The basic structure was discovered during the research for the preprint in Chapter 2 and refined by Martin Hyland at a meeting in Copenhagen[2] in 2006, hence the name "Copenhagen interpretation". A thorough analysis of the clause for implication can be found in Chapter 4.

Though chronologically it came first, the paper on BI-hyperdoctrines in 6 is placed at the very end of the dissertation, because the story, though related, is somewhat separate from the rest. In Chapter 6 we present a precise correspondence between separation logic (for an introduction see [Rey02] and the references therein) and a simple notion of higher order predicate BI (logic of bunched implications), extending an earlier correspondence given between propositional separation logic and propositional BI. Moreover, we introduce the notion of a BI hyperdoctrine, show that it soundly models classical and intuitionistic first- and higher-order predicate BI, and use it to show that we may easily extend separation logic to *higher order*. We also show that the "canonical guess" for a model of higher order separation logic, namely a topos, though it *is* a BI hyperdoctrine, it is a trivial one in the sense that the monoidal structure $(\otimes, \multimap)$ coincides with the intuitionistic structure $(\wedge, \rightarrow)$. The extension of separation logic to higher order has proved very useful for modular reasoning (data abstraction) in [Bir07a, Bir07b, Bie08] and for formalization of separation logic in [Bir08].

Indications for future work are given in the chapters they belong to. All together this dissertation comprises the following:

1. B. Biering, *Extended Introduction to Topos Theoretic Versions of Dialectica Interpretations*, Unpublished manuscript.

2. B. Biering, L. Birkedal, C. Butz, J.M.E. Hyland, J. van Oosten, G. Rosolini, T. Streicher, *Topos Theoretic Versions of Dialectica Interpretations*, Preprint for publication.

3. B. Biering, *A Unified View on the Dialectica Triposes*, Unpublished manuscript.

4. B. Biering, *Cartesian Closed Dialectica Categories*, Submitted for publication in Annals of Pure and Applied Logic.

[2]The authors of [BBLBCB07] have held two "Dialectica meetings" the first in Copenhagen in September 2006, the second in Genoa, June 2007

5. B. Biering, *The Copenhagen Interpretation*, Submitted for publication in Annals of Pure and Applied Logic.

6. B. Biering, L. Birkedal, N. Torp-Smith, *BI Hyperdoctrines and Higher Order Separation Logic*, published in ACM Transactions on Programming Languages and Systems, Volume 29, Issue 5, Article 24 (2007), Special Issue ESOP'05.

Acknowledgements

I am indebted to my advisor, Lars Birkedal, who has been the driving force in creating an attractive research environment in the PLS group at ITU, with many PhD students, lots of activities and lots of international visitors. Lars is always interested and encouraging, he is generous with his time and keeps his door open, he has pointed me in fruitful directions and helped me establish good contacts around the world. One of these contacts is Martin Hyland, who kindly hosted my five month visit in Cambridge during my PhD studies. Anyone who knows Martin also knows that he is very generous with sharing his ideas and patient when explaining them, and he certainly made my visit to Cambridge worth while!

The most rewarding single experiences has probably been the two “Dialectica meetings”; the first in Copenhagen, the second in Genoa, with Lars Birkedal, Carsten Butz, Martin Hyland, Jaap van Oosten, Pino Rosolini, Thomas Streicher, and myself. I have benefited greatly from discussions and correspondences with all of the above, and we also had a lot of fun!

I should also like to thank Mads Tofte (principal of ITU) and everyone else who took part in starting the IT-University in 1999, it is a unique and inspiring environment to be part of, and I am proud to be able to say that I did my my PhD studies here.

Thanks to all of my colleagues in the PLS group for nice company, good discussions about mathematics, computer science, life, and everything in between, especially the latter. In particular I would like to thank Rasmus Møgelberg, Søren Debois and Noah Torp-Smith, for giving constructive input, and reading and commenting on parts of the material. Mikkel Bundgaard for computer support (I am a *theoretical* computer scientist, Mikkel, and I know quite well what a prime number is, thank you). Troels Damgaard for making the work environment more glamorous with his MTV hair.

Last, but not least, I am grateful to my wonderful husband for a lot of things which are probably not appropriate to mention here, but among those which can be mentioned: Thanks for making sure that everything always work smoothly at home (and I am not talking about the coffeemaker). And of course, our two lovely sons - never a dull moment.

References

[BBLBCB07] J.M.E. Hyland & J. van Oosten & G. Rosolini & T. Streicher B. Biering & L. Birkedal & C. Butz. Topos theoretic versions of Dialectica interpretations. 2007. Unpublished draft.

[Bie08] M. Parkinson & G. Biermann. Separation logic, abstraction and inheritance. *Proc. 35th POPL*, 2008.

[Bir07a] A. Nanevski & A. Ahmed & G. Morrisett & L. Birkedal. Abstract predicates and mutable adts in hoare type theory. *Proc. ESOP'07, LNCS 4421*, pages 189–204, 2007.

[Bir07b] N. Krishnaswami & J. Aldrich & L. Birkedal. Modular verification of the subject-observer pattern via higher-order separation logic. *9th Workshop on Formal Techniques for Java-like Programs (FTfJP 207)*, 2007.

[Bir08] C. Varming & L. Birkedal. Higher-order separation logic in isabelle/holcf. *Submitted for publication*, 2008.

[DN74] Justus Diller and Werner Nahm. Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen. *Arch. Math. Logik Grundlagenforsch.*, 16:49–66, 1974.

[dP89] V. C. V. de Paiva. The Dialectica categories. In *Categories in computer science and logic (Boulder, CO, 1987)*, volume 92 of *Contemp. Math.*, pages 47–62. Amer. Math. Soc., Providence, RI, 1989.

[dP91] V.C.V de Paiva. *The Dialectica Categories, Technical Report 213 from Computer Laboratory (Thesis)*. PhD thesis, University of Cambridge, 1991.

[Göd90] Kurt Gödel. *Collected works. Vol. II.* The Clarendon Press Oxford University Press, New York, 1990. Publications 1938–1974, Edited and with a preface by Solomon Feferman, pages 217–251.

[Göd06] Kurt Gödel. Über formal unentscheidbare Sätze der *principia mathematica* und verwandter Systeme. I. *Monatsh. Math.*, 149(1):1–30, 2006. Reprinted from Monatsh. Math. Phys. **38** (1931), 173–198 [MR1549910], With an introduction by Sy-David Friedman.

[HHI$^+$01] Joseph Y. Halpern, Robert Harper, Neil Immerman, Phokion G. Kolaitis, Moshe Y. Vardi, and Victor Vianu. On the unusual effectiveness of logic in computer science. *Bull. Symbolic Logic*, 7(2):213–236, 2001.

[Hyl82] J. M. E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981)*, volume 110 of *Stud. Logic Foundations Math.*, pages 165–216. North-Holland, Amsterdam, 1982.

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

[Jac99] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

[KO03] U. Kohlenbach and P. Oliva. Proof mining: a systematic way of analyzing proofs in mathematics. *Tr. Mat. Inst. Steklova*, 242(Mat. Logika i Algebra):147–175, 2003.

[Koh] U. Kohlenbach. Gödel's functional interpretation and its use in current mathematics. *To appear in: Horizons of Truth, Gödel Centenary. Cambridge University Press.*

[MW85] Zohar Manna and Richard Waldinger. *The logical basis for computer programming. Vol. I.* Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1985. Deductive reasoning.

[Oli08] Paulo Oliva. Functional interpretations of linear and intuitionistic logic. *special issue of Information and Computation*, 2008.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. 2002.

Contents

Chapter 1

Extended Introduction to Topos Theoretic Versions of Dialectica Interpretations

This note contains some background material for the paper in Chapter 3. It is mostly material which is folklore, but can be hard to find written accounts of. There are also a few new results including: A connected geometric morphism of triposes lifts to a surjection of toposes, and any geometric morphism of triposes factors uniquely as an inclusion followed by a connected geometric morphism.

The reader should be familiar with basic tripos theory (see [Pit02, HJP80, Pit81]) and have some knowledge about toposes and j-topologies (see e.g. [MLM94, Joh02]).

References

[HJP80] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980.

[Joh02] Peter T. Johnstone. *Sketches of an elephant: a topos theory compendium.*, volume 44 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, Oxford, 2002.

[MLM94] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic*. Universitext. Springer-Verlag, New York, 1994. A first introduction to topos theory, Corrected reprint of the 1992 edition.

[Pit81] A.M. Pitts. *The Theory of Triposes*. PhD thesis, Cambridge University, 1981.

[Pit02] Andrew M. Pitts. Tripos theory in retrospect. *Math. Structures Comput. Sci.*, 12(3):265–279, 2002. Realizability (Trento, 1999).

Extended Introduction to Topos Theoretic Versions of Dialectica Interpretations

Bodil Biering

In this note we have collected some background material for the paper in [Bie08, Chapter 3]. It is mostly material which is folklore, but can be hard to find written accounts of. Some of it exists in the literature (when that is the case, we give references), and some of the results are new.

1 Triposes and Geometric Morphisms of Triposes

In this section we recall some definitions and standard results, and we show a new result: that the lifting of a connected geometric morphism of triposes results in a surjection of toposes.

The Logic of the Topos $\mathbb{C}[P]$ Reduced to the Logic of the Tripos P

Definition 1.1. *For an object $(X, \sim)$ of the topos $\mathbb{C}[P]$ a* strict predicate *on $(X, \sim)$ is a predicate $A \in P(X)$ which satisfies*

$$A(x),\, x \sim x' \vdash A(x') \quad \textit{and} \quad A(x) \vdash x \sim x.$$

Proposition 1.2. *For each object $(X, \sim)$, there is an isomorphism between strict predicates on $(X, \sim)$ and subobjects of $(X, \sim)$ in $\mathbb{C}[P]$.*

For details see [HJP80, Pit81].

Now assume that subobjects are represented by strict predicates. If we mark the connectives of the topos with $\tilde{\cdot}$, we can express the logic of $\mathrm{Sub}_{\mathbb{C}[P]}(X, \sim)$ in terms of the tripos logic of $P(X)$ in the following way:

- Propositional connectives: $\tilde{\bot} = \bot$, $\tilde{\vee} = \vee$, $\tilde{\top} = x \sim x$, $\tilde{\wedge} = \wedge$, $A \tilde{\to} B = (x \sim x) \wedge (A \to B)$.
- Quantifiers: For a morphism $F : (X, \sim) \to (Y, \sim)$,

$$\begin{array}{rcl} \tilde{\exists}_F(A)(y) & = & \exists x : X.F(x,y) \wedge A(x) \\ \tilde{\forall}_F(A)(y) & = & y \sim y \wedge (\forall x : X.F(x,y) \to A(x)). \end{array}$$

In the special case where F is a projection $(X,\sim) \times (J,\sim) \to (J,\sim)$ we get

$$\begin{array}{rcl} \tilde{\exists} x : X.A(x,j) & = & \exists x : X.A(x,j) \\ \tilde{\forall} x : X.A(x,j) & = & j \sim j \wedge (\forall x : X.(x \sim x) \to A(x,j)). \end{array}$$

Example 1.3. *We now look at how a decidable subobject in a topos $\mathbb{C}[P]$ is expressed in the logic of P. Let $A \in P(X)$ be a strict predicate over $(X,\sim)$. We want to know what is means in terms of tripos logic that A is decidable in $\mathbb{C}[P]$, i.e., that*

$$\mathbb{C}[P] \models A(x) \vee \neg A(x).$$

Using the translation into tripos logic that we gave above, this reads

$$x \sim x \vdash A(x) \vee (x \sim x \wedge (A(x) \to \bot))$$

in $P(X)$. Since $A(x)$ is strict, this is equivalent to

$$x \sim x \vdash (A(x) \vee \neg A(x)) \wedge x \sim x$$

and clearly

$$\begin{array}{ll} x \sim x \vdash (A(x) \vee \neg A(x)) \wedge x \sim x & \textit{iff} \\ x \sim x \vdash A(x) \vee \neg A(x). & \end{array}$$

Definition 1.4. *Let $\mathbb{C}$ be a finitely complete category, and let P and Q be triposes over $\mathbb{C}$. A* geometric morphism *$f : P \to Q$ is a pair of fibred functors (f^*, f_*) over $\mathbb{C}$, with $f^* : Q \to P$ and $f_* : P \to Q$ such that f^* is fibred left adjoint to f_*, and f^* preserves fibred finite limits.*

Definition 1.5. *A* connected geometric morphism *is a geometric morphism $f : P \to Q$ with the property that f^* is full and faithful, or, equivalently, the unit $\eta : id \Rightarrow f_* f^*$ is iso.*

Definition 1.6. *A* geometric inclusion *is a geometric morphism $i : P \to Q$ with the property that i_* is full and faithful, or, equivalently, the counit $\epsilon : f^* f_* \Rightarrow id$ is iso.*

Both f_* and f^* preserve finite limits so if $\sim \in P(X \times X)$ is a partial equivalence relation then so is $f_*(\sim) \in Q(X \times X)$ and likewise for f^*. Since the left adjoint f^* preserves $\exists$ as well, it preserves all the properties of being a functional relation, so if $F \in Q(X \times Y)$ is a functional relation, then $f^*(F) \in P(X \times Y)$ is a functional relation. The right adjoint does not necessarily preserve $\exists$ so if we apply f_* to a functional relation F, this will in general only result in a partial functional relation $f_*(F)$. However, there are objects $(Y,\sim)$ with the property that if $F : (X,\sim) \to (Y,\sim)$ is a functional relation in P, then $f_*(F)$ is a functional relation in Q. These are called weakly complete.

Definition 1.7. *An object $(Y,\sim)$ of a topos $\mathbb{C}[P]$ is called* weakly complete *if given a partial function $F : (X,\sim) \to (Y,\sim)$, there is a morphism $f : X \to Y$ in $\mathbb{C}$, such that*

$$P \models \exists y : Y.F(x,y) \leftrightarrow F(x,fx).$$

It was shown in [Pit81] that any object of $\mathbb{C}[P]$ is isomorphic to a weakly complete one. We give a detailed proof here (in Proposition 1.9) since we shall make use of it for proving Theorem 1.15. First recall the definition of the membership predicate which is part of the tripos definition:

For every object X of $\mathbb{C}$ there is an object $\pi(X)$ of $\mathbb{C}$ and an element $\in_X$ of $P(X \times \pi(X))$ with the following property: For every object Y of $\mathbb{C}$ and every element ϕ of $P(X \times Y)$, there is a morphism $\{\phi\} : Y \to \pi(X)$ in $\mathbb{C}$ such that in $P(X \times Y)$, ϕ is isomorphic to $x \in_X \{\phi\}$.

The object $(\pi(X), \sim_S)$ is defined as follows: $\pi(X)$ is the carrier object for the membership predicate, and the partial equality relation in $(\pi(X), \sim_S)$ is defined by

$$S(U) \equiv \exists x.\, x \sim x \wedge \forall x'.(x' \in_X U \leftrightarrow x \sim x')$$

and

$$U \sim_S V \equiv S(U) \wedge \forall x.(x \in_X U \leftrightarrow x \in_X V).$$

$(\pi(X), \sim_S)$ is the object of singletons with respect to $\sim_X$, that is, equivalence classes of X.

Proposition 1.8. *The object* $(\pi(X), \sim_S)$ *is weakly complete.*

For a detailed proof of this see [vO08].

Proposition 1.9. *The object* $(\pi(X), \sim_S)$ *is isomorphic to* $(X, \sim)$.

Proof: The object $(\pi(X), \sim_S)$ is isomorphic to $(X, \sim)$, the isomorphism is given by

$$K(x, U) \equiv x \sim x \wedge S(U) \wedge x \in_X U$$

and K has itself as inverse.

On arrows $F : (X, \sim) \to (Y, \sim)$ we first compose to get the map:

$$(\pi(X), \sim_S) \xrightarrow{\sim} (X, \sim) \xrightarrow{F} (Y, \sim) \xrightarrow{\sim} (\pi(X), \sim_S)$$

and then apply f_* to the composite.

We give the proof that K is a functional relation and $K^{-1} = K$. Clearly we have:

Strict:

$$K(x, U) \vdash x \sim x \wedge U \sim_S U$$

Relational:

$$x \sim x' \wedge U \sim_S V \wedge K(x, U) \vdash x' \sim x' \wedge S(V) \wedge x' \in_X V \equiv K(x', V)$$

Single valued:

$$K(x, U) \wedge K(x, V) \vdash U \sim_S V$$

Total: To show that

$$\vdash x \sim x \to \exists U.K(x, U)$$

we use the properties of the membership predicate: For $\sim \in P(X \times X)$ there is an arrow $\{\phi\} : X \to \pi(X)$ in $\mathbb{C}$ such that in $P(X \times X)$ we have

$$x \sim x' \dashv\vdash x \in_X \{\phi\}(x')$$

so we can put $U = \{\phi\}(x)$.

Next we show $K^{-1} = K$. We must prove that

$$x \sim x \equiv \exists U.K(x, U) \wedge K(x, U)$$

Clearly

$$\exists U.K(x, U) \wedge K(x, U) \equiv \exists U.K(x, U) \vdash x \sim x$$

and

$$x \sim x \vdash \exists U.K(x, U)$$

follows from K being total. We must also show that

$$U \sim_S U \equiv \exists x.K(x, U) \wedge K(x, U)$$

again the direction from right to left is clear, to see that

$$U \sim_S U \vdash \exists x.x \sim x \wedge S(U) \wedge x \in_X U$$

just unfold the definition of $U \sim_S U$. $\square$

Proposition 1.10. *Any geometric morphism $f = (f^*, f_*) : P \to Q$ of triposes, can be lifted to a geometric morphism $\bar{f} = (\bar{f}^*, \bar{f}_*) : \mathbb{C}[P] \to \mathbb{C}[Q]$ of toposes.*

For a proof of this see [HJP80, Pit81] or [vO08], we give the definition here for convenience. Suppose $F : (X, \sim) \to (Y, \sim)$ is a map in $\mathbb{C}[Q]$, then $\bar{f}^*$ is simply

$$f^*(F) : (X, f^*(\sim)) \to (Y, f^*(\sim)).$$

On objects $(X, \sim)$, the functor $\bar{f}_*$ is defined as $(\pi(X), f_*(\sim_S))$. If $G : (Y, \sim) \to (X, \sim)$, let G' represent the composite map

$$\begin{tikzcd}(\pi(X), \sim_S) \arrow[r, "\sim"] & (X, \sim) \arrow[r, "G"] & (Y, \sim) \arrow[r, "\sim"] & (\pi(Y), \sim_S)\end{tikzcd}$$

then $\bar{f}_*(G) = f_*(G')$.

Proposition 1.11. *If $f : P \to Q$ is an inclusion of triposes, then $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ is an inclusion of toposes.*

Proof: Proof sketch: We have

$$\bar{f}^*\bar{f}_*(X, \sim) = \bar{f}^*(\pi(X), f_*(\sim_S)) = (\pi(X), f^*f_*(\sim_S)) \cong (\pi(X), \sim_S) \cong (X, \sim).$$

$\square$

The following example shows that if $f : P \to Q$ is a connected geometric morphism of triposes, then it is *not* necessarily the case that $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ is connected.

Example 1.12. [1] *Consider the* **Set***-triposes* **Set**$(-,2)$ *and* **Set**$(-,2\times 2)$*. There is a connected geometric morphism* $\delta \dashv \wedge$ *from* **Set**$(-,2\times 2)$ *to* **Set**$(-,2)$*, where* δ *is the diagonal. The induced geometric morphism on the topos level is* $\Delta \dashv P$ *from* **Set** $\times$ **Set** $\to$ **Set** *is given by*

$$\Delta(A) = (A, A) \qquad P(A,B) = A \times B$$

with unit $\eta_A = \delta_A = A \mapsto A \times A$*, which is not an iso. see [Fre07].*

Next we show a result regarding the lifting from tripos to topos of a connected geometric morphism. We have seen that any geometric morphism $f : P \to Q$ of $\mathbb{C}$-triposes P and Q lifts to a geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$, and also that if f is an inclusion, then so is $\bar{f}$.

As Example 1.12 demonstrates, it is not in general the case that a connected geometric morphism of triposes lifts to a connected geometric morphism of toposes. In his thesis [Bir99], Lars Birkedal shows that if f is connected and if f_* preserves $\exists$, then $\bar{f}$ is also connected. The result that we show in this section is a "local" version of this result, in the sense that if f_* only preserves $\exists_h$ for certain morphisms h, then f^* is fully faithful for certain homsets.

Definition 1.13. *Let* $\mathbb{C}, \mathbb{D}$ *be categories and* X, Y *objects of* $\mathbb{C}$*. We say that a functor* $H : \mathbb{C} \to \mathbb{D}$ *is fully faithful on* (X,Y) *if there is a bijection* $\mathrm{Hom}_{\mathbb{C}}(X,Y) \cong \mathrm{Hom}_{\mathbb{D}}(HX, HY)$*.*

We now proof a local version of a well known fact.

Lemma 1.14. *Suppose* $H = (H^*, H_*)$ *is an adjunction. If the component* $\eta_Y : Y \to H_*H^*(Y)$ *on* Y *of the unit* η *is an iso, then* H^* *is fully faithful on* (X,Y) *for all objects* X*.*

Proof. Let $F : H^*(X) \to H^*(Y)$, we are going to define $F' : X \to Y$ such that $H^*(F') = F$. We define F' as the composite

$$\begin{tikzcd} X \arrow[r, "\eta_X"] & H_*H^*(X) \arrow[r, "H^*(F)"] & H_*H^*(Y) \arrow[r, "\eta_Y^{-1}"] & Y \end{tikzcd}$$

If we can show that $H^*(\eta_Y^{-1}) = \epsilon_{H^*(Y)}$ it follows from the universal properties of adjunctions that $H^*(F') = F$. By the triangle equalities we have

$$\epsilon_{H^*Y} \circ H^*(\eta_Y) = id_{H^*Y}$$

so $H^*(\eta_Y)^{-1} = \epsilon_{H^*Y}$, and since $H^*(\eta_Y)^{-1} = H^*(\eta_Y^{-1})$, we are done. On the other hand let $G : X \to Y$, we show that $(H^*G)' = G$. $(H^*G)' = \eta_Y^{-1} \circ H_*H^*G \circ \eta_X$, and by naturality of η, we have $\eta_Y \circ G = H_*H^*G \circ \eta_X$. We have shown that H^* and $(-)'$ are inverses. □

[1] I am happy to thank Jonas Frey for providing this counter example.

Theorem 1.15. *Suppose $f : P \to Q$ is a connected geometric morphism of triposes and let $\pi : X \times J \to J$ be a projection in $\mathbb{C}$ for some fixed object X. Suppose further that for any strict predicate $\phi(x)$ over an object $\bar{f}^*(X,\sim)$, where $(X,\sim)$ is in $\mathbb{C}[Q]$, we have $f_*\exists^P_\pi(\phi(x)) \cong \exists^Q_\pi f_*(\phi(x))$. Then the induced geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ satisfies that $\bar{f}^*$ is fully faithful on $((Y,\sim),(X,\sim))$ for any object $(Y,\sim)$ in $\mathbb{C}[Q]$.*

Proof. We are going to show that the component $\eta_{(X,\sim)} : (X,\sim) \to \bar{f}_*\bar{f}^*(X,\sim)$ of the unit η is iso, then the theorem follows from Lemma 1.14.

First we recall the definition of $\bar{f}$.

$$\bar{f}^*(X,\sim) = (X, f^*(\sim))$$

and if $F : (X,\sim) \to (Y,\sim)$ is a functional relation, then $f^*(F) : (X, f^*(\sim)) \to (Y, f^*(\sim))$ is a functional relation. On objects we have

$$\bar{f}_*(X,\sim) = (\pi(X), f_*(\sim_S))$$

where the partial equality relation in $(\pi(X), \sim_S)$ is defined by

$$S(U) \equiv \exists x.\, x \sim x \wedge \forall x'.(x' \in_X U \leftrightarrow x \sim x')$$

and

$$U \sim_S V \equiv S(U) \wedge \forall x.(x \in_X U \leftrightarrow x \in_X V)$$

Let $K(x,U) \equiv f^*(x \sim x) \wedge S(U) \wedge x \in_X U$. as in Prop. 1.9. Consider f_* applied to the composite

$$\begin{tikzcd}(X, f^*(\sim)) \arrow[r, "id=f^*(\sim)"] & (X, f^*(\sim)) \arrow[r, "K"] & (\pi(X), [f^*(\sim)]_S)\end{tikzcd}$$

which we call $K'(x,U)$, it is defined by

$$\begin{aligned} K'(x,U) &\equiv f_*(\exists x'.f^*(x \sim x') \wedge K(x',U)) \\ &\equiv f_*(f^*(x \sim x) \wedge K(x,U)) \\ &\equiv x \sim x \wedge f_*(K(x,U)) \end{aligned}$$

Now $\eta_{(X,\sim)}$ is K' precomposed with the map $H : (X,\sim) \to (X, f_*f^*(\sim))$ defined by

$$H(x,x') \equiv x \sim x \wedge f_*f^*(x \sim x') \equiv x \sim x'$$

so

$$\eta_{(X,\sim)}(x,U) \equiv x \sim x \wedge f_*(K(x,U)) \equiv f_*(K(x,U)).$$

We are finally ready to show that $\eta_{(X,\sim)}$ is an iso with itself as inverse.

$$\begin{aligned} f_*(K)(x,U) &\equiv f_*(f^*(x \sim x) \wedge x \in_X U \wedge S(U)) \\ &\equiv x \sim x \wedge f_*(x \in_X U) \wedge f_*(\exists x.f^*(x \sim x) \wedge \forall x'.(x' \in_X U \leftrightarrow f^*(x \sim x'))) \\ &\equiv x \sim x \wedge f_*(x \in_X U) \wedge \exists x.x \sim x \wedge \forall x'.f_*(x' \in_X U \leftrightarrow f^*(x \sim x'))) \end{aligned}$$

where we use the fact that $f^*(x \sim x) \wedge \forall x'.(x' \in_X U \leftrightarrow f^*(x \sim x')$ is a strict predicate in $(X, f^*(\sim))$, so that f_* commutes with $\exists$.

Clearly

$$\exists U.f_*(K)(x,U) \vdash x \sim x.$$

Suppose $x \sim x$, put $U = [x]_{f^*(\sim)}$, i.e., $x' \in_X U \equiv f^*(x \sim x')$, then $f_*(x \in_X U) \equiv f_*(f^*(x \sim x)) \equiv x \sim x$, which shows that

$$x \sim x \vdash \exists U.f_*(K)(x,U),$$

which shows that $f_*(K)(x,U) \circ f_*(K)(x,U) \equiv id_{(X,\sim)}$. By definition we have

$$U \sim_S U \equiv f_*(S(U) \wedge \forall x.(x \in_X U \leftrightarrow x \in_X U)) \equiv f_*(S(U))$$

so clearly

$$\exists x.f_*(K)(x,U) \wedge f_*(K)(x,U) \vdash U \sim U.$$

Suppose $U \sim U$ that is

$$\begin{array}{rcl} f_*(S(U)) & \equiv & f_*(\exists x.f^*(x \sim x) \wedge \forall x'.(x' \in_X U \leftrightarrow f^*(x \sim x'))) \\ & \equiv & \exists x.x \sim x \wedge \forall x'.f_*(x' \in_X U \leftrightarrow f^*(x \sim x')); \end{array}$$

this implies

$$\exists x.x \sim x \wedge \forall x'.f_*(x' \in_X U) \leftrightarrow x \sim x',$$

since $f_*(A \to B) \vdash f_*A \to f_*B$, hence

$$x \sim x \wedge f_*(x \in_X U),$$

so we conclude that

$$U \sim_S U \vdash x \sim x \wedge f_*(S(U)) \wedge f_*(x \in_X U).$$

That is,

$$U \sim_S U \vdash \exists x.f_*(K)(x,U),$$

so we have shown that $f_*(K)(x,U) \circ f_*(K)(x,U) \equiv id_{\bar{f}^*\bar{f}_*(X,\sim)}$. □

We have seen that connected geometric morphisms of triposes do not in general lift to connected geometric morphisms of toposes. However, as we show next, a connected geometric morphism of triposes lifts to a surjection of toposes. Later we will show some factorization results that make this property very natural.

Theorem 1.16. *Let $f = (f^*, f_*) : P \to Q$ be a connected geometric morphism of triposes, the induced geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ is then a surjection, i.e., $\bar{f}^*$ is faithful.*

Proof. There are many equivalent definitions of a surjective geometric morphism, we shall show that the unit $\eta_X : X \to \bar{f}_*\bar{f}^*(X)$ in $\mathbb{C}[Q]$ is mono. We must show that for $(X, \sim)$ in $\mathbb{C}[Q]$,

$$Q \models \eta_{(X,\sim)}(x, U) \wedge \eta_{(X,\sim)}(x', U) \to x \sim x'.$$

Recall that

$$\begin{array}{rcl}
\eta_{(X,\sim)}(x, U) & \equiv & f_*(K(x, U)) \\
 & \equiv & f_*(f^*(x \sim x) \wedge x \in_X U \wedge S(U)) \\
 & \equiv & x \sim x \wedge f_*(x \in_X U) \wedge f_*(\exists x.f^*(x \sim x) \wedge \forall x'.(x' \in_X U \leftrightarrow f^*(x \sim x'))) \\
 & \vdash & x \sim x \wedge f_*(x \in_X U) \wedge f_*(f^*(x \sim x)) \wedge \forall x'.f_*((x' \in_X U \leftrightarrow f^*(x \sim x'))) \\
 & \vdash & \forall x'.f_*(x' \in_X U) \to f_*f^*(x \sim x') \qquad \text{since } f_*(A \to B) \vdash f_*A \to f_*B
\end{array} \tag{1}$$

Now for $\eta_{(X,\sim)}(x', U)$ we have

$$\begin{array}{rcl}
\eta_{(X,\sim)}(x', U) & \equiv & f_*(K(x', U)) \\
 & \vdash & f_*(x' \in_X U)
\end{array}$$

together with equation (1) this implies that in the internal logic of Q we have

$$\begin{array}{ll}
 & \eta_{(X,\sim)}(x, U) \wedge \eta_{(X,\sim)}(x', U) \\
\vdash & (\forall y.f_*(y \in_X U) \to f_*f^*(x \sim y)) \wedge f_*(x' \in_X U) \\
\vdash & f_*f^*(x \sim x') \\
\vdash & x \sim x' \qquad\qquad\qquad\qquad \text{since } f \text{ is connected.}
\end{array}$$

□

Some easy observations:

Proposition 1.17. *We assume that P and Q are canonically presented, $\mathbb{C}$-based triposes, where $\mathbb{C}$ is ccc.*

1. *The generic predicate $\sigma \in P(\Sigma_P) \cong \mathrm{Hom}(\Sigma_P, \Sigma_P)$ can be chosen as id_σ.*
2. *Any tripos morphism $f : P \to Q$ is given by composition with a morphism $\hat{f} : \Sigma_P \to \Sigma_Q$ in $\mathbb{C}$.*
3. *The membership predicate $\in_X$ of $P(X \times \pi(X))$ is given by $\mathrm{ev}_X : X \times \Sigma_P^X \to \Sigma_P$.*
4. *$f(x \in_X^P S) \cong x \in_X^Q f(S)$.*

Proof. 1. Let $\phi \in P(X)$ be given. We can choose $[\phi] = \phi : X \to \Sigma_P$ and clearly $\phi \dashv\vdash P([\phi])(id_{\Sigma_P})$ since $P([\phi])(id_{\Sigma_P}) = id_{\Sigma_P} \circ [\phi] = \phi$.

2. This follows from the Yoneda Lemma, we get

$$\hat{f} = f_{\Sigma_P}(id_{\Sigma_P}) : \Sigma_P \to \Sigma_Q.$$

Now let $\phi \in P(X) = \mathrm{Hom}(X, \Sigma_P)$, then $f(\phi) = f(P(\phi)(id_{\Sigma_P})) = Q(\phi)(f(id_{\Sigma_P})) = Q(\phi)(\hat{f}) = \hat{f} \circ \phi$.

3. When $\mathbb{C}$ is ccc, the membership predicate is defined by

$$\in_X = P(\mathrm{ev}_X)(\sigma) = P(\mathrm{ev}_X)(id_{\Sigma_P}) = \mathrm{ev}_X$$

.

4. From the above it follows that $f(x \in^P_X S) = f(\mathrm{ev}^P_X(x, S)) = \hat{f} \circ \mathrm{ev}^P_X(x, S)$. And $x \in^Q_X f(S) = \mathrm{ev}^Q_X(x, f(S)) = \mathrm{ev}^Q_X(x, \hat{f} \circ S)$. By naturality of ev we get
$$\hat{f} \circ \mathrm{ev}^P_X(x, S) = \mathrm{ev}^Q_X(x, \hat{f} \circ S).$$

□

The following theorem is due to Pitts [Pit81].

Theorem 1.18 (Iteration). *Let $\mathbb{C}$ be a finitely complete category, P a $\mathbb{C}$-tripos, and R a $\mathbb{C}[P]$-tripos, and assume that Δ_R preserves epis. Then $R \circ \Delta_P^{op}$ is a $\mathbb{C}$-tripos and $\Delta_{R\circ\Delta_P^{op}} : \mathbb{C} \to \mathbb{C}[R \circ \Delta_P^{op}]$ is equivalent over $\mathbb{C}$ to $\Delta_R \circ \Delta_P : \mathbb{C} \to \mathbb{C}[P][R]$ and the following diagram commutes:*

$$\begin{tikzcd}
\mathbb{C} \arrow[rr, "\Delta_{R\circ\Delta_P^{op}}"] \arrow[dr, "\Delta_P^{op}"'] & & \mathbb{C}[R \circ \Delta_P^{op}] \arrow[dd, leftrightarrow, "\cong"] \\
& \mathbb{C}[P] \arrow[dr, "\Delta_R"'] & \\
& & \mathbb{C}[P][R]
\end{tikzcd}$$

The Toposes Mod, Eff and Eff$_2$ We recall the definitions of the realizability toposes **Eff**, **Mod** and **Eff**$_2$. We use the following abbreviations for $A, B, C \subseteq \mathbb{N}$:

$$\begin{array}{lcl}
1 & = & \{0\}\,, \\
A \otimes B & = & \{\langle a, b\rangle \mid a \in A,\, b \in B\}\,, \\
A \oplus B & = & (\{0\} \otimes A) \cup (\{1\} \otimes B)\,, \\
A \oplus B \oplus C & = & \{0\} \otimes A \cup \{1\} \otimes B \cup \{2\} \otimes C\,, \\
A \Rightarrow B & = & \{e \in \mathbb{N} \mid \forall a \in A.\, e \cdot a \in B\}\,.
\end{array}$$

For $A \subseteq \mathbb{N}\times\mathbb{N}$, we often write $A(x, y)$ for $(x, y) \in A$. We assume that a coding is chosen such that $\langle 0, 0\rangle = 0$ and $0 \cdot x = 0$, for all $x \in \mathbb{N}$. Moreover, we often write $f(x)$ or even fx instead of $f \cdot x$ and $F(x, y)$ instead of $F \cdot \langle x, y\rangle$. We write $\mathcal{P}_f(\mathbb{N})$ for the set of finite subsets of $\mathbb{N}$. Let $e : \mathbb{N} \to \mathcal{P}_f(\mathbb{N})$ be the bijection where $e_n = S$ iff $n = \sum_{k\in S} 2^k$. We write $m \in n$ as abbreviation for $m \in e_n$. If $A \subseteq \mathbb{N}$ we write $\mathcal{P}_f(A)$ for $\{n \in \mathbb{N} \mid e_n \subseteq A\}$ and $\mathcal{P}_{\leq 1}(A)$ for $\{n \in \mathcal{P}_f(A) \mid |e_n| \leq 1\}$.

The effective topos, **Eff** is the topos one obtains by the tripos-to-topos construction on the **Set**-based tripos e, defined as follows. For a set X, $\mathsf{e}(X)$ is the Heyting pre-algebra $(\Sigma_{\mathsf{e}}^X, \vdash_X)$, where $\Sigma_{\mathsf{e}} = \mathcal{P}(\mathbb{N})$ and the connectives are defined by

$$\begin{array}{lclcl}
(A \wedge B)(x) & = & A(x) \wedge B(x) & = & A(x) \otimes B(x), \\
(A \vee B)(x) & = & A(x) \vee B(x) & = & A(x) \oplus B(x), \\
(A \to B)(x) & = & A(x) \to B(x) & = & A(x) \Rightarrow B(x), \\
\bot_X & = & \emptyset, & & \\
\top_X & = & \mathbb{N}. & &
\end{array}$$

and the order on Σ_{e}^X is

$$A \vdash_X B \quad \text{iff} \quad \bigcap_{x \in X} (A \to B)(x) \neq \emptyset.$$

$$\begin{array}{lcl}
\forall_u(A)(j) & = & \bigcap_{i \in u^{-1}(j)} \{0\} \Rightarrow A(i) \\
\exists_u(A)(j) & = & \bigcup_{i \in u^{-1}(j)} A(i)
\end{array}$$

for $u : I \to J$ and $A \in \Sigma_{\mathsf{e}}^I$ For more details see [Hyl82].

The modified realizability topos, **Mod** is the topos one obtains by the tripos-to-topos construction on the **Set**-based tripos m, defined as follows. For a set X, $\mathsf{m}(X)$ is the Heyting pre-algebra $(\Sigma_{\mathsf{m}}^X, \vdash_X)$, where

$$\Sigma_{\mathsf{m}} = \{A = (A_a, A_p) \in \mathcal{P}(\mathbb{N})^2 \mid A_a \subseteq A_p \text{ and } 0 \in A_p\}.$$

The connectives are defined by

$$\begin{array}{lcl}
(A \wedge B)_a(x) & = & A_a(x) \otimes B_a(x), \qquad (A \wedge B)_p(x) = A_p(x) \otimes B_p(x), \\
(A \to B)_a(x) & = & (A_a(x) \Rightarrow B_a(x)) \cap (A_p(x) \Rightarrow B_p(x)), \\
(A \to B)_p(x) & = & (A_p(x) \Rightarrow B_p(x)), \\
\top_a(x) & = & \mathbb{N}, \qquad \top_p(x) = \mathbb{N}, \\
\bot_a(x) & = & \mathbb{N}, \qquad \bot_p(x) = \emptyset.
\end{array}$$

and the order on Σ_{m}^X is

$$A \vdash_X B \quad \text{iff} \quad \bigcap_{x \in X} (A \to B)_a(x) \neq \emptyset.$$

The quantifiers are given by

$$\begin{array}{l}
\forall_u(A)(j) = \big(\bigcap_{i \in u^{-1}(j)} \{0\} \Rightarrow A_a(i) \ , \ \bigcap_{i \in u^{-1}(j)} \{0\} \Rightarrow A_p(i) \ \big) \\
\exists_u(A)(j) = \big(\mathtt{succ}\big(\bigcup_{i \in u^{-1}(j)} A_a(i) \big) \ , \ \{0\} \cup \mathtt{succ}\big(\bigcup_{i \in u^{-1}(j)} A_p(i) \big) \ \big)
\end{array}$$

for $u : I \to J$ and $A \in \mathsf{m}^I$. Notice that this description of quantifiers is easily seen to be equivalent to the one of [vO97].

The topos $\mathbf{Eff}_2$ is defined as the topos built on the realizability tripos over $\mathbf{Set}^{\to}$, i.e., the $\mathbf{Set}^{\to}$-tripos

$$X \mapsto \mathbf{Set}^{\to}(X, \mathcal{P}(N)).$$

where N is the natural numbers object in $\mathbf{Set}^{\to}$. The following is due to van Oosten [vO97].

Proposition 1.19. *Let* $\Sigma_{\mathsf{e}_2} = \{A = (A_a, A_p) \in \mathcal{P}(\mathbb{N})^2 \mid A_a \subseteq A_p\}$*, and define the preorder* $\vdash_X$ *on* $\Sigma^X_{\mathsf{e}_2}$ *just as* $\vdash_X$ *on* Σ^X_{m}*, then* $\mathsf{e}_2(X) = (\Sigma^X_{\mathsf{e}_2}, \vdash_X)$ *defines a* **Set***-tripos, and the topos* $\mathbf{Eff}_2$ *is obtained by the tripos-to-topos construction on this tripos.*

Proof: It is well known that $\mathbf{Set}^{\to}$ is sheaves over Sierpinski space, so $\mathbf{Set}^{\to} \cong \mathbf{Set}[\mathbf{Set}(-, \Sigma_S)]$, where Σ_S is the locale corresponding to the open sets of Sierpinski space. Using Pitt's Iteration Theorem with $P = \mathbf{Set}(-, \Sigma_S)$, and with $R = \mathbf{Set}^{\to}(-, \mathcal{P}(N))$, we get that

$$\mathbf{Set}[P][R] \cong \mathbf{Set}[R \circ \Delta_P^{op}],$$

i.e., that

$$\mathbf{Eff}_2 \cong \mathbf{Set}[\mathbf{Set}^{\to}(\Delta_P(-), \mathcal{P}(N))].$$

Now, the tripos $\mathbf{Set}^{\to}(\Delta_P(-), \mathcal{P}(N))$ can be described as follows: $\mathcal{P}(N)$ in $\mathbf{Set}^{\to}$ is

$$\begin{tikzcd} \{(A,B) \mid A \subseteq B \subseteq \mathbb{N}\} \arrow[d, "\pi_1"] \\ \mathcal{P}(\mathbb{N}) \end{tikzcd}$$

and $\Delta_P(X) = id : X \to X$, so an element of the set $\mathbf{Set}^{\to}(\Delta_P(X), \mathcal{P}(N))$, are two functions (ϕ, ϕ') such that the following commutes

$$\begin{tikzcd} X \arrow[r, "\phi"] \arrow[d, "id"'] & \{(A,B) \mid A \subseteq B \subseteq \mathbb{N}\} \arrow[d, "\pi_1"] \\ X \arrow[r, "\phi'"'] & \mathcal{P}(\mathbb{N}) \end{tikzcd}$$

and clearly this is defined by $\phi : X \to \{(A,B) \mid A \subseteq B \subseteq \mathbb{N}\}$. The order is derived from the definition of a standard realizability tripos:

$$\phi \vdash_X \psi \quad \text{iff} \quad \mathbf{Set}^{\to} \models \exists n : N \forall x : X. n \in (\phi(x) \Rightarrow \psi(x)).$$

□

2 Local Operators

In this section we recall some results about local operators for toposes as well as for triposes. We also prove a factorization theorem for geometric morphisms of triposes.

Definition 2.1 (Local Operator)**.** *A* local operator *(also known as a Lawvere-Tierney topology or j-topology) on a topos* $\mathcal{E}$*, is a morphism* $j : \Omega \to \Omega$*, satisfying the following axioms in the internal logic of* $\mathcal{E}$*:*

1. $j(\top) = \top$,

2. $jj(a) = j(a)$,

3. $j(a \wedge b) = j(a) \wedge j(b)$.

There is a correspondence between local operators and universal closure operation, which we define now:

Definition 2.2 (Universal Closure Operation)**.** *A universal closure operation on $\mathcal{E}$ consists of, for every object X of $\mathcal{E}$ an operation cl_X on $\mathrm{Sub}(X)$ satisfying:*

1. $A \leq \mathrm{cl}_X(A) = \mathrm{cl}_X(\mathrm{cl}_X(A))$,

2. stability under pullback: for $f : X \to Y$, and $B \in \mathrm{Sub}(Y)$, we have $\mathrm{cl}_X(f^*(B)) = f^*(\mathrm{cl}_Y(B))$.

We have the following

Proposition 2.3. *For any topos $\mathcal{E}$, there is a bijection between universal closure operations on $\mathcal{E}$ and local operators on $\mathcal{E}$.*

Proof: We only give the bijection, for a detailed proof see e.g. [Joh02, MLM94, vO08]. Given $j : \Omega \to \Omega$ let c_j be the operation on subobject defined by composing the classifying map with j. Conversely, given a universal closure operation cl, Let $\Omega_j = \mathrm{cl}_\Omega(\top)$, where $\top : 1 \to \Omega$ is the subobject classifier, then Ω_j is a subobject of Ω with classifying map $j : \Omega \to \Omega$. □

Example 2.4. *Typical examples of local operators are (using the internal logic of the topos) $\neg\neg : \Omega \to \Omega$, and for a subterminal object $U \rightarrowtail 1$, with classifying map $u : 1 \to \Omega$, the* open topology *$o(u)$ defined by $u \Rightarrow (-)$, and the* closed topology *$c(u)$ defined by $u \vee (-)$.*

Theorem 2.5. *Every local operator j on $\mathcal{E}$, determines a subtopos $\mathcal{E}_j \hookrightarrow \mathcal{E}$. Conversely, every geometric inclusion of toposes $i : \mathcal{E}' \hookrightarrow \mathcal{E}$ gives rise to a unique local operator j on $\mathcal{E}$, and $\mathcal{E}_j$ is equivalent to $\mathcal{E}'$.*

For a proof see [Joh02] or [MLM94].

We now turn to toposes of the form $\mathcal{E}[P]$, for a canonical $\mathcal{E}$-tripos $P = \mathcal{E}(-\Sigma)$. First we define what we mean by local operator on a canonical tripos $P = \mathcal{E}(-, \Sigma)$ over a topos $\mathcal{E}$.

Definition 2.6 (Local Operator for Tripos)**.** *A local operator on a tripos $P = \mathcal{E}(-, \Sigma)$, where $\mathcal{E}$ is a topos, is a map $J : \Sigma \to \Sigma$ satisfying*

1. $P \models (p \to q) \to (Jp \to Jq)$,

2. $P \models J(\top) \leftrightarrow \top$,

3. $P \models J(Jp) \leftrightarrow Jp$, *and*

4. $P \models J(p \wedge q) \leftrightarrow (Jp \wedge Jq)$. *It follows that*

12

5. $P \models p \to J(p)$.

Proposition 2.7. *Every local operator j on $\mathcal{E}[P]$ determines a local operator J on Σ, and vice versa.*

Proof: Given local operator j on the topos $\mathcal{E}[P]$, let $\Omega_j \rightarrowtail \Omega$ be the subobject classified by j (the generic j-dense subobject). Since $\Omega = (\Sigma, \leftrightarrow_P)$, the subobject Ω_j is represented by a strict predicate $J : \Sigma \to \Sigma$. Since $p \equiv p \leftrightarrow \top$ in P, one can infer that $j(p,q) \equiv q \leftrightarrow j(p,\top)$. Hence J is given by

$$J(p) \equiv j(p,\top).$$

Conversely, given a local operator $J : \Sigma \to \Sigma$ it defines a strict relation on Ω in $\mathcal{E}[P]$ whose classifying map is a local operator on $\mathcal{E}[P]$. For more details see [Pit81]. □

The operator J on the tripos P determines a subtripos P_J, defined as follows: P_J is canonically presented on Σ, but $\leftrightarrow, \forall$ and **D** (the designated truth values) are redefined by letting:

- $\to^{P_J}$ be $\Sigma \times \Sigma \xrightarrow{id \times J} \Sigma \times \Sigma \xrightarrow{\to} \Sigma$.
- $(\forall_f)(p)$ be $(\forall_f)(Jp)$.
- $1 \xrightarrow{p} \Sigma$ be in $\mathbb{D}_{P_J}$ iff $1 \xrightarrow{p} \Sigma \xrightarrow{J} \Sigma$ is in $\mathbb{D}_P$. That is, $\top \vdash_{P_J} p$ iff $\top \vdash_P J(p)$.
- It follows that $p \vdash_{P_J} q$ iff $p \vdash_P J(q)$.

That is, P_J is the canonically presented tripos $(\Sigma, \vdash_{P_J})$, where $p \vdash_{P_J} q$ iff $p \vdash_P J(q)$. Notice that this is actually the Kleisli category for the monad J on P (a local operator is also a left exact monad).

One may define the tripos P_J as a canonically presented tripos on

$$\Sigma_J = \{p \in \Sigma \mid p \leftrightarrow_P J(p)\} = \{p \in \Sigma \mid J(p) \vdash^P p\}$$

and with the connectives $\wedge, \top, \to, \forall$ inherited from P, and $\exists, \vee, \bot$ defined by $J(\exists p)$, $J(p \vee q)$, $J(\bot)$. This is perhaps a more natural way to define P_J, bearing in mind that local operators correspond to subtriposes (we come to that later).

To see why the tripos $(\Sigma_J, \vdash_P)$ is equivalent to $(\Sigma, \vdash_{P_J})$, as indexed categories (by the equivalence $J \dashv i$), just notice that $J(p) \dashv\vdash_{P_J} p$ iff $J(p) \vdash_P J(p)$ and $p \vdash_P JJ(p)$.

Notice that the definition of Σ_J makes sense in $\mathcal{E}$ because P is canonically presented, so $\to_P \in \mathcal{E}(\Sigma \times \Sigma, \Sigma)$. Thus we have a map $p \mapsto (p \leftrightarrow_P J(p))$ from Σ to Σ, so $\top = (p \leftrightarrow_P J(p))$ is a predicate defining a subobject of Σ in $\mathcal{E}$.

Example 2.8. *Let $j : \Omega \to \Omega$ in $\mathcal{E}[P]$ be defined in terms of the internal logic of $\mathcal{E}[P]$, e.g., $j(p) = \neg\neg p$. Since $\Omega = (\Sigma, \leftrightarrow_P)$, j determines a functional relation represented by $\hat{j}$ on $\Sigma \times \Sigma$. $\hat{j}$ is defined by*

$$\hat{j}(p,q) \equiv j(p) \leftrightarrow q.$$

Now, on the tripos level, we get a local operator $J : \Sigma \to \Sigma$ by

$$J(p) \equiv \hat{j}(p, \top) \equiv j(p),$$

where the latter equivalence is due to the fact that $p \leftrightarrow \top \equiv p$ in P.

Proposition 2.9. *Let $(\mathcal{E}[P])_j$ be the sheaf subtopos corresponding to a local operator J on P, then $(\mathcal{E}[P])_j$ is equivalent over $\mathcal{E}$ to $\mathcal{E}[P_J]$.*

For a proof see [Pit81].

We now show the tripos version of a well known factorization theorem for toposes, namely:

Proposition 2.10. *Every geometric morphism factors as a surjection followed by an inclusion. The factorization is essentially unique.*

For a proof see e.g. [Joh02, MLM94]. Now the tripos version is:

Theorem 2.11. *Let P and Q be canonically presented triposes, over a topos $\mathcal{E}$. Every geometric morphism $f : P \to Q$ factors as a connected geometric morphism followed by an inclusion. The factorization is essentially unique.*

Proof. Let $L = f^* f_*$ be the comonad on P defined from f, then $L \dashv id$ and

$$(L, id) : P \twoheadrightarrow P_L$$

is a connected geometric morphism, and P_L the Kleisli category of L. $L \dashv id$ is the usual adjunction associated with the Kleisli category. We show that $L : P_L \to P$ is full and faithful: $Lp \vdash^P Lq$ implies $Lp \vdash^P Lq \vdash^P q$, i.e., $p \vdash^{P_L} q$.

On Q we have a local operator defined by $j = f_* f^*$. j is clearly functorial and preserves finite limits. Since $f_* f^*$ also defines a monad on Q, we have

$$p \vdash^Q jp \qquad \text{and} \qquad jjp \vdash^Q jp$$

so j is idempotent. Q_j is the full subcategory of Q of j-sheaves, i.e.,

$$\Sigma_{Q_j} = \{p \in \Sigma_Q \mid jp \dashv\vdash^Q p\} = \{p \in \Sigma_Q \mid jp \vdash^Q p\}.$$

Notice that Q_j is equivalent to the category of algebras for the monad j. We have $j \dashv i : Q_j \hookrightarrow Q$. We now show that i is full and faithful: For all $p, q \in Q_j$, $ip \vdash^Q iq$ iff $p \vdash^{Q_j} q$.

We will show that the following diagram commutes

$$\begin{tikzcd}
P \arrow[rr, "f"] \arrow[dr, "g"'] & & Q \\
& P_L \simeq Q_j \arrow[ur, hook, "i"'] &
\end{tikzcd}$$

where $g_* = in$, $g^* = L$ and $i_* = i$, $i^* = j$.

First we show the equivalence $P_L \cong Q_j$. It is given by $f_*L : P_L \to Q_j$, and $f^* : Q_j \to P_L$. Let $p \in P_L$, to see that $f_*Lp \in Q_j$, we check that f_*Lp is j-closed: $j(f_*Lp) \equiv^Q f_*Lp$ iff $j(f_*f^*f_*p) \equiv^Q f_*f^*f_*p$ which is to say that $jjf_*p \equiv^Q jf_*p$. Let $q \in Q_j$, then $f^*q \in P_L$. To see that this defines an equivalence, let $p \in P_L$, then $f^*f_*Lp = LLp \equiv^{P_L} p$. And for $q \in Q_j$, $f_*Lf^*q = f_*f^*f_*f^*q = jjq \equiv^{Q_j} q$.

To see that the diagram commutes (up to iso), we must show that

$$i_*f_*Lg_* = if_*Lid \cong f_* \quad \text{and} \quad g^*f^*i^* = Lf^*j \cong f^*$$

The latter is equal to

$$LLf^* \cong f^*$$

which holds since for all $q \in Q$, $LLf^*q \vdash^P f^*q$ and $f^*q \vdash^P LLf^*q$ iff $q \vdash^Q f_*LLf^*q = jjjq = jq$. The other equation holds by uniqueness of adjoints (because an equivalence is an adjunction). Uniqueness: Consider two factorizations:

$$\begin{tikzcd}
 & A \arrow[rd, hook, "u"] \arrow[dd, dashed, "g"] & \\
P \arrow[ru, "p"] \arrow[rd, "q"'] & & Q \\
 & B \arrow[ru, hook, "v"'] &
\end{tikzcd}$$

where u, v are inclusions and p, q are connected. Then g is an equivalence. $g^* : B \to A$ is defined by q_*p^* and $g_* : A \to B$ is q_*p^*. Observe that $p_*q^*b \vdash^A a$ iff $u_*p_*q^*b \vdash^Q u_*a$ iff $v_*q_*q^*b \vdash^Q u_*a$ iff $u^*v_*b \vdash^A a$, so $p_*q^* \cong u^*v_*$. Likewise we have $q_*p^* \cong v^*u_*$. Verifying that g is an equivalence and that the diagram commutes is routine.

□

Together with Theorem 1.16 which says that a connected geometric morphism of triposes lifts to a surjection of toposes, this ties the factorization theorems for triposes and toposes very neatly together. Notice that there is a similar factorization theorem for locales, see [MLM94].

For the sake of completeness we also mention that even if P is not a canonical tripos over a topos, we can still have a notion of closure operation on P that corresponds to inclusions of toposes into $\mathbb{C}[P]$.

Definition 2.12 (Closure Operation on Tripos). *A closure operation Φ on a tripos P is an indexed functor $\Phi : P \to P$, which preserves finite meets, is idempotent, and inflationary ($id \leq \Phi$).*

The following is from [vO08].

Theorem 2.13. *let P be a $\mathbb{C}$-tripos.*

1. *Every inclusion of toposes into $\mathbb{C}[P]$ is, up to equivalence, of the form $\mathbb{C}[Q] \to \mathbb{C}[P]$ for a $\mathbb{C}$-tripos Q and an inclusion of triposes $Q \to P$.*

2. Inclusions into P correspond, up to equivalence, to closure operations on P.

The following example is from [vO97].

Example 2.14. **Eff** *is an open subtopos of* $\mathbf{Eff}_2$ *and* **Mod** *is its closed complement. Let $U = (\emptyset, \mathbb{N})$. U is a subobject of 1 in the topos $\mathbf{Eff}_2$. We have inclusions $\delta : \mathbf{Eff} \hookrightarrow \mathbf{Eff}_2$ and $i : \mathbf{Mod} \hookrightarrow \mathbf{Eff}_2$. These give rise to local operators k_E, k_M on $\mathbf{Eff}_2$. Since $\mathbf{Eff}_2 = \mathbf{Set}[\mathsf{e}_2]$ we get local operators K_E, K_M. On the tripos level, these are given by:*

$$K_E(A, B) = \delta^*\delta_*(A, B) = (B, B),$$

and

$$K_M(A, B) = i_*i^*(A, B) = (\mathtt{succ}(A), \mathtt{succ}(B) \cup \{0\}).$$

Now in $\Sigma_{\mathsf{e}_2}^{\Sigma_{\mathsf{e}_2}}$ we have $K_E(A, B) \equiv U \Rightarrow (A, B)$ hence k_E is given by $p \mapsto (u \to p)$, where u is the classifying map of $U \rightarrowtail 1$. Likewise, $K_M(A, B) \equiv U \vee (A, B)$ so k_M is the closed topology $p \mapsto u \vee p$ on $\mathbf{Eff}_2$.

Definition 2.15 (Localic)**.** *A topos $\mathcal{E}$ is localic over a topos $\mathcal{F}$ via a geometric morphism f, if $\mathcal{E}$ is equivalent to the topos of $\mathcal{F}$-valued sheaves on the internal locale $f_*(\Omega_{\mathcal{E}})$, i.e., $\mathcal{E} \cong \mathcal{F}[f_*(\Omega_{\mathcal{E}})] = \mathcal{F}[\mathcal{F}(-, f_*(\Omega_E))]$.*

The following is from [Bir99]

Theorem 2.16. *let $\mathbb{C}$ be a finitely complete category and let P and Q be $\mathbb{C}$ triposes. Suppose $f = (f^*, f_*) : P \to Q$ is a geometric morphism of triposes, then $\mathbb{C}[P]$ is localic over $\mathbb{C}[Q]$ via the induced geometric morphism $\bar{f} = (\bar{f}^*, \bar{f}_*) : \mathbb{C}[P] \to \mathbb{C}[Q]$.*

We now give some more details of the situation in [Bie08, Chapter 3] regarding the pullbacks of the form

$$\begin{tikzcd} \mathbf{DN}_e \arrow[r] \arrow[d, hook] & \mathbf{Eff}, \arrow[d, hook] \\ \mathbf{DN} \arrow[r, "q"'] & \mathbf{Eff}_2 \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} \mathbf{DN}_m \arrow[r] \arrow[d, hook] & \mathbf{Mod} \arrow[d, hook] \\ \mathbf{DN} \arrow[r, "q"'] & \mathbf{Eff}_2 \end{tikzcd}$$

Suppose $\mathcal{D} = \mathbf{Set}[d]$ is a topos over a tripos d and $q : \mathcal{D} \to \mathbf{Eff}_2$ a geometric morphism induced by a geometric morphism $q : \mathsf{d} \to \mathsf{e}_2$, then we have $\mathcal{D} \cong \mathbf{Eff}_2[q_*(\Omega_D)]$, and $q_*(\Omega_D)$ is an internal local in $\mathbf{Eff}_2$. Consider the open topology $o(u)$ on $\mathbf{Eff}_2$ from Example 2.14 above. Let $o(u)(\Omega_{\mathbf{Eff}_2})$ denote the generic $o(u)$-dense subobject of $\Omega_{\mathbf{Eff}_2}$. $o(u)(\Omega_{\mathbf{Eff}_2})$ is an internal locale, and by [Joh02, Lemma 1.2.10] we have a pullback of internal locales in $\mathbf{Eff}_2$:

$$\begin{tikzcd} o(q^*(u))(q_*(\Omega_{\mathcal{D}})) \arrow[r] \arrow[d, hook] & o(u)(\Omega_{\mathbf{Eff}_2}) \arrow[d, hook] \\ q_*(\Omega_{\mathcal{D}}) \arrow[r, "q"'] & \Omega_{\mathbf{Eff}_2} \end{tikzcd}$$

(The same holds for closed topologies.) Since the functor $L \mapsto \mathbf{Eff}_2[L]$ from locales to toposes preserves limits (see [MLM94, Joh02, JT84]), there is a pullback of toposes

$$\begin{tikzcd}
\mathbf{Eff}_2[o(q^*(u))(q_*(\Omega_{\mathcal{D}}))] \arrow[r] \arrow[d, hook] & \mathbf{Eff}_2[o(u)(\Omega_{\mathbf{Eff}_2})] \cong \mathbf{Eff} \arrow[d, hook] \\
\mathbf{Eff}_2[q_*(\Omega_{\mathcal{D}})] \arrow[r, "q"'] & \mathbf{Eff}_2[\Omega_{\mathbf{Eff}_2}] \cong \mathbf{Eff}_2
\end{tikzcd}$$

Now, by Theorem 2.16 we have $\mathbf{Eff}_2[q_*(\Omega_{\mathcal{D}})] \cong \mathcal{D}$, so using Proposition 2.9 we get $\mathbf{Eff}_2[o(q^*(u))(q_*(\Omega_{\mathcal{D}}))] \cong \mathbf{Eff}_2[(q_*(\Omega_{\mathcal{D}}))]_{o(q^*(u))} \cong \mathcal{D}_{o(q^*(u))}$.

More generally, it holds that (see [Joh02]):

Proposition 2.17. *Pullback along a localic geometric morphism of toposes preserves open/closed subtoposes.*

References

[Bie08] Bodil Biering. *Dialectica Interpretations: a Categorical Analysis.* PhD thesis, IT-University of Copenhagen, 2008.

[Bir99] L. Birkedal. *Developing Theories of Types and Computability.* PhD thesis, School of Computer Science, Carnegie Mellon University, December 1999.

[Fre07] Jonas Frey. A universal characterization of the tripos-to-topos construction. Technical report, Technische Universität Darmstadt, 2007.

[HJP80] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980.

[Hyl82] J.M.E. Hyland. The effective topos. In A.S. Troelstra and D. Van Dalen, editors, *The L.E.J. Brouwer Centenary Symposium*, pages 165–216. North Holland Publishing Company, 1982.

[Joh02] Peter T. Johnstone. *Sketches of an elephant: a topos theory compendium.*, volume 44 of *Oxford Logic Guides.* The Clarendon Press Oxford University Press, Oxford, 2002.

[JT84] André Joyal and Myles Tierney. An extension of the Galois theory of Grothendieck. *Mem. Amer. Math. Soc.*, 51(309):vii+71, 1984.

[MLM94] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic.* Universitext. Springer-Verlag, New York, 1994. A first introduction to topos theory, Corrected reprint of the 1992 edition.

[Pit81] A.M. Pitts. *The Theory of Triposes.* PhD thesis, Cambridge University, 1981.

[vO97] J. van Oosten. The modified realizability topos. *Journal of Pure and Applied Algebra*, 116:273–289, 1997.

[vO08] J. van Oosten. *Realizability - An Introduction to its Categorical Side.* Studies in Logic, 152, 2008.

Chapter 2

Topos Theoretic Versions of Dialectica Interpretations

The preprint presented in this chapter has grown out of two sets of unpublished notes, one written by Thomas Streicher, the other by Lars Birkedal. These notes in turn grew out of ideas originating from Martin Hyland, and from many discussions between all of the authors. The two sets of notes were merged and thoroughly revised by the author, changes have been made and new results added along the way. For this paper we assume that the reader is familiar with the background material from Chapter 1. Again, the reader should be familiar with basic tripos theory (see [Pit02, HJP80, Pit81]) and have some knowledge about toposes and j-topologies (see e.g. [MLM94, Joh02]).

In Chapter 2 we present four new triposes reflecting as much as possible of the Dialectica interpretation, which we call the *Dialectica tripos* and denote by d. The resulting topos is denoted by $\mathbf{Dia}$. From d we get a closed subtripos, the *modified Dialectica tripos*, denoted by d_m, and the resulting topos is denoted by $\mathbf{Dia}_m$. We also define a tripos reflecting as much as possible of the Diller-Nahm interpretation, which we call the *Diller-Nahm tripos*, denoted dn. The resulting topos is denoted by $\mathbf{DN}$. From dn we get a closed subtripos, the *modified Diller-Nahm tripos*, denoted by dn_m, and the resulting topos is denoted by $\mathbf{DN}_m$. The modified versions are in closer correspondence with the standard interpretations of Dialectica, respectively Diller-Nahm, since "modified" corresponds to having non-empty types. We give an account of the first order logic of the toposes and find that first order logic of dn_m corresponds to the Diller-Nahm interpretation, and that first order logic of $\mathbf{Dia}_m$ does not correspond to the Dialectica interpretation, but instead to a variant of Dialectica, which we call the Copenhagen interpretation. This is perhaps not so surprising when we recall that the Dialectica interpretation assumes that atomic formulas are decidable, and that there is no such restriction for the tripos. Though we have some nice results regarding the decidable fragment of predicates over the natural numbers in $\mathbf{Dia}_m$, we argue that it is not possible to interpret first order logic with decidable atomic formulas in this fragment. Hence we do not find a correspondence between first order logic with decidable atomic formulas in $\mathbf{Dia}_m$ and the Dialectica interpretation. The tripos setting allows us to reveal many new relations in the form of geometric morphisms to other functional interpretations, which are also represented by triposes.

The exponent construction of the triposes d and d_m is studied in Chapter 4 and has also lead to the Copenhagen interpretation presented in Chapter 5.

References

[HJP80] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980.

[Joh02] Peter T. Johnstone. *Sketches of an elephant: a topos theory compendium.*, volume 44 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, Oxford, 2002.

[MLM94] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic*. Universitext. Springer-Verlag, New York, 1994. A first introduction to topos theory, Corrected reprint of the 1992 edition.

[Pit81] A.M. Pitts. *The Theory of Triposes*. PhD thesis, Cambridge University, 1981.

[Pit02] Andrew M. Pitts. Tripos theory in retrospect. *Math. Structures Comput. Sci.*, 12(3):265–279, 2002. Realizability (Trento, 1999).

Topos Theoretic Versions of Dialectica Interpretations

B. Biering, L. Birkedal, C. Butz, J.M.E. Hyland,
J. van Oosten, G. Rosolini, T. Streicher

Contents

In this paper we present four new triposes built on Gödel's Dialectica interpretation [Göd58] (named so after the journal in which it was published) and the Diller-Nahm variant of Gödel's Dialectica interpretation [DN74]. We define triposes reflecting as much as possible the original idea of Gödel's Dialectica Interpretation and the Diller-Nahm variant of Gödel's Dialectica Interpretation. The motives for studying these triposes and their related toposes (by the tripos-to-topos construction, see [HJP80]) are many-fold.

By studying these categorical constructions as models of the Dialectica and Diller-Nahm interpretations, we can obtain new insight into the Dialectica and Diller-Nahm interpretations and their relations to other functional interpretations. Moreover, the categorical analysis is likely to lead to new, interesting variants of functional interpretations and at the same time the triposes and toposes are interesting in themselves as new models for logic and type theory. The Effective Topos [Hyl82] gave rise to a higher order version of Kleene's number realizability. One aim is to achieve a similar result for the Dialectica and Diller-Nahm interpretations. At the moment, it is not quite clear whether the toposes presented in this paper are exactly the "right" ones for achieving this goal.

Clearly, there are many questions related to the triposes and toposes we present here that deserve an answer, and admittedly, more questions are posed than answered by this paper. Any one of these questions requires a thorough analysis (an example of this is the somewhat surprising exponent construction in the Dialectica tripos, which has been studied in detail in [Bie07a]), and the authors therefore think that the best way to proceed is to present the core of the subject in this paper and outsource or postpone some of the related questions.

We define a tripos reflecting as much as possible of the Dialectica interpretation, which we call the *Dialectica tripos* and denote by d. The resulting topos is denoted by **Dia**. From d we get a closed subtripos, the *modified Dialectica tripos*, denoted by d_m, and the resulting topos is denoted by $\mathbf{Dia}_m$.

We also define a tripos reflecting as much as possible of the Diller-Nahm interpretation, which we call the *Diller-Nahm tripos*, denoted dn. The resulting topos is denoted by **DN**. From dn we get a closed subtripos, the *modified Diller-Nahm tripos*, denoted by dn_m, and the resulting topos is denoted by $\mathbf{DN}_m$.

For each of the four resulting toposes, we give an account of first order logic (over the natural numbers). The non-modified toposes, **Dia** and **DN** have strong connections with the effective topos, **Eff** via open inclusions. These open inclusions gives a lot of information about the first order logic of **Dia** and **DN**. For the modified versions, $\mathbf{Dia}_m$ and $\mathbf{DN}_m$, the situation is not as straight forward, for these two toposes we give a characterizations of first order logic and prove it by induction. The modified versions are in closer correspondence with the standard interpretations of Dialectica, respectively Diller-Nahm, since "modified" corresponds to having non-empty types. We also study relations (in the form of geometric morphisms) between the new triposes/toposes and more familiar realizability triposes/toposes. The relations can be summed up in diagrams:

At the tripos level, we get the following diagram of fibred adjunctions, where, however, only some give rise to geometric morphisms:

$$\begin{tikzcd}
 & \mathsf{dn}_m \arrow[rr, shift left, "!"] \arrow[d, shift right, "q_*"'] & & \mathsf{d}_m \arrow[ll, shift left, "id"] \arrow[rr, shift left, "F"] \arrow[d, shift left, "H_*"] \arrow[dll, shift left, "p_*"] & & \mathbf{Set} \arrow[ll, shift left, "\nabla"] \\
 & \mathsf{m} \arrow[u, shift right, "q^*"'] \arrow[urr, shift left, "p^*"] \arrow[rr, shift left, "v_*"] \arrow[dl, shift left, "\Gamma"] & & \mathsf{e} \arrow[u, shift left, "H^*"] \arrow[ll, shift left, "v^*"] \arrow[rr, shift left, "i_*"] \arrow[dl, shift left, "\Gamma"] \arrow[dr, shift left, "j^*"] & & \mathsf{dn} \arrow[ll, shift left, "i^*"] \\
\mathbf{Set} \arrow[ur, shift left, "\Delta"] \arrow[rr, "\cong"] & & \mathbf{Set} \arrow[ur, shift left, "\Delta"] & & \mathsf{d} \arrow[ul, shift left, "j_*"] &
\end{tikzcd}$$

where H, v, and q all are connected geometric morphisms, so they lift to surjective geometric morphisms on the induced toposes, and i and j are open geometric inclusions, so they lift to open geometric inclusions. The left adjoints of the adjunctions $! \dashv id$, $\nabla \dashv F$, and $p^* \dashv p_*$ are all full and faithful. At the topos level we get the following geometric morphisms

$$\begin{tikzcd}
 & \mathbf{DN}_m \arrow[d, "q"] & & \mathbf{Dia}_m \arrow[d, "H"] & & \\
 & \mathbf{Mod} \arrow[rr, "v"] & & \mathbf{Eff} \arrow[rr, hook, "i"] \arrow[dr, hook, "j"] & & \mathbf{DN} \\
\mathbf{Mod}_{\neg\neg} \arrow[ur, hook] \arrow[rr, "\cong"] & & \mathbf{Eff}_{\neg\neg} \arrow[ur, hook] & & \mathbf{Dia} &
\end{tikzcd}$$

with i, j open inclusions.

We find that the triposes d and d_m have a doubly closed structure, and it is $(\otimes, \multimap)$ and not $(\wedge, \rightarrow)$ that corresponds to Gödel's interpretation of conjunction and implication. In Gödel's Dialectica interpretation one assumes that atomic predicates are decidable, and this assumption is directly related to the interpretation of conjunction. For $\mathbf{Dia}_m$ we show that in the fragment of decidable subobjects over the natural numbers, conjunction and implication correspond exactly to Gödel's definition. One would hope to get a subtopos by restricting to this decidable fragment, since this would yield a higher order version of Dialectica interpretation, but we conjecture that this is not possible. Any decidable predicate is also $\neg\neg$-stable, and for **Dia** we show that $\mathbf{Dia}_{\neg\neg} \cong \mathbf{Set}$.

Outline The rest of the paper is organized as follows:

We start by defining the Dialectica tripos in detail and we study its relation to the tripos e_2 and to number realizability, finally we record some results regarding the first order logic of the topos **Dia**, which is built on the Dialectica tripos using the tripos to topos construction. The modified Dialectica tripos arises as a closed subtripos of the Dialectica tripos. For the modified Dialectica tripos we study decidable predicates, relation to number realizability, first order logic of the natural numbers objects, and the relation to modified realizability and to **Set**. In section 3 we move on to the Diller-Nahm tripos and again we study the relation to $\mathbf{Eff}_2$ and to number realizability, and we also get a modified Diller-Nahm tripos as a closed subtripos, which we then study. In Section 5 we make a connection between the triposes d and dn and also between d_m and dn_m. Finally, we describe a first order fibration that corresponds exactly to the Dialectica interpretation.

Throughout this paper we use the following abbreviations for $A, B, C \subseteq \mathbb{N}$:

$$\begin{array}{lcl} 1 & = & \{0\}\,, \\ A \otimes B & = & \{\, \langle a, b\rangle \mid a \in A, b \in B \,\}\,, \\ A \oplus B & = & (\{0\} \otimes A) \cup (\{1\} \otimes B)\,, \\ A \oplus B \oplus C & = & \{0\} \otimes A \cup \{1\} \otimes B \cup \{2\} \otimes C\,, \\ A \Rightarrow B & = & \{\, e \in \mathbb{N} \mid \forall a \in A.\, e \cdot a \in B \,\}\,. \end{array}$$

For $A \subseteq \mathbb{N} \times \mathbb{N}$, we often write $A(x, y)$ for $(x, y) \in A$. We assume that a coding is chosen such that $\langle 0, 0\rangle = 0$ and $0 \cdot x = 0$, for all $x \in \mathbb{N}$. Moreover, we often write $f(x)$ instead of $f \cdot x$ and $F(x, y)$ instead of $F \cdot \langle x, y\rangle$. We write $\mathcal{P}_f(\mathbb{N})$ for the set of finite subsets of $\mathbb{N}$. Let $e : \mathbb{N} \to \mathcal{P}_f(\mathbb{N})$ be the bijection where $e_n = S$ iff $n = \sum_{k \in S} 2^k$. We write $m \in n$ as abbreviation for $m \in e_n$. If $A \subseteq \mathbb{N}$ we write $\mathcal{P}_f(A)$ for $\{n \in \mathbb{N} \mid e_n \subseteq A\}$ and $\mathcal{P}_{\leq 1}(A)$ for $\{n \in \mathcal{P}_f(A) \mid |e_n| \leq 1\}$.

1 The Dialectica Tripos

In this section we define the canonically presented **Set**-based *Dialectica tripos* d. The set of propositions of d is given by

$$\Sigma_{\mathsf{d}} = \{(U, X, R) \in \mathcal{P}(\mathbb{N})^2 \times \mathcal{P}(\mathbb{N}^2) \mid R \subseteq U \times X\}$$

and for $A = (U, X, R) \in \Sigma_{\mathsf{d}}$ we write A^+ for U, A^- for X and $A(u, x)$ for $(u, x) \in R$. For $I \in \mathbf{Set}$ we define a preorder $\vdash_I$ on the fibre Σ_{d}^I where $A \vdash_I B$ iff there exist $f, F \in \mathbb{N}$ such that, for all $i \in I$, $f \in A_i^+ \Rightarrow B_i^+$ and $F \in A_i^+ \otimes B_i^- \Rightarrow \mathcal{P}_{\leq 1}(A_i^-)$ and, moreover, for all $u \in A_i^+$, $y \in B_i^-$, $\big(\forall x \in F(u, y)\, A_i(u, x)\big) \supset B_i(f(u), y)$.

Truth and Falsity
The terminal object and the initial object are given by

$$\top = \big(\{0\}, \emptyset, \emptyset\big) \quad \text{and} \quad \bot = \big(\emptyset, \emptyset, \emptyset\big)$$

respectively.

Conjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their conjunction $A_0 \wedge A_1$ is given by $(A_0 \wedge A_1)^+ = A_0^+ \otimes A_1^+$, $(A_0 \wedge A_1)^- = A_0^- \oplus A_1^-$ and $(A_0 \wedge A_1)(\langle n_0, n_1\rangle, m) \equiv A_{\pi(m)}(n_{\pi(m)}, \pi'(m))$.

Implication
For $A, B \in \Sigma_{\mathsf{d}}$ their implication $A \to B$ is given by

$$\begin{aligned}
(A \to B)^+ &= \big\{\langle f, F\rangle \in (A^+ \Rightarrow B^+) \otimes (A^+ \otimes B^- \Rightarrow \mathcal{P}_{\leq 1}(A^-)) \mid \\
&\qquad \forall u \in A^+, y \in B^-.\, \forall x \in F(u,y).\, (A(u,x) \supset B(f(u),y))\big\} \\
(A \to B)^- &= A^+ \otimes B^- \\
(A \to B)(\langle f, F\rangle, \langle u, y\rangle) &\equiv \big(\forall x \in F(u,y).\, A(u,x)\big) \supset B(f(u),y)
\end{aligned}$$

Notice that since $e_0 = \emptyset$ we may reformulate the third clause as

$$(A \to B)(\langle f, F\rangle, \langle u, y\rangle) \equiv F(u,y) = 0 \supset B(f(u),y)$$

because if $F(u,y)$ contains x as its single element then $B(f(u),y)$ follows from $\forall x \in F(u,y).\,(A(u,x) \supset B(f(u),y))$ as ensured by $\langle f, F\rangle \in (A \to B)^+$.

Notice also that there is an alternative way of defining the implication:

$$\begin{aligned}
(A \to B)^+ &= \big\{\langle f, F\rangle \in (A^+ \Rightarrow B^+) \otimes (A^+ \otimes B^- \Rightarrow A^- \oplus 1) \mid \\
&\qquad \forall u \in A^+, y \in B^-.\, (\pi F(u,y) = 0 \wedge A(u, \pi' F(u,y))) \supset B(f(u),y)\big\} \\
(A \to B)^- &= A^+ \otimes B^- \\
(A \to B)(\langle f, F\rangle, \langle u, y\rangle) &\equiv \pi F(u,y) = 1 \supset B(f(u),y)
\end{aligned}$$

This equivalent definition has been explored in [Bie07a] and [Bie07b]

Disjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their disjunction $A_0 \vee A_1$ is given by $(A_0 \vee A_1)^+ = A_0^+ \oplus A_1^+$, $(A_0 \vee A_1)^- = A_0^- \oplus A_1^-$ and $(A_0 \vee A_1)(\langle i, n\rangle, \langle j, m\rangle) \equiv i = j \supset A_i(n,m)$.

Universal Quantification
Suppose $A \in \Sigma_{\mathsf{d}}^I$ and $h \colon I \to J$ in **Set**. Then $\forall_h(A) \in \Sigma_{\mathsf{d}}^J$ is given by

$$\begin{aligned}
\forall_h(A)_j^+ &= \bigcap_{i \in I} \big([h(i) = j] \Rightarrow A_i^+\big) \\
\forall_h(A)_j^- &= \bigcup_{i \in h^{-1}(j)} A_i^- \\
\forall_h(A)_j(u,x) &\equiv \forall i \in h^{-1}(j).\, \big(x \in A_i^- \supset A_i(u \cdot 0, x)\big)
\end{aligned}$$

where $[i = j] \equiv \{0 \mid i = j\}$. For *epic* h this may be simplified as follows

$$\begin{aligned}
\forall_h(A)_j^+ &= \bigcap_{i \in h^{-1}(j)} A_i^+ \\
\forall_h(A)_j^- &= \bigcup_{i \in h^{-1}(j)} A_i^- \\
\forall_h(A)_j(u,x) &\equiv \forall i \in h^{-1}(j).\, \big(x \in A_i^- \supset A_i(u,x)\big)
\end{aligned}$$

Existential Quantification
For arbitrary maps $h : I \to J$ in **Set** and $A \in \Sigma_{\mathsf{d}}^{I}$ existential quantification of A along h is given by

$$\exists_h(A)^+_j = \bigcup_{i \in h^{-1}(j)} A^+_i$$

$$\exists_h(A)^-_j = \bigcap_{i \in h^{-1}(j)} \big(A^+_i \Rightarrow \mathcal{P}_{\leq 1}(A^-_i)\big)$$

$$\exists_h(A)_j(u, f) \equiv \exists i \in h^{-1}(j).\big(u \in A^+_i \wedge \forall x \in f(u).\, A_i(u, x)\big)$$

Lawvere Equality
As a particular case of existential quantification we have for every set I, Lawvere equality $\mathsf{eq}_i \equiv \exists_{\delta_I}(\top)$, which can be simplified as follows

$$\mathsf{eq}_I(i, j) = \big([i = j], \{0\}, [i = j] \otimes \{0\}\big)$$

Generic Family
The generic family of propositions is given by the identity on Σ_{d}.

The Dialectica tripos has an additional fibred closed symmetric monoidal structure, defined as follows. The definition is inspired by the work on Dialectica categories by de Paiva and Hyland [dP89].

Tensor Product The tensor is given by

$$(U_i, X_i, A_i) \otimes (V_i, Y_i, B_i) = (U_i \otimes V_i, X_i \otimes Y_i, A_i \otimes B_i),$$

where

$$(A_i \otimes B_i)(\langle u, v\rangle, \langle x, y\rangle) \iff A_i(u, x) \wedge B_i(v, y).$$

Tensor is symmetric monoidal with

Unit $I = (\{0\}, \{0\}, \{(0, 0)\})$.

Closed Structure The closed structure is given by

$$(U_i, X_i, A_i) \multimap (Z_i, W_i, C_i) = ((U_i \Rightarrow Z_i) \otimes (U_i \otimes W_i \Rightarrow X_i), U_i \otimes W_i, A_i \multimap C_i),$$

where

$$(A_i \multimap C_i)(\langle f, F\rangle, \langle u, w\rangle) \iff A_i(u, F\langle u, w\rangle) \supset C_i(fu, w).$$

Remark 1.1. There is no diagonal $(U, X, A) \vdash (U, X, A) \otimes (U, X, A)$ (we do have projections in those special cases where $\bigcap_{i \in I} X_i \neq \emptyset$).

Proposition 1.2. *The Dialectica tripos is a BI-hyperdoctrine. BI-hyperdoctrines were introduced in [BBTS05].*

Definition 1.3. The *Dialectica topos*, denoted **Dia**, is the topos **Set**[d] obtained by the tripos-to-topos construction (see [HJP80]) from the tripos d.

It is easy to see that a predicate $A \in \Sigma_{\mathsf{d}}^I$ over I is true, i.e. $\top_I \vdash_I A$, iff there exists an $u \in \bigcap_{i\in I} A_i^+$ such that $A_i(u,x)$ for all $i \in I$ and $x \in A_i^-$ and that A is false, i.e. $A \vdash_I \bot$, iff $A_i^+ = \emptyset$ for all $i \in I$.

In d negation is fairly simple as can be seen from the following lemma.

Lemma 1.4. *For $A \in \Sigma_{\mathsf{d}}^I$ its negation $\neg A \equiv A \to \bot$ is isomorphic to $(A_i^+ \Rightarrow \emptyset, \emptyset, \emptyset)$. Thus $\neg\neg A$ is isomorphic to $\big(\{0 \mid A_i^+ \neq \emptyset\}, \emptyset, \emptyset\big)$.*

This gives rise to the following characterization of $\neg\neg$-stable predicates in the tripos d.

Corollary 1.5. *An $A \in \Sigma_{\mathsf{d}}^I$ is $\neg\neg$-stable, i.e. $\neg\neg A \vdash_I A$, if and only if there exists an $u \in \mathbb{N}$ such that for all $i \in I$ if $A_i^+ \neq \emptyset$ then $u \in A_i^+$ with $A_i(u,x)$ for all $x \in A_i^-$.*

Contrary to standard realizability toposes like the effective topos, the Dialectica topos is *not* 2-valued. This holds because the Dialectica tripos is not 2-valued since the preorder Σ_{d} is at least as complicated as the lattice of Turing degrees which can be seen as follows. With every subset P of $\mathbb{N}$ we associate the proposition $A_P = \big(\mathbb{N}, \{0,1\}, \{(n,m) \mid n \in \mathbb{N} \text{ and } (m = 0 \iff n \in P)\}\big)$ in Σ_{d}.

Lemma 1.6. *For $P, Q \subseteq \mathbb{N}$ from $A_P \vdash_1 A_Q$ it follows that $P \leq_T Q$.*

Proof. Suppose $A_P \vdash_1 A_Q$. Then there exist $f : \mathbb{N} \Rightarrow \mathbb{N}$ and $F : \mathbb{N} \otimes \{0,1\} \Rightarrow \mathcal{P}_{\leq 1}(\{0,1\})$ such that $\forall n \in \mathbb{N}, i \in \{0,1\}.\ \big(\forall j \in F(n,i).\, A_P(n,j)\big) \supset A_Q(fn,i)$ and thus (where χ_P and χ_Q are the characteristic predicates of P and Q, respectively)

$$\forall n \in \mathbb{N}.\ \exists j \in F(n, 1-\chi_Q(fn)).\ j = 1-\chi_P(n)$$

from which it folds that there is a program for χ_P in terms of χ_Q, i.e. that $P \leq_T Q$ as desired. □

From this lemma it follows that there exists at least as many propositions in $\Omega_{\mathbf{Dia}}$ as there are Turing degrees.

Finally in this section we show that quantification in the canonically presented tripos d is not standard in the sense of [HJP80]. First we recall the respective definitions from [HJP80]. Let $\mathcal{C}(-,\Sigma)$ be a canonically presented tripos. Let $I = \Sigma/\dashv\vdash$ and $q : \Sigma \to I$ be the quotient map. The tripos is called $\exists$-*standard* iff $q \circ \exists_q(id_\Sigma) = id_I$ and it is called $\forall$-*standard* iff $q \circ \forall_q(id_\Sigma) = id_I$.

Proposition 1.7. *The canonically presented Dialectica tripos d is neither $\exists$-standard nor $\forall$-standard.*

Proof. For $n, m \in \mathbb{N}$ let $A_{n,m}$ be the proposition $\big(\{n\}, \{m\}, \emptyset\}\big)$. All these propositions are equivalent but none of them is equivalent to $\top$.

Let $i_0 := [A_{0,0}]_{\dashv\vdash}$. One easily sees that $\exists_q(id_\Sigma)(i_0) = (\mathbb{N}, \{0\}, \mathbb{N}\times\{0\})$ which is true and thus its equivalence class w.r.t. $\dashv\vdash$ is different from i_0 showing that d is not $\exists$-standard.

One easily sees that $\forall_q(id_\Sigma)(i_0) = (\emptyset, \mathbb{N}, \emptyset)$ which is not equivalent to any of the $A_{n,m}$ showing that d is not $\forall$-standard. □

1.1 Relation of Dia to $\mathbf{Eff}_2$

We first briefly recall the definition of m, the modified realizability tripos, from which the modified realizability topos **Mod** is obtained via the tripos-to-topos construction. The propositions of m are given by

$$\Sigma_{\mathsf{m}} = \{(A^a, A^p) \in \mathcal{P}(\mathbb{N})^2 \mid A^a \subseteq A^p \ni 0\}$$

i.e. a proposition is given by a set A^p of potential realizers containing 0 and a subset $A^a \subseteq A^p$ of actual realizers. For $I \in \mathbf{Set}$ the fibre m^I is defined as $(\Sigma^I_{\mathsf{m}}, \vdash_I)$, where Σ^I_{m} is the set of all functions from I to Σ_{m} and with pre-order $\varphi \vdash_I \psi$ (for $\varphi, \psi \in \mathsf{m}^I$) if $\bigcap_{i\in I} (\varphi^a_i \Rightarrow \psi^a_i) \cap (\varphi^p_i \Rightarrow \psi^p_i)$ is inhabited. The set of propositions of m has a pre-Heyting structure with constants and operations

$$\begin{array}{rcl}
\top & = & (\{0\}, \{0\}) \\
\bot & = & (\emptyset, \{0\}) \\
A \to B & = & \big((A^a \Rightarrow B^a) \cap (A^p \Rightarrow B^p), A^p \Rightarrow B^p\big) \\
A \wedge B & = & (A^a \otimes B^a, A^p \otimes B^p) \\
A \vee B & = & (A^a \oplus B^a, A^p \oplus B^p) \ .
\end{array}$$

The propositional structure is on an arbitrary $(\Sigma^I_{\mathsf{m}}, \vdash_I)$ is given by component-wise application of the operations on Σ_{m}. The quantifiers are given by

$$\begin{array}{l}
\forall_u(\varphi)_j = \big(\ \bigcap_{i\in u^{-1}(j)}\{0\} \Rightarrow \varphi^a_i\ ,\ \bigcap_{i\in u^{-1}(j)}\{0\} \Rightarrow \varphi^p_i\ \big) \\
\exists_u(\varphi)_j = \big(\ succ\big(\bigcup_{i\in u^{-1}(j)} \varphi^a_i\big)\ ,\ \{0\} \cup succ\big(\bigcup_{i\in u^{-1}(j)} \varphi^p_i\big)\ \big)
\end{array}$$

for $u : I \to J$ and $\varphi \in \mathsf{m}^I$. Notice that this description of quantifiers is easily seen to be equivalent to the one of [vO97].

Van Oosten also showed that there is a localic connected geometric morphism $v \colon \mathsf{m} \to \mathsf{e}$, defined by

$$v_*(A^a, A^p) = A^a \qquad v^*(A) = (succ(A), succ(A) \cup \{0\})$$

which lifts to a localic surjective geometric morphism from the modified realizability topos **Mod** to the effective topos, **Eff**.

See [vO97] for a definition of the injective geometric morphisms $e : \mathsf{e} \hookrightarrow \mathsf{e}_2$ and $m : \mathsf{m} \hookrightarrow \mathsf{e}_2$ giving rise to the complemented subtoposes **Eff** and **Mod** of $\mathbf{Eff}_2$.

Proposition 1.8. *There is a fibred adjunction* $p : \mathsf{d} \to \mathsf{e}_2$ *given by*

$$p^*(A_1, A_0) = (A_0, \{0\}, A_1 \times \{0\}) \quad p_*(A) = (\{u \in A^+ \mid \forall x \in A^-.\, A(u, x)\}, A^+)$$

and p^* *is full and faithful.*

Proof. It is straight forward to show that p is a fibred adjunction and that p^* is full and faithful.

□

Remarks 1.9.

(i) The functor p_* does not preserve existential quantification, so it does not have any right adjoints.

(ii) The functor $p^* : \mathsf{e}_2 \to \mathsf{d}$ does *not* preserve conjunction so p is not a geometric morphism of triposes hence, presumably, p^* does not lift to a functor between the induced toposes. To show this claim suppose that p^* does preserve conjunction, that is, for (U_a, U_p) and V_a, V_p in e_2,

$$\begin{array}{lcll} p^*(U^a \times V^a, U^p \times V^p) & = & p^*(U^a, U^p) \wedge^{\mathsf{d}_m} p^*(V^a, V^p) & \text{i.e.} \\ (U^p \otimes V^p, \{0\}, U^a \otimes V^a \otimes \{0\}) & = & (U^p \otimes V^p, \{0,1\}, R) & \text{where} \end{array}$$

$$R = (U^a \otimes V^p \otimes \{0\}) \cup (U^p \otimes V^a \otimes \{1\})$$

and

$$(U^p \otimes V^p, \{0,1\}, R) \vdash (U^p \otimes V^p, \{0\}, U^a \otimes V^a \otimes \{0\})$$

in d, so there exists $f, F \in \mathbb{N}$ such that,

$$f : U^p \otimes V^p \Rightarrow U^p \otimes V^p \text{ and } F : U^p \otimes V^p \otimes \{0\} \Rightarrow \{0,1\}$$

and, for all $u \in U^p, v \in V^p$,

$$R((u,v), F(u,v)) \supset (u,v) \in U^a \otimes V^a$$

that is,

$$\begin{array}{lcccc} F(u,v) = 0 & \wedge & u \in U^a & \Rightarrow & v \in V^a \\ F(u,v) = 1 & \wedge & v \in V^a & \Rightarrow & u \in U^a \end{array}$$

Now let I be the index set $\{1,2\}$, and let $U_i^p = V_i^p = \mathbb{N}$, $V_1^a = \{0\}$, $U_1^a = \mathbb{N}$, $V_2^a = \{1\}$, and $U_2^a = \mathbb{N} \setminus \{k\}$. Then for all $i \in \{1,2\}$, for all $u \in \mathbb{N}, v \in \mathbb{N}$,

$$u \in U_i^a \wedge v \notin V_i^a \Rightarrow F(u,v) = 1 \quad \text{and } u \notin U_i^a \wedge v \in V_i^a \Rightarrow F(u,v) = 0.$$

Since $U_1^a = \mathbb{N}$ and $1 \notin V_1^a$, we get $\forall u \in \mathbb{N}$, $F(u,1) = 1$ and since $k \notin U_2^a$ and $1 \in V_2^a$ we also get $F(k,1) = 0$; contradiction.

The subtoposes **Eff** and **Mod** of **Eff**$_2$ are induced by the local operators $o_U(P) = U \to P$ and $c_U(P) = U \vee P$ where $U = (\emptyset, \{0\})$.

The local operators $o_{p^*(U)}$ and $c_{p^*(U)}$, where $p^*(U) = (\{0\}, \{0\}, \emptyset)$ give rise to the open subtripos d_e and the closed subtripos d_m of d, respectively. We denote the corresponding open and closed subtoposes of **Dia** by **Dia**$_e$ and **Dia**$_m$, respectively. For more details see [Bie08, Chapter 1].

$$\begin{tikzcd} \mathsf{d}_e \arrow[d, hook] & \mathsf{e} \arrow[d, hook] \\ \mathsf{d} \arrow[r, "p"'] & \mathsf{e}_2 \end{tikzcd} \qquad \begin{tikzcd} \mathsf{d}_m \arrow[d, hook] & \mathsf{m} \arrow[d, hook] \\ \mathsf{d} \arrow[r, "p"'] & \mathsf{e}_2 \end{tikzcd}$$

$$\begin{tikzcd} \mathbf{Dia}_e \arrow[d, hook] & \mathbf{Eff} \arrow[d, hook] \\ \mathbf{Dia} & \mathbf{Eff}_2 \end{tikzcd} \qquad \begin{tikzcd} \mathbf{Dia}_m \arrow[d, hook] & \mathbf{Mod} \arrow[d, hook] \\ \mathbf{Dia} & \mathbf{Eff}_2 \end{tikzcd}$$

Notice that $p^*(U) \vee A \vdash_I A$ iff $p^*(U) \vdash_I A$ iff $\bigcap_{i \in I} A_i^+ \neq \emptyset$. Thus, up to isomorphism we may define d_m as canonically presented via $\Sigma_{\mathsf{d}_m} = \{(U, X, R) \in \Sigma_{\mathsf{d}} \mid 0 \in U\}$. The logical operations for d_m are defined as for d, only for $\perp$, $\vee$ and $\exists$ we have to correct the constructions by finally applying the local operator $c_{p^*(U)}$.

1.2 Relation of Dia to Number Realizability

Theorem 1.10. *Let $f = (f^*, f_*) : P \to Q$ be a connected geometric morphism of triposes, the induced geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ is then a surjection, i.e., $\bar{f}^*$ is faithful.*

For a proof see [Bie08, Chapter 1].

There is a connected geometric morphism $r : \mathsf{d} \to \mathsf{e}$ giving rise to a surjective geometric morphism from **Dia** to **Eff**, given by:

$$r_*(U, X, A) = \{u \in U \mid \forall x \in X. A(u, x)\} \qquad r^*(A) = (A, \{0\}, A \times \{0\}).$$

Natural Number Object Since inverse image parts of geometric morphisms preserve natural number objects the nno of **Dia** is given by $N_{\mathbf{Dia}} = r^*(N_{\mathbf{Eff}}) = (\mathbb{N}, \approx_N)$ where $[n \approx_N m] = (\{n \mid n = m\}, \{0\}, \{(n, 0) \mid n = m\})$.

Next we will show that d contains as subtripos the number realizability tripos e from which the effective topos $\mathbf{Eff} = \mathbf{Set}[\mathsf{e}]$ arises via the tripos-to-topos construction. Recall that $\Sigma_{\mathsf{e}} = \mathcal{P}(\mathbb{N})$ and $\varphi \vdash_I \psi$ iff $\bigcap_{i \in I} [\varphi_i \Rightarrow \psi_i]$ is inhabited. We can draw the following diagram:

$$\begin{tikzcd}
\mathbf{Dia} \arrow[rr, phantom, "\simeq"] \arrow[dr, "r"'] & & \mathbf{Dia} \\
& \mathbf{Eff} \arrow[ur, hook, "j"'] &
\end{tikzcd}$$

Proposition 1.11. *There is an injective geometric morphism $j : \mathsf{e} \to \mathsf{d}$ and, accordingly,* **Eff** *is a subtopos of* **Dia**. *This geometric inclusion arises from the* open *topology $p^*(U) \to (-)$ on* **Dia** *where $p^*(U) = (\{0\}, \{0\}, \emptyset)$. Moreover, one obtains* **Set** *as the subtopos of $\neg\neg$-sheaves of* **Dia**.

Proof. We define $j : \mathsf{e} \to \mathsf{d}$ as the fibred adjunction

$$j^*(A^+, A^-, A) = A^+ \qquad \text{and} \qquad j_*(X) = (X, \emptyset, \emptyset)$$

and one easily checks that j^* preserves $\top$ and also conjunction in each fibre.

One easily checks that $j_* j^*(A^+, A^-, A) = (A^+, \emptyset, \emptyset) \cong p^*(U) \to (A^+, A^-, A)$ from which it follows that the geometric inclusion j is induced by the topology $p^*(U) \to (-)$ on **Dia**. Thus the open subtripos d_e of d as induced by $o_{p^*(U)}$ is isomorphic to e.

Hence we have

$$\begin{tikzcd}
\mathbf{Eff} \cong \mathbf{Dia}_e \arrow[d, hook] \\
\mathbf{Dia}
\end{tikzcd}$$

Notice that $p^*(U) \to A$ in d is given (up to isomorphism) by

$$(p^*(U) \to A)^+ = A^+ \otimes \left(A^- \Rightarrow \mathcal{P}_{\leq 1}(\{0\})\right)$$
$$(p^*(U) \to A)^- = A^-$$
$$(p^*(U) \to A)(\langle u, f \rangle, x) \equiv f(x) = \emptyset \supset A(u, x)$$

which in turn is isomorphic to $(A^+, \emptyset, \emptyset)$. Thus it is easy to see that $\neg\neg(p^*(U) \to A) = \neg\neg A$.

It is well known that **Set** $\simeq$ **Eff**$_{\neg\neg}$. Thus, since $\neg\neg(p^*(U) \to A) = \neg\neg A$ it follows that **Set** $\simeq$ **Eff**$_{\neg\neg}$ $\simeq$ **Dia**$_{\neg\neg}$.

□

In general direct image parts of geometric morphisms do not preserve natural numbers objects. But in the particular case $j\colon \mathbf{Eff} \to \mathbf{Dia}$ one readily checks that $j_*(N) = (\mathbb{N}, E_N)$ with $E_N(n,m) = (\{n \mid n = m\}, \emptyset, \emptyset)$ happens to be isomorphic to the natural numbers object in **Dia**.

Proposition 1.12. *The direct image functor $j_* : \mathbf{Eff} \to \mathbf{Dia}$ preserves the finite type structure of the natural numbers object in* **Eff**.

Proof. We have already seen that $j_*(N_{\mathbf{Eff}}) \cong N_{\mathbf{Dia}}$, and since j_* preserves products, the statement obviously holds for products. To see that $j_*(X^Y) \cong (j_*X)^{(j_*Y)}$ consider the following string of bi-implications:

$$\begin{array}{lcl}
A & \vdash & j_*(X^Y) \\ \hline
j^*(A) & \vdash & X^Y \\ \hline
j^*(A) \times Y & \vdash & X \\ \hline
j^*(A) \times j^*j_*(Y) & \vdash & X \\ \hline
A \times j_*(Y) & \vdash & j_*(X) \\ \hline
A & \vdash & j_*(X)^{j_*(Y)}
\end{array}$$

So in particular we have $N_{\mathbf{Dia}}^{N_{\mathbf{Dia}}} \cong j_*(N_{\mathbf{Eff}}^{N_{\mathbf{Eff}}})$. □

Proposition 1.13. *The inverse image j^* preserves first-order logic and the finite type structure of the natural number object in* **Dia**.

Proof. By the above proposition the inclusion is open, *i.e.*, the inverse image preserves first-order logic. It thus suffices to show that the inverse image also preserves the finite type structure, which obviously holds for products. We prove inductively the stronger statement that the inverse image functor j^* preserves the finite type structure over the natural numbers, and that every such type in **Dia** is (up to isomorphism) in the image of the direct image functor, which completes the proof.

The statement is obviously true in the base case, which is the type of natural numbers. Moreover, since both inverse image and direct image preserve finite products, the statement holds true for products. For exponentials we consider X and Y in **Eff**, and note that $j^*((j_*X)^{(j_*Y)}) \cong j^*j_*(X^Y) \cong X^Y \cong (j^*j_*X)^{(j^*j_*Y)}$, which completes the proof. □

1.3 First Order Logic in Dia

In this section we record some results about the logic in **Dia**.

Proposition 1.14. **Dia** $\models \phi$ *implies* **Eff** $\models \phi$ *for* ϕ *a formula of HA.*

Proof. This follows immediately from Proposition 1.13. □

Recall that a formula is called negative if it is built up from atomic formulas using $\wedge, \rightarrow, \forall$. Following results in [Hyl82] we have

Theorem 1.15. *If equality on an object* $(I, \simeq)$ *in* **Dia** *is* j*-closed (where* $j = p^*(U) \rightarrow (-)$*) and* ϕ *is a negative formula of first order logic with* $\|\phi\|_{\mathbf{Dia}} \rightarrow (I, \simeq)$*, then interpretation of* ϕ *in* **Eff** *agrees with the interpretation of* ϕ *in* **Dia** *in the sense that*

$$j_*(\|\phi\|_{\mathbf{Eff}}) = \|\phi\|_{\mathbf{Dia}}$$

2 Modified Dialectica Tripos $\mathbf{Dia}_m$

As we saw in Section 1.1 we have a modified version of the Dialectica tripos, d_m which may be defined as canonically presented via $\Sigma_{\mathsf{d}_m} = \{(U, X, R) \in \Sigma_{\mathsf{d}} \mid 0 \in U\}$. The logical operations for d_m are defined as for d, only for $\bot$, $\vee$ and $\exists$ we have to correct the constructions by finally applying the local operator $c_{p^*(U)}$.

For convenience we now give the concrete definition of d_m, for those logical operations that differ from d.

Over set I, the preorder is defined to consist of families of triples (U_i, X_i, A_i) satisfying that $0 \in U_i \subseteq \mathbb{N}$, $X_i \subseteq \mathbb{N}$, and $A_i \subseteq U_i \times X_i$. We declare that

$$(U_i, X_i, A_i) \vdash (V_i, Y_i, B_i)$$

iff there exist $f, F \in \mathbb{N}$ such that, for all $i \in I$,

$$f \in U_i \Rightarrow V_i \qquad \text{and} \qquad F \in U_i \otimes Y_i \Rightarrow \mathcal{P}_{\leq 1}(X_i)$$

and, moreover, for all $u \in U_i$, $y \in Y_i$,

$$(\forall z \in F(u, y)\, A_i(u, z) \supset B_i(f \cdot u, y).$$

Falsity
The initial object $\bot$ is given by

$$\bot = i \mapsto \big(\{0\}, \{0\}, \emptyset\big).$$

Disjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their disjunction $A_0 \vee A_1$ is given by $(A_0 \vee A_1)^+ = A_0^+ \oplus A_1^+$, $(A_0 \vee A_1)^- = A_0^- \oplus A_1^-$ and $(A_0 \vee A_1)(\langle i, n\rangle, \langle j, m\rangle) \equiv i = j \supset A_i(n, m)$.

Existential Quantification
Suppose $h \colon I \rightarrow J$ in **Set** *is epic.* Then

$$\exists_h(A)^+_j = \bigcup_{i \in h^{-1}(j)} A^+_i$$

$$\exists_h(A)^-_j = \bigcap_{i \in h^{-1}(j)} \big(A^+_i \Rightarrow \mathcal{P}_{\leq 1}(A^-_i)\big)$$

$$\exists_h(A)_j(u,f) \equiv \exists i \in h^{-1}(j).(u \in A^+_i \wedge \forall x \in f(u).\, A_i(u,x))$$

In case h *is not epic*,

$$\exists_h(A)^+_j = \{0\} \cup succ\big(\bigcup_{i \in h^{-1}(j)} A^+_i\big)$$

$$\exists_h(A)^-_j = \bigcap_{i \in h^{-1}(j)} \big(A^+_i \Rightarrow \mathcal{P}_{\leq 1}(A^-_i)\big)$$

$$\exists_h(A)_j(u,f) \equiv u \neq 0 \text{ and } \exists i \in h^{-1}(j).(u-1 \in A^+_i \wedge \forall x \in f(u-1).\, A_i(u-1,x))$$

Lawvere Equality
We have for every set I, Lawvere equality, $\mathsf{eq}_i \equiv \exists_{\delta_I}(\top)$, which can be simplified as follows

$$\mathsf{eq}_I(i,j) = \big(\{0\} \cup succ([i=j]), \{0\}, succ([i=j]) \otimes \{0\}\big)$$

The generic object is, of course, the identity function on the set

$$\Sigma = \{\, (X,Y,A) \mid A \subseteq X \times Y, \text{ and } X, Y \subseteq \mathbb{N}, 0 \in X \,\}.$$

We also have

Tensor Product The tensor is given by

$$(U_i, X_i, A_i) \otimes (V_i, Y_i, B_i) = (U_i \otimes V_i, X_i \otimes Y_i, A_i \otimes B_i),$$

where

$$(A_i \otimes B_i)(\langle u,v\rangle, \langle x,y\rangle) \iff A_i(u,x) \wedge B_i(v,y).$$

Tensor is symmetric monoidal with

Unit $I = (\{0\}, \{0\}, \{(0,0)\})$.

Closed Structure The closed structure is given by

$$(U_i, X_i, A_i) \multimap (Z_i, W_i, C_i) = ((U_i \Rightarrow Z_i) \otimes (U_i \otimes W_i \Rightarrow X_i), U_i \otimes W_i, A_i \multimap C_i),$$

where

$$(A_i \multimap C_i)(\langle f, F\rangle, \langle u,w\rangle) \iff A_i(u, F\langle u,w\rangle) \supset C_i(fu, w).$$

Remark 2.1. There is no diagonal $(U,X,A) \vdash (U,X,A) \otimes (U,X,A)$, but we do have projections.

13

Proposition 2.2. *The Dialectica tripos is a BI-hyperdoctrine. BI-hyperdoctrines were introduced in [BBTS05]. It is affine, i.e., $I \equiv \top$.*

Lemma 2.3. *The negation of (X_i, Y_i, A_i), defined by*

$$\neg(X_i, Y_i, A_i) \overset{\text{def}}{=} (X_i, Y_i, A_i) \to \bot_I,$$

is isomorphic to

$$\big(\{\, F\colon X_i \Rightarrow 1 \oplus Y_i \mid \forall x \in X_i.\ \pi(F(x)) = 1 \supset \neg A_i(x, \pi'(F(x)))\,\}, X_i, A_i \to \emptyset\big),$$

where

$$(A_i \to \emptyset)(F, x) \iff \pi(F(x)) = 1.$$

It is easy to see that, over 1, a predicate (X, Y, A) is (isomorphic to) true just in case there exists an x in X such that for all y in Y we have that $(x, y) \in A$ and that a predicate (X, Y, A) is (isomorphic to) false just in case there exists an $f \in \mathbb{N}$ such that $f \in X \Rightarrow Y$ and such that for all $x \in X$, $(x, f \cdot x) \notin A$. We now characterize when a predicate (X_i, Y_i, A_i) in the fibre over I in d is true (for $I \neq \emptyset$). That holds iff

$$\forall_I(X_i, Y_i, A_i) = \top_1,$$

i.e., iff

$$\Big(\bigcap_i X_i, \bigcup_i Y_i, \tilde{A}\Big) = \top_1,$$

where

$$\tilde{A}(x, y) \text{ iff for all } i \in I.y \in Y_i \Rightarrow A_i(x, y)$$

i.e., iff

$$\exists x \in \bigcap_i X_i.\ \forall i \in I.\ \forall y \in Y_i.\ y \in Y_i \Rightarrow A_i(x, y).$$

Lemma 2.4. *Let $p = (X, U, R)$ be a proposition with $U = \{0\}$ and R a decidable predicate on $X \times \{0\}$. Then $\neg\neg p \vdash p$*

Proof. Due to the above explicitation of negation we have that $(\neg p)^+$ consists $e : p^+ \to \{0\} + \{0\}$ such that $\forall a \in p^+ . \pi(ea) = 1 \Rightarrow \neg p(a, 0)$, $(\neg p)^- = p^+$ and $(\neg p)(e, a) \equiv \pi(ea) = 1$. Accordingly, $(\neg\neg p)^+$ consists of all $f \in (\neg p)^+ \to \{0\} + p^+$ such that $\forall e \in (\neg p)^+ . \pi(fe) = 1 \Rightarrow \pi(e \cdot \pi'(f \cdot e)) = 0$, $(\neg\neg p)^+ = (\neg p)^+$ and $(\neg\neg p)(f, e) \equiv \pi(f \cdot e) = 1$.

Let $n_0 \in p^+ \to \{0\} + \{0\}$ with $\pi(n_0 \cdot a) = 1$ iff $p(a, 0)$ for all $a \in p^+$. Such an n_0 exists as $p(\_, 0)$ is decidable by assumption and, obviously, it is an element of $(\neg p)^+$ by construction. Now $\neg\neg p \vdash p$ is realized by

$$e^+ = \Lambda f.\, \pi'(f \cdot n_0) \qquad \text{and} \quad e^- = \Lambda z.\, n_0$$

as it holds that

$$\forall f \in (\neg\neg p)^+ . \pi(f \cdot n_0) = 1 \Rightarrow p(\pi'(f \cdot n_0), 0)$$

which follows immediately from the expansions of the statements $f \in (\neg\neg p)^+$ and $n_0 \in (\neg p)^+$. ☐

Definition 2.5. The **Modified Dialectica Topos**, denoted $\mathbf{Dia}_m$, is the topos $\mathbf{Set}[\mathsf{d}_m]$ obtained by the tripos-to-topos construction (see [HJP80]) from the tripos d_m.

Lemma 2.6. *The Dialectica tripos is not 2-valued. Hence the Dialectica topos is not 2-valued either.*

Proof. Then $(\mathbb{N}, \mathbb{N}, A)$ with

$$A = \{ (n,m) \mid n \in \mathbb{N} \wedge m = \begin{cases} 1 & \text{if } n \cdot n \downarrow \\ 0 & \text{if } n \cdot n \uparrow \end{cases} \}$$
$$\cup \{ (n,m) \mid n \in \mathbb{N} \wedge m \geq 2 \}$$

is neither true, nor false. $\square$

Indeed, we can show that the structure of the subobject classifier Ω in $\mathbf{Dia}_m$ is very rich: all many-one reducibility degrees are represented. To see this, let us, for an $\alpha \subset \mathbb{N}$, set

$$A_\alpha = \text{Graph}(\chi_\alpha) \cup \{ (n,m) \mid m \geq 2 \},$$

where χ_α is the characteristic function of α.

Proposition 2.7. *If $\alpha \leq_m \beta$ (*i.e.*, α is m-reducible to β), then $(\mathbb{N}, \mathbb{N}, A_\alpha) \vdash (\mathbb{N}, \mathbb{N}, A_\beta)$.*

Proof. By definition $(\mathbb{N}, \mathbb{N}, A_\alpha) \vdash (\mathbb{N}, \mathbb{N}, A_\beta)$ iff there exist $f \in \mathbb{N} \Rightarrow \mathbb{N}$ and $F \in \mathbb{N} \otimes \mathbb{N} \Rightarrow \mathbb{N}$ such that

$$A_\alpha(n, F\langle n,m\rangle) \supset A_\beta(fn, m),$$

which amounts to

$$F\langle n,m\rangle \geq 2 \supset (m \geq 2 \vee m = \chi_\beta(fn))$$

or

$$F\langle n,m\rangle = \chi_\alpha(n) \supset (m \geq 2 \vee m = \chi_\beta(fn))$$

By the assumption $\alpha \leq_m \beta$ there exists a $g \in \mathbb{N} \Rightarrow \mathbb{N}$ such that $\chi_\alpha(n) = \chi_\beta(gn)$. Hence we can simply let $f = g$ and $F\langle n,m\rangle = m$. $\square$

From the proof it is easy to see that, if $\alpha \leq_m \beta$, we will in general not have the converse of $(\mathbb{N}, \mathbb{N}, A_\alpha) \vdash (\mathbb{N}, \mathbb{N}, A_\beta)$, *i.e.*, the many-one reducibility degrees are not collapsed when viewed as global elements of Ω.

Lemma 2.8. *The d_m tripos is not $\exists$-standard.*

Proof. By the definition of $\exists$-standard, d is such iff, for all $\varphi \in \Sigma$, $\varphi \dashv\vdash \exists_{q(\varphi')=q(\varphi)}\varphi'$. Now consider $\varphi = (\mathbb{N}, \{1\}, \emptyset)$, which is not true. Then, for all $n \in \mathbb{N}$, $\varphi \dashv\vdash \varphi_n$ with $\varphi_n = (\mathbb{N}, \{n\}, \emptyset)$. But $\exists_q(id_\Sigma)(\phi) = (\mathbb{N}, \{0\}, \mathbb{N} \otimes \{0\})$, which is true, and if there are more elements $\varphi'' \in \Sigma$ for which $\varphi \dashv\vdash \varphi''$, then $\exists_{q(\varphi)=q(\varphi')}\varphi'$ will still be $(\mathbb{N}, \{0\}, \mathbb{N} \otimes \{0\})$. Thus $\exists_{q(\varphi')=q(\varphi)}\varphi'$ is true, but φ is not and hence the tripos is not $\exists$-standard. $\square$

Lemma 2.9. *The d_m tripos is not $\forall$-standard.*

Proof. Consider $\psi_n = (\{0,n\}, \{0,n\}, \{(n,0), (n,n)\})$, which is true for all n, but $(\bigcap\{0,n\}, \bigcup\{0,n\}, \tilde{\psi}) = (\{0\}, \mathbb{N}, \emptyset)$, which is equivalent to $\bot$. So $a[\top] = (\{0\}, \mathbb{N}, \{(0,0)\}) \not\dashv\vdash \top$. $\square$

2.1 Relation of $\mathbf{Dia}_m$ to Number Realizability

Proposition 2.10. *There is a connected geometric morphism from* d_m *to* e. *Hence, by Theorem 1.10 there is a surjective geometric morphism from* $\mathbf{Dia}_m$ *to* $\mathbf{Eff}$.

Proof. By composition of the adjunction from Proposition 2.23 and the geometric morphism $v : \mathsf{m} \to \mathsf{e}$ from [vO97]. Explicitly, the geometric morphism $H = (H^*, H_*) \colon \mathsf{d}_m \to \mathsf{e}$ is defined over 1 by

$$\begin{aligned} H^*(A) &= (succ(A) \cup \{0\}, \{0\}, succ(A) \times \{0\}), \\ H_*(X, Y, A) &= \{\, x \in X \mid \forall y \in Y.\ A(x, y)\,\}. \end{aligned}$$

We have to check that H^* preserves finite limits. It is straight forward to verify that it preserves $\top$, we now show that H^* preserves conjunction. Let A_i, B_i in e be given, we must show that

$$H^*(A_i \wedge^{\mathsf{e}} B_i) \dashv\vdash H^*(A_i) \wedge^{\mathsf{d}_m} H^*(B_i)$$

in d_m. We have

$$H^*(A_i \wedge B_i) = (succ(A_i \otimes B_i) \cup \{0\}, \{0\}, succ(A_i \otimes B_i) \times \{0\}) \tag{1}$$

and

$$H^*(A_i) \wedge H^*(B_i) = ((succ(A_i) \cup \{0\}) \otimes (succ(B_i) \cup \{0\}), \{0, 1\}, R_i), \tag{2}$$

where

$$R_i = (succ(A_i) \otimes (\{0\} \cup succ(B_i)) \otimes \{0\}) \cup ((\{0\} \cup succ(A_i)) \otimes succ(B_i) \otimes \{1\}).$$

To see that $1 \vdash 2$, we must find

$$\begin{array}{lcll} f : succ(A_i \otimes B_i) \cup \{0\} & \Rightarrow & (succ(A_i) \cup \{0\}) \otimes (succ(B_i) \cup \{0\}) & \text{and} \\ F : (succ(A_i \otimes B_i) \cup \{0\}) \otimes \{0, 1\} & \Rightarrow & \mathcal{P}_{\leq 1}(\{0\}) & \end{array}$$

such that, for all i, $n \in succ(A_i \otimes B_i) \cup \{0\}$, $j \in \{0, 1\}$,

$$n \in succ(A_i \otimes B_i) \supset R_i(f\langle n\rangle, j).$$

Define f as follows:

$$\begin{array}{lcl} f(\langle a, b\rangle + 1) & = & \langle a + 1, b + 1\rangle \\ f(0) = 0. & & \end{array}$$

And

$$F(\langle a, b\rangle, i) = \{0\}.$$

To see that $(2) \vdash (1)$, we must find

$$\begin{array}{lcll} g : (succ(A_i) \cup \{0\}) \otimes (succ(B_i) \cup \{0\}) & \Rightarrow & succ(A_i \otimes B_i) \cup \{0\} & \text{and} \\ G : (succ(A_i) \cup \{0\}) \otimes (succ(B_i) \cup \{0\}) & \Rightarrow & \{0, 1\} & \end{array}$$

such that for all i,

$$R_i(\langle n, m\rangle, G(\langle n, m\rangle)) \Rightarrow g(\langle n, m\rangle) \in succ(A_i \otimes B_i).$$

To this end, we define

$$g(\langle n,m\rangle) = \begin{cases} \langle n-1, m-1\rangle + 1 & \text{if } n \neq 0, \text{ and } m \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$G(\langle n,m\rangle) = \begin{cases} 1 & \text{if } n \neq 0, \text{ and } m = 0 \\ 0 & \text{otherwise} \end{cases}$$

□

Natural Number Object Since inverse image parts of geometric morphisms preserve natural number objects the nno of $\mathbf{Dia}_m$ is given by $N_{\mathbf{Dia}_m} = H^*(N_{\mathbf{Eff}}) = (\mathbb{N}, \approx_N)$ where $[n \approx_N m] = (\{0\} \cup \{n+1 \mid n = m\}, \{0\}, \{(n+1, 0) \mid n = m\})$.

Here comes a useful observation about strict predicates over $N^k_{\mathbf{Dia}_m}$.

Proposition 2.11. *Let A be a strict predicate over $N_{\mathbf{Dia}_m}$. Then there are recursive functions*

$$\mu : \bigcap_{n:\mathbb{N}} (A^+_n \Rightarrow \{0, n+1\}) \qquad \delta : \bigcap_{n:\mathbb{N}} (A^+_n \Rightarrow A^-_n)$$

such that

- *If $\mu(u) = n+1$ then*
  - *$u \in A^+_n$ and $u \notin A^+_k$ for $k \neq n$*
  - *$\neg A_k(u, \delta(u))$ for all $k \neq n$.*
- *If $\mu(u) = 0$ then $\neg A_n(u, \delta(u))$ for all n.*
- *$A_n(u, \delta(u))$ implies $\mu(u) = n+1$.*

Proof. That A_n is strict means that $(A^+_n, A^-_n, A_n) \vdash [n = n]$ in d_m, and this means that there are recursive functions

$$\mu : \bigcap_{n:\mathbb{N}} (A^+_n \Rightarrow \{0, n+1\}) \qquad \delta : \bigcap_{n:\mathbb{N}} (A^+_n \Rightarrow A^-_n)$$

such that

$$\forall n : \mathbb{N}, u \in A^+_n.\, A_N(u, \delta(u)) \supset \mu(u) = n+1.$$

□

It is a well known fact that any geometric morphism $f : P \to Q$ of $\mathbb{C}$-triposes P and Q lifts to a geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ (see [HJP80]). It is also known that if f is an inclusion, then so is $\bar{f}$ (see e.g. [vO08]). In his thesis [Bir99], Lars Birkedal shows that if f is connected and if f_* preserves $\exists$, then $\bar{f}$ is also connected. It is not in general the case that a connected geometric morphism of triposes lifts to a connected geometric morphism of toposes. For a counter example, see [Fre07].

In [Bie08, Section 2] a "local" version of this result is shown, local in the sense that if f_* only preserves $\exists_h$ for certain morphisms h, then f^* is fully faithful for certain homsets.

Definition 2.12. Let $\mathbb{C}, \mathbb{D}$ be categories and X, Y objects of $\mathbb{C}$. We say that a functor $H : \mathbb{C} \to \mathbb{D}$ is fully faithful on (X, Y) if there is a bijection $\mathrm{Hom}_{\mathbb{C}}(X, Y) \cong \mathrm{Hom}_{\mathbb{D}}(HX, HY)$.

Theorem 2.13. *Suppose $f : P \to Q$ is a connected geometric morphism of $\mathbb{C}$-triposes and let $\pi : X \times J \to J$ be a projection in $\mathbb{C}$. Suppose further that for any strict predicate $\phi(x)$ over an object $\bar{f}^*(X, \sim)$, $(X, \sim)$ in $\mathbb{C}[Q]$, we have $f_* \exists^P_\pi(\phi(x)) \cong \exists^Q_\pi f_*(\phi(x))$. Then the induced geometric morphism $\bar{f} : \mathbb{C}[P] \to \mathbb{C}[Q]$ satisfies that $\bar{f}^*$ is fully faithful on $((Y, \sim), (X, \sim))$ for any object $(Y, \sim)$ in $\mathbb{C}[Q]$.*

For a proof, see [Bie08, Chapter 1].

Lemma 2.14. *Let $\pi : \mathbb{N} \times J \to J$, and let $A \in \mathsf{d}_m(\mathbb{N})$ be a strict predicate over $(\mathbb{N}, \sim) = N_{\mathbf{Dia}_m}$, then*

$$H_* \exists^{\mathsf{d}}_\pi(A) \cong \exists^{\mathsf{e}}_\pi H_*(A).$$

Proof. Let $A = (U, X, A)$, then

$$(H_* \exists^{\mathsf{d}}_\pi(U, X, A))_j = \{u \in \bigcup_{n \in \mathbb{N}} U(n, j) \mid \forall f \in \bigcap_{\mathbb{N}} (U(n, j) \Rightarrow \mathcal{P}_{\leq 1}(X(n, j)).\hat{A}_j(u, f))\}$$

where

$$\begin{aligned} \hat{A}_j(u, f) &\equiv \exists n : \mathbb{N}.(u \in U(n, j) \wedge \forall x \in f(u).A(n, j)(u, x)). \\ &= u \in \textstyle\bigcup_{n \in \mathbb{N}} U(n, j).\forall x \in \bigcap_{\{n \mid u \in U(n,j)\}} X(n, j).\exists n : \mathbb{N}.(u \in U(n, j) \wedge A(n, j)(u, x)). \end{aligned}$$

On the other hand

$$(\exists^{\mathsf{e}}_\pi H_*(U, X, A))_j = \bigcup_{n \in \mathbb{N}} \{u \in U(n, j) \mid \forall x.X(n, j).A(n, j)(u, x)\}$$

Clearly

$$(\exists^{\mathsf{e}}_\pi H_*(U, X, A))_j \vdash (H_* \exists^{\mathsf{d}}_\pi(U, X, A))_j$$

to show the other direction we use the fact that A is strict over $(\mathbb{N}, \sim)$ so that for $u \in \bigcup_{n \in \mathbb{N}} U(n, j)$, $\mu(u) = n + 1$ implies $u \in U(n, j)$ and for for all $k \neq n$, $u \notin U(k, j)$. And $\mu(u) = 0$ implies for all n. $\neg A(n, j)(u, \delta(u))$. Now suppose $u \in (H_* \exists^{\mathsf{d}}_\pi(U, X, A))_j$, if $\mu(u) = n + 1$, then $u \in U(n, j)$ for n only, so $\bigcap_{\{k \mid u \in U(k,j)\}} X(k, j) = X(n, j)$, so $u \in (\exists^{\mathsf{e}}_\pi H_*(U, X, A))_j$. If $\mu(u) = 0$ we have $\delta \in \bigcap_{n \in \mathbb{N}}(U(n, j) \Rightarrow X(n, j))$ with $\neg \hat{A}_j(u, \delta(u))$, so $u \notin (H_* \exists^{\mathsf{d}}_\pi(U, X, A))_j$. ∎

Corollary 2.15. *Maps $F : \bar{H}^*(X, \sim) \to \bar{H}^*(\mathbb{N}, \sim) = N_{\mathbf{Dia}_m}$ in $\mathbf{Dia}_m$ are in bijective correspondence with maps from $(X, \sim)$ to $(\mathbb{N}, \sim)$ in $\mathbf{Eff}$.*

2.2 Decidable Predicates in the Modified Dialectica Topos

In the Dialectica interpretation there is a side condition saying that atomic (and hence all quantifier-free) predicates must be decidable. We do not have this condition as part of the tripos. It is the $(\otimes, \multimap)$ structure of the tripos that corresponds to Gödel's interpretation of conjunction and implication. In this section we show that when restricting to the decidable fragment of $N_{\mathbf{Dia}_m}$,

$(\otimes, \multimap)$ coincides with $(\wedge, \to)$, so that conjunction and implication is exactly like Gödel's definition.

Recall that a predicate $x\colon X \mid \varphi\colon \mathsf{Prop}$ is *decidable* if $\top \vdash \varphi \vee \neg\varphi$ holds in the fibre over X.

Proposition 2.16. *Let $(X, \approx)$ be an object of* **Eff**. *There is an isomorphism between decidable subobjects of $H^*(X, \approx)$ in $\mathbf{Dia}_m$ and decidable subobjects of $(X, \approx)$ in* **Eff**.

Proof. The object 2 acts as a classifier for decidable subobjects, so decidable subobjects of $(X, \approx)$ are in one-to-one correspondence with maps from $(X, \approx)$ to 2. Now $2_{\mathsf{d}_m} = H^*(2_{\mathsf{e}})$, and so the proposition holds because 2_{e} satisfies the condition of Theorem 2.13, so that H^* is full and faithful on $((X, \approx), 2_{\mathsf{e}})$ for any object $(X, \approx)$ of **Eff**. ☐

A subobject in the Dialectica topos on an object $(X, \approx)$ is a strict and extensional predicate in the Dialectica tripos in the fibre over X. Thus a predicate $(X_n, Y_n, A_n)_{n\in\mathbb{N}}$ in the fibre over the natural numbers object N in $\mathbf{Dia}_m$ is decidable iff

$$n\colon \mathbb{N} \mid E_N(n) \vdash (X_n, Y_n, A_n) \vee \neg(X_n, Y_n, A_n) \tag{3}$$

holds in d_m in the fibre over $\mathbb{N}$. Recalling the definitions we see that, for a given $n \in \mathbb{N}$, the left hand side in (3) is equal to

$$\big(\{n+1, 0\}, \{0\}, \{(n+1, 0)\}\big),$$

and the right hand side is equal to

$$(Z_n, Y_n \oplus X_n, C),$$

where

$$\begin{aligned}
&Z_n = X_n \oplus Z'_n \\
&Z'_n = \{\, F \in X_n \Rightarrow 1 \oplus Y_n \mid \forall x \in X_n.\ \pi(Fx) = 1 \supset \neg A_n(x, \pi'(Fx)) \,\} \\
&C_n = \{\, \langle 0, x\rangle, \langle o, y\rangle \mid A_n(x, y) \,\} \\
&\qquad \cup \{\, \langle 1, F\rangle, \langle 1, x'\rangle \mid \pi(Fx') = 1 \wedge \neg A_n(x', \pi'(Fx')) \,\}.
\end{aligned}$$

Thus (3) holds iff there exist $g, G \in \mathbb{N}$, such that for all $n \in \mathbb{N}$,

$$\begin{gathered}
g \in \{n+1, 0\} \Rightarrow Z_n \\
G \in \{n+1, 0\} \otimes (Y_n \oplus X_n) \Rightarrow \{0\}
\end{gathered}$$

and

$$\pi(gn) = 0 \wedge \forall y \in Y_n.\ A_n(\pi'(gn), y)$$

or

$$\pi(gn) = 1 \wedge \exists F \in X_n \Rightarrow Y_n.\ \forall x_0 \in X_n.\ \neg(A_n(x_0, Fx_0)).$$

In words, $(X_n, Y_n, A_n)_{n\in\mathbb{N}}$ is decidable in the Dialectica topos just in case there is a g which, for all n, gives a pair $\langle 0, x_0\rangle$ or $\langle 1, F\rangle$, with x_0 a witness for the predicate being true at n, and F a witness for the predicate being false at n.

Remark 2.17. For a any predicate $\psi : I \to \Sigma$ in d_m in the fibre over I, we can associate the subset $S(\psi) = \{\, i \in I \mid \psi(i) \dashv\vdash \top \,\}$ of I. For predicates over the natural numbers, we note that if ψ is decidable then $S(\psi)$ is a recursive set.

Recall that the Dialectica tripos has an additional fibred closed symmetric monoidal structure, defined as follows (compare the work on Dialectica categories by de Paiva and Hyland [dP89]), making the Dialectica tripos into a *BI-hyperdoctrine*, see [BBTS05]. The tensor is given by

$$(X_i, Y_i, A_i) \otimes (U_i, V_i, B_i) = (X_i \otimes U_i, Y_i \otimes V_i, A_i \otimes B_i),$$

where $(A_i \otimes B_i)(\langle x, u\rangle, \langle y, v\rangle)$ is defined by $A_i(x, y) \wedge B_i(u, v)$. Tensor is symmetric monoidal with $I = \top$ as the unit. The closed structure is given by

$$(X_i, Y_i, A_i) \multimap (Z_i, W_i, C_i) = ((X_i \Rightarrow Z_i) \otimes (X_i \otimes W_i \Rightarrow Y_i), X_i \otimes W_i, A_i \multimap C_i),$$

where $(A_i \multimap C_i)(\langle f, F\rangle, \langle x, w\rangle)$ is defined by $A_i(x, F\langle x, w\rangle) \supset C_i(fx, w)$. It is easy to see that tensor has projections. However, there are no diagonals.

Proposition 2.18. *Let (X_n, Y_n, A_n) and (U_n, V_n, B_n) be decidable predicates over the NNO N in $\mathbf{Dia}_m$. Then*

$$(X_n, Y_n, A_n) \otimes (U_n, V_n, B_n) \dashv\vdash (X_n, Y_n, A_n) \wedge (U_n, V_n, B_n)$$
$$(X_n, Y_n, A_n) \multimap (U_n, V_n, B_n) \dashv\vdash (X_n, Y_n, A_n) \to (U_n, V_n, B_n)$$

both hold in the topos $\mathbf{Dia}_m$ (in the fibre over N).

Proof. The truth of the bi-implications in the topos reduce to the truth of the following bi-implications in the d_m tripos (in the fibre over $\mathbb{N}$):

$$n \approx_N n \vdash (X_n, Y_n, A_n) \otimes (U_n, V_n, B_n) \leftrightarrow (X_n, Y_n, A_n) \wedge (U_n, V_n, B_n) \qquad (4)$$
$$n \approx_N n \vdash (X_n, Y_n, A_n) \multimap (U_n, V_n, B_n) \leftrightarrow (X_n, Y_n, A_n) \to (U_n, V_n, B_n). \qquad (5)$$

Since (X_n, Y_n, A_n) and (U_n, V_n, B_n) are strict predicates over the natural numbers object, there exists a function $F : \bigcap_n (X_n \Rightarrow Y_n)$, so $F(0) \in Y_n$ for all n, this is needed to show the implication from left to right of (4) and (5), so it suffices to show

$$n \approx_N n \wedge (X_n, Y_n, A_n) \wedge (U_n, V_n, B_n) \vdash (X_n, Y_n, A_n) \otimes (U_n, V_n, B_n) \qquad (6)$$
$$n \approx_N n \wedge (X_n, Y_n, A_n) \multimap (U_n, V_n, B_n) \vdash (X_n, Y_n, A_n) \to (U_n, V_n, B_n). \qquad (7)$$

One easily sees that (6) implies (7), so it is enough to show that (6) holds. Unpacking the definitions, (6) holds iff

$$\big(\{n{+}1, 0\} \otimes X_n \otimes U_n, \{0\} \oplus Y_n \oplus V_n, \{(n{+}1, 0)\} \wedge A_n \wedge B_n\big) \vdash \big(X_n \otimes U_n, Y_n \otimes V_n, A_n \otimes B_n\big).$$

That this holds is witnessed by

$$f \in \{n+1, 0\} \otimes X_n \otimes U_n \Rightarrow X_n \otimes U_n,$$

and

$$F \in \{n+1, 0\} \otimes X_n \otimes U_n \otimes Y_n \otimes V_n \Rightarrow \{0\} \oplus Y_n \oplus V_n,$$

where

$$f\langle n,x,u\rangle = \langle x,u\rangle$$
$$F\langle n,x,u,y,v\rangle = \begin{cases}(2,v) & \text{if } \pi(g(n+1))=0,\\ (1,y) & \text{if } \pi(g(n+1))=1,\end{cases}$$

where g is witnessing the decidability of A_n (see the discussion in the beginning of this section). □

Remark 2.19. Note that the above proposition shows that for decidable predicates over N, our interpretation of conjunction and implication coincides with the one directly inspired by Gödel's Dialectica interpretation.

2.3 First Order Logic in $\mathbf{Dia}_m$

For $\mathbf{Dia}_m$ there does not seem to be the same easy results about first order logic that we have for the non-modified **Dia**. As already mentioned, there are two closed structures in $\mathbf{Dia}_m$: $(\otimes, \multimap)$ correspond to Gödel's interpretation, and $(\wedge, \to)$ gives a new variant. This new variant bas been described in [Bie07b] and is called the Copenhagen interpretation. It is a variant of Dialectica that does not have any requirements about decidable predicates. In this section we show that first order logic of $N_{\mathbf{Dia}_m}$ corresponds to the Copenhagen interpretation in the standard model **HRO**.

We are going to compare the interpretation of formulas ϕ of first order logic over the natural numbers in $\mathbf{Dia}_m$ with the Copenhagen translated (see [Bie07b]) formulas ϕ^C in the standard model **HRO**. The formulas ϕ^C have the form $\exists u \forall x \phi_C(n,u,x)$, where $n : N^k$ and ϕ_C is an atomic formula of HA. The interpretation of ϕ^C in **HRO** is denoted $\exists u : \mathsf{tp}_1(\phi) \forall x : \mathsf{tp}_2(\phi) . R_C(\phi)(n)(u,x)$, where $R_C(\phi)(n)(u,x) \subseteq \mathsf{tp}_1(\phi) \times \mathsf{tp}_2(\phi)$ is defined inductively as follows:

$\phi \equiv t = s$**:**

$$\mathsf{tp}_1(\phi) = \mathsf{tp}_2(\phi) = \{0\}, \qquad R_C(\phi)(n)(0,0) \text{ iff } t(n) = s(n).$$

$\phi \equiv \alpha \wedge \beta$**:**

$$\mathsf{tp}_1(\phi) = \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_1(\beta), \qquad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \oplus \mathsf{tp}_2(\beta),$$

$$\begin{array}{lll} R_C(\phi)(n)((u,v),\kappa(x)) & \text{iff} & R_C(\alpha)(n)(u,x) \text{ and} \\ R_C(\phi)(n)((u,v),\kappa'(y)) & \text{iff} & R_C(\beta)(n)(v,y). \end{array}$$

$\phi \equiv \alpha \to \beta$**:**

$$\begin{array}{lll} \mathsf{tp}_1(\phi) & = & \{m,h,H : N^k \otimes (\mathsf{tp}_1(\alpha) \Rightarrow \mathsf{tp}_2(\beta)) \otimes \\ & & (\mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) \Rightarrow \mathsf{tp}_2(\alpha) + 1) \mid \\ & & \forall a,b \in \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) . \mathsf{case}\, H(a,b) \in \mathsf{tp}_2(\alpha). \\ & & R_C(\alpha)(m)(a,H(a,b)) \supset R_C(\beta)(m)(ha,b)\} \\ \mathsf{tp}_2(\phi) & = & \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) \\ R_C(\phi)(n)(m,h,H,a,b) & \equiv & n = m \text{ and} \\ & & \left(\begin{array}{ll} \mathsf{case}\, H(a,b) \in \mathsf{tp}_2(\alpha). & \top, \\ \mathsf{case}\, H(a,b) \in 1. & R_C(\beta)(n)(ha,b) \end{array} \right) \end{array}$$

21

$\phi \equiv \exists z.\alpha(z)$**:**

$$\mathsf{tp}_1(\phi) = N \otimes \mathsf{tp}_1(\alpha), \qquad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha),$$
$$R_C(\phi)(n)(k, a, b) \equiv R_C(\alpha)(n, k)(a, b).$$

$\phi \equiv \forall z.\alpha(z)$**:**

$$\mathsf{tp}_1(\phi) = N \Rightarrow \mathsf{tp}_1(\alpha), \qquad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \otimes N,$$
$$R_C(\phi)(n)(h, a, k) = R_C(\alpha)(n, k)(h(k), a).$$

Lemma 2.20. *Any map $G : N^k_{\mathbf{Dia}_m} \to N_{\mathbf{Dia}_m}$ in $\mathbf{Dia}_m$ has the form $H^*(F)$ for $F : N^k_{\mathbf{Eff}} \to N_{\mathbf{Eff}}$ in* **Eff**, *and G is represented by $H^*([n = n] \wedge [s(n) = m])$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function.*

Proof. By Corollary 2.15, H^* is fully faithful on $N_{\mathbf{Eff}}$, and from [vO08] we know that any map $F : N^k_{\mathbf{Eff}} \to N_{\mathbf{Eff}}$ in **Eff** is represented by $[n = n] \wedge [s(n) = m])$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function. Thus G is represented by $H^*([n = n] \wedge [s(n) = m])$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function. □

Recall that a formula $\phi(x_1, \dots, x_k)$ with $x^k : N^k$ is interpreted as a subobject of $N^k_{\mathbf{Dia}_m}$, i.e., a strict predicate over $(\mathbb{N}^k, \approx)$. We denote this interpretation by $\|\phi\|(\bar{n}) = (\|\phi\|^+(\bar{n}), \|\phi\|^-(\bar{n}), \|\phi\|(\bar{n}))$.

Proposition 2.21. *For every formula $\phi(x_1, \dots, x_k)$ of HA and for every k-tuple $\bar{n} = n_1, \dots, n_k$, there are primitive recursive functions*

$$\begin{array}{ll} t_\phi(\bar{n}) : \mathsf{tp}_1(\phi) \to \|\phi\|^+(\bar{n}), & T_\phi(\bar{n}) : \mathsf{tp}_1(\phi) \times \|\phi\|^-(\bar{n}) \to \mathsf{tp}_2(\phi) \\ s_\phi(\bar{n}) : \|\phi\|^+(\bar{n}) \to \mathsf{tp}_1(\phi), & S_\phi(\bar{n}) : \|\phi\|^+(\bar{n}) \times \mathsf{tp}_2(\phi) \to \|\phi\|^-(\bar{n}) \end{array}$$

such that the following holds:

1. *for all $a \in \mathsf{tp}_1(\phi)$, $x \in \|\phi\|^-(\bar{n})$.*

$$R_C(\phi)(\bar{n})(a, T_\phi(\bar{n})(a, x))$$

   implies

$$\|\phi\|(\bar{n})(t_\phi(\bar{n})(a), x).$$

2. *For all $u \in \|\phi^+(\bar{n})\|$, $b \in \mathsf{tp}_2(\phi)$.*

$$\|\phi\|(\bar{n})(u, S_\phi(\bar{n})(u, b))$$

   implies

$$R_C(\phi)(\bar{n})(s_\phi(\bar{n})(u), b).$$

Proof. By induction on the structure of ϕ.

$\phi \equiv \mathbf{t} = \mathbf{s}$:
Let $s, t : (\mathbb{N}^k, \simeq) \to (\mathbb{N}, \simeq)$ be terms. By Lemma 2.20 s is represented by the functional relation

$$S(n, m) = H^*([\bar{n} = \bar{n}] \wedge [s(\bar{n}) = m])$$

22

where $s : \mathbb{N}^k \to \mathbb{N}$ is a function, and similar for t. By definition,

$$\begin{aligned}\|s = t\|(\bar{n}) &= \exists m : N.H^*(S(\bar{n}, m) \wedge T(\bar{n}, m)) \\ &= \begin{cases} H^*(\langle \bar{n}, s(\bar{n}), t(\bar{n})\rangle) & \text{if} \qquad s(\bar{n}) = t(\bar{n}) \\ H^*(\emptyset) = \bot & \text{otherwise.}\end{cases}\end{aligned}$$

We can define

$$t_\phi(\bar{n})(0) = \begin{cases} \langle \bar{n}, s(\bar{n}), t(\bar{n})\rangle & \text{if} \qquad s(\bar{n}) = t(\bar{n}), \\ 0 & \text{otherwise,}\end{cases}$$

$$T_\phi(\bar{n})(0, 0) = 0,$$

$$s_\phi(\bar{n})(u) = 0,$$

$$S_\phi(\bar{n})(u, 0) = 0.$$

$\phi \equiv \alpha \wedge \beta$:
We define

$$\begin{aligned} t_\phi(\bar{n})(a, b) &= \langle t_\alpha(\bar{n})(a), t_\beta(\bar{n})(b)\rangle \\ T_\phi(\bar{n})(a, b, (i, z)) &= \begin{cases} \kappa \circ T_\alpha(\bar{n})(a, z) & \text{if} \quad i = 0 \\ \kappa' \circ T_\beta(\bar{n})(b, z) & \text{if} \quad i = 1 \end{cases} \\ s_\phi(\bar{n})(u, v) &= \langle s_\alpha(\bar{n})(u), s_\beta(\bar{n})(v)\rangle \\ S_\phi(\bar{n})(u, v, (i, c)) &= \begin{cases} \kappa \circ S_\alpha(\bar{n})(u, c) & \text{if} \quad i = 0 \\ \kappa' \circ S_\beta(\bar{n})(v, c) & \text{if} \quad i = 1 \end{cases}\end{aligned}$$

We now show that these functions satisfy the requirements of the proposition.

1. Let $a \in \mathtt{tp}_1(\phi)$ and $x \in \|\phi\|^-(\bar{n})$, suppose $R_C(\phi)(\bar{n})(a, b, T_\phi(\bar{n})(a, b, (i, z)))$, that is,
$$\equiv \begin{cases} R_C(\alpha)(\bar{n})(a, T_\alpha(\bar{n})(a, z)) & \text{if} \quad i = 0 \\ R_C(\beta)(\bar{n})(b, T_\beta(\bar{n})(b, z)) & \text{if} \quad i = 1 \end{cases}$$
which by induction entails
$$\begin{cases} \|\alpha\|(\bar{n})(t_\alpha(\bar{n})(a), z) & \text{if} \quad i = 0 \\ \|\beta\|(\bar{n})(t_\beta(\bar{n})(b), z) & \text{if} \quad i = 1 \end{cases}$$
$$\equiv \|\phi\|(\bar{n})(t_\phi(\bar{n})(a, b), (i, z))$$
2. This case is identical.

$\phi \equiv \alpha \to \beta$:

$$\begin{array}{lcl} t_\phi(\bar{n})(m,h,H) & = & \langle g, G\rangle \quad \text{where} \\ g & = & \lambda u : \|\alpha\|^+(\bar{n}).\, t_\beta \cdot h \cdot s_\alpha(\bar{n})(u), \\ G & = & \lambda u, y : \|\alpha\|^+(\bar{n}) \otimes \|\beta\|^-(\bar{n}). \end{array}$$

$$\begin{cases} S_\alpha(\bar{n})(u, H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y))) & \text{if} \quad H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in \mathsf{tp}_2(\alpha) \\ * \in 1 & \text{if} \quad H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in 1 \text{ or } n \neq m \end{cases}$$

$$T_\phi(\bar{n})(m,h,H,u,y) \;=\; \langle s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(u), y)\rangle$$

$$\begin{array}{lcl} s_\phi(\bar{n})(f,F) & = & \langle n, k, K\rangle \quad \text{where} \\ k & = & \lambda a : \mathsf{tp}_1(\alpha). s_\beta \cdot f \cdot t_\alpha(\bar{n})(a), \\ K & = & \lambda a, b : \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta). \end{array}$$

$$\begin{cases} T_\alpha(\bar{n})(a, F(t_\alpha(\bar{n})(a), S_\beta(\bar{n})(f \cdot t_\alpha(\bar{n})(a), b))) & \text{if} \quad F(t_\alpha(\bar{n})(a), S_\beta(\bar{n})(f \cdot t_\alpha(\bar{n})(a), b)) \in \|\alpha\|^-(\bar{n}) \\ * \in 1 & \text{if} \quad F(t_\alpha(\bar{n})(a), S_\beta(\bar{n})(f \cdot t_\alpha(\bar{n})(a), b)) \in 1 \end{cases}$$

$$S_\phi(\bar{n})(f,F,a,b) \;=\; \langle t_\alpha(\bar{n})(a), S_\beta(\bar{n})(f \cdot t_\alpha(\bar{n})(a), b)\rangle$$

Next we type check t_ϕ (the type checking for s_ϕ is similar, so we leave that out). Let $(u,y) \in \|\alpha\|^+(\bar{n}) \otimes \|\beta\|^-(\bar{n})$, and suppose $G(u,y) \in \|\alpha\|^-(\bar{n})$ by the definition of G, this implies that $n = m$ and $H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in \mathsf{tp}_2(\alpha)$, we need to show that

$$\|\alpha\|(\bar{n})(u, G(u,y)) \to \|\beta\|(\bar{n})(gu, y).$$

To that end suppose

$$\|\alpha\|(\bar{n})(u, S_\alpha(\bar{n})(u, H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y))))$$

by induction we get

$$R_C(\alpha)(\bar{n})(s_\alpha(\bar{n}), H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)))$$

and since $H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in \mathsf{tp}_2(\alpha)$ and by definition of $\mathsf{tp}_1(\phi)$, we get

$$R_C(\beta)(\bar{n})(h \cdot s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y))$$

which by induction implies

$$\|\beta\|(\bar{n})(t_\beta \cdot h \cdot s_\alpha(\bar{n})(u), y) = \|\beta\|(\bar{n})(gu, y)$$

and finally we show the two conditions of the Proposition

1. Let $(m,h,H) \in \mathsf{tp}_1(\phi)$, $(u,y) \in \|\phi\|^-(\bar{n})$.

   $$\begin{array}{l} R_C(\phi)(\bar{n})(\bar{m}, h, H, T_\phi(\bar{n}(\bar{m}, h, H, u, y))) \equiv \\ R_C(\phi)(\bar{n})(h, H, s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \equiv \end{array}$$

   $$\begin{cases} \top & \text{if} \quad H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in \mathsf{tp}_2(\alpha) \\ R_C(\beta)(\bar{n})(h \cdot s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) & \text{if} \quad H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in 1 \end{cases}$$

   We must show that

   $$\|\phi\|(\bar{n})(t_\phi(\bar{n})(h,H), u, y)$$

   but this follows from the observation that $G(u,y) \in 1$ iff $\bar{n} \neq \bar{m}$ or $H(s_\alpha(\bar{n})(u), T_\beta(\bar{n})(h \cdot s_\alpha(\bar{n})(u), y)) \in 1$.

2. The other direction is similar.

$\phi \equiv \forall \mathbf{y} : \mathbb{N}.\alpha(\bar{\mathbf{n}}, \mathbf{y})$:

First recall that the interpretation $\|\forall y : \mathbb{N}.\alpha(\bar{n}, y)\|$ in the topos is defined in the tripos logic as follows:

$$\|\forall y : \mathbb{N}.\alpha(\bar{n}, y)\| = [\bar{n} = \bar{n}] \wedge \forall y : \mathbb{N}.([y = y] \to \|\alpha\|(\bar{n}, y)).$$

We have

$(\forall y : \mathbb{N}.([y = y] \to \|\alpha\|(\bar{n}, y)))^+(\bar{n}) \quad =$

$\bigcap_{y\in\mathbb{N}}\{(f, F) : (\{0, y+1\} \to \|\alpha\|^+(\bar{n}, y)) \otimes (\{0, y+1\} \otimes \|\alpha\|^-(\bar{n}, y) \to 2) \mid$
$\quad \forall u, x \in \{0, y+1\} \otimes \|\alpha\|^-(\bar{n}, y).\, F(u, x) = 0 \supset u = y + 1 \supset \|\alpha\|(\bar{n}, y)(fu, y)\},$

$(\forall y : \mathbb{N}.([y = y] \to \|\alpha\|(\bar{n}, y)))^-(\bar{n}) = \bigcup_{y\in\mathbb{N}}\{0, y+1\} \otimes \|\alpha\|^-(\bar{n}, y),$

$(\forall y : \mathbb{N}.([y = y] \to \|\alpha\|(\bar{n}, y)))(\bar{n})(f, F, u, x) \equiv \mathsf{case}\, F(u, x) = 1.\, \|\alpha\|(\bar{n}, u-1)(fu, x),$
$\mathsf{case}\, F(u, x) = 0.\, \top.$

We define functions as follows:

$$\begin{array}{lcl}
t_\phi(\bar{n})(h) & = & \langle \bar{n}+1, g, G\rangle \qquad \text{where} \\[2ex]
g & = & \lambda y : \mathbb{N}, u : \{0, y+1\}. \begin{cases} 0 & \text{if } u = 0 \\ t_\alpha(\bar{n}, u-1)(hu) & \text{otherwise} \end{cases} \\
G & = & \lambda y : \mathbb{N}, u : \{0, y+1\},\ x : \|\alpha\|^-(\bar{n}, y). \begin{cases} 0 & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases} \\[3ex]
T_\phi(\bar{n})(h, u, x) & = & \begin{cases} \langle 0, 0\rangle & \text{if } u = 0 \\ \langle T_\alpha(\bar{n}, u-1)(hu, x), u\rangle & \text{if } u \neq 0 \end{cases} \\[3ex]
s_\phi(\bar{n})(f, F) & = & \lambda k.\, s_\alpha(\bar{n}, k)(f(k+1)) \\[3ex]
S_\phi(\bar{n})(f, F, b, k) & = & \langle k+1, S_\alpha(\bar{n}, k)(f(k+1), b)\rangle
\end{array}$$

We show that the two conditions are satisfied:

1. Suppose $h \in \mathsf{tp}_1(\phi)$, $(u, x) \in \|\phi\|^-(\bar{n})$. And suppose that

$$R_C(\phi)(\bar{n})(h, T_\phi(\bar{n})(h, u, x))$$

holds, i.e., if $u \neq 0$ this is

$$\begin{array}{cl}
 & R_C(\phi)(\bar{n})(h, T_\alpha(\bar{n}, u-1)(hu, x), u)) \\
\equiv & R_C(\alpha)(\bar{n}, u-1)(h(u), T_\alpha(\bar{n}, u-1)(hu, x)) \\
\Rightarrow & \|\alpha\|(\bar{n}, u-1)(t_\alpha(\bar{n}, u-1)(hu), x) \\
\equiv & \|\phi\|(\bar{n})(t_\phi(\bar{n})(h), u, x)
\end{array}$$

In case $u = 0$ we get

$$\begin{array}{cl}
 & R_C(\phi)(\bar{n})(h, T_\phi(\bar{n})(h, u, x)) \\
\equiv & R_C(\phi)(\bar{n})(h, (0, 0)) \\
\equiv & R_C(\alpha)(\bar{n}, 0)(h(0), 0) \\
\Rightarrow & \|\phi\|(\bar{n})(t_\phi(\bar{n})(h), u, x)
\end{array}$$

where the last implication holds because when $u = 0$ we have $G(u, x) = 0$, hence $\|\phi\|(\bar{n})(t_\phi(\bar{n})(h), u, x) \equiv \|\phi\|(\bar{n})(\bar{n}+1, g, G, u, x) \equiv \top$.

2. If $(f, F) \in \|\phi\|^+(\bar{n})$ and $(b, k) \in tp_2(\phi)$, we have

$$\begin{array}{ll} & \|\phi\|(\bar{n})(f, F, S_\phi(\bar{n})(f, F, b, k)) \\ \equiv & \|\phi\|(\bar{n})(f, F, k+1, S_\alpha(\bar{n}, k)(f(k+1), b)) \\ \equiv & \|\alpha\|(\bar{n}, k)(f(k+1), S_\alpha(\bar{n}, k)(f(k+1), b)) \\ \Rightarrow & R_C(\alpha)(\bar{n}, k)(s_\alpha(\bar{n}, k)(f(k+1)), b) \\ \equiv & R_C(\phi)(\bar{n})(\lambda k.s_\alpha(\bar{n}, k)(f(k+1)), b, k) \\ \equiv & R_C(\phi)(\bar{n})(s_\phi(\bar{n})(f, F), b, k) \end{array}$$

$\phi \equiv \exists \mathbf{y}.\alpha$:
For this case we use μ and δ from Proposition 2.11.

$$\begin{array}{lcl} t_\phi(\bar{n})(k, a) & = & t_\alpha(\bar{n}, k)(a) \\ T_\phi(\bar{n})(k, a, H) & = & T_\alpha(\bar{n}, k)(a, H(t_\alpha(\bar{n}, k))(a)) \\ s_\phi(\bar{n})(x) & = & (\mu(x) - 1, s_\alpha(\bar{n}, \mu(x) - 1)(x)) \end{array}$$

$$S_\phi(\bar{n})(x, b) \quad = \quad \begin{cases} \lambda v : \|\alpha\|^+(\bar{n}, \mu(v) - 1).\, S_\alpha(\bar{n}, \mu(v) - 1)(v, b) & \text{if } \mu(v) = m + 1 \\ \delta & \text{if } \mu(v) = 0 \end{cases}$$

Now we can show

1. If $(k, a) \in \mathsf{tp}_1(\phi)$, $H \in \|\phi\|^-(\bar{n})$, we have

$$\begin{array}{ll} & R_C(\phi)(\bar{n})(k, a, T_\phi(\bar{n})(k, a, H)) \\ \equiv & R_C(\alpha)(\bar{n}, k)(a, T_\alpha(\bar{n}, k)(a, H(t_\alpha(\bar{n}, k)(a)))) \\ \Rightarrow & \|\alpha\|(\bar{n}, k)(t_\alpha(\bar{n}, k)(a), H(t_\alpha(\bar{n}, k))(a)) \\ \equiv & \|\phi\|(\bar{n})(t_\phi(\bar{n})(k, a)) \end{array}$$

2. If $x \in \|\phi\|^+(\bar{n}), b \in \mathsf{tp}_2(\alpha)$ then if $\mu(x) = y + 1$ we have

$$\begin{array}{ll} & \|\phi\|(\bar{n})(x, S_\phi(\bar{n})(x, b)) \\ \equiv & \|alpha\|(\bar{n}, y)(x, S_\alpha(\bar{n}, y)(x, b)) \\ \Rightarrow & R_C(\alpha)(\bar{n}, y)(s_\alpha(\bar{n}, y)(x), b) \\ \equiv & R_C(\phi)(\bar{n})(y, s_\alpha(\bar{n}, y)(x), b) \\ \equiv & R_C(\phi)(\bar{n})(s_\phi(\bar{n})(x), b) \end{array}$$

   And if $\mu(x) = 0$ we get

$$\begin{array}{ll} & \|\phi\|(\bar{n})(x, S_\phi(\bar{n})(x, b)) \\ \equiv & \|\alpha\|(\bar{n}, y)(x, \delta(x)) \qquad \text{for some } y \in \mathbb{N} \end{array}$$

   but since by Proposition 2.11 we have

$$\forall y : \mathbb{N}.\neg\|\alpha\|(\bar{n}, y)(x, \delta(x))$$

   we conclude that in case $\mu(x) = 0$,

$$\|\phi\|(\bar{n})(x, S_\phi(\bar{n})(x, b)) \equiv \bot$$

   so the implication we needed to show holds trivially.

□

Theorem 2.22. *For all formulas ϕ of HA,*

$$\mathbf{HRO} \models \|\phi^C\| \qquad \textit{iff} \qquad \mathbf{Dia}_m \models \|\phi\|.$$

Proof. $\mathbf{HRO} \models \|\phi^C\|$ means that there is a natural number $a \in \mathsf{tp}_1(\phi)$ such that $\forall b : \mathsf{tp}_2.R_C(\phi)(\bar{n})(a,b)$. And $\mathbf{Dia}_m \models \|\phi\|$ means that there is a $u \in \|\phi\|^+(\bar{n})$ such that $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(u,x)$. Suppose $\forall b : \mathsf{tp}_2.R_C(\phi)(\bar{n})(a,b)$ then by Proposition 2.21 we have $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(t_\phi(\bar{n})(a),x)$. On the other hand, suppose $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(u,x)$ then by Proposition 2.21 we have $\forall b : \mathsf{tp}_2.R_C(\phi)(\bar{n})(s_\phi(\bar{n})(u),b)$. □

2.4 Relation of $\mathbf{Dia}_m$ to Modified Realizability and to Set

Proposition 2.23. *There is a fibred adjunction p from d_m to m. The functor p^* is full and faithful. Moreover, it satisfies $p^*(A \wedge B) \dashv\vdash p^*(A) \otimes p^*(B)$.*

Proof. Over 1 the fibred adjunction is defined by

$$\begin{aligned} p^*(U^a, U^p) &= (U^p, \{0\}, U^a \times \{0\}), \\ p_*(X,Y,A) &= (\{\, x \in X \mid \forall y \in Y.\ A(x,y) \,\}, X). \end{aligned}$$

The functor p^* is full and (necessarily) faithful. Verification of the equivalence $p^*(A \wedge B) \dashv\vdash p^*(A) \otimes p^*(B)$ is straight forward. □

Remarks 2.24.

(i) The functor p_* does not preserve existential quantification, so it does not have any right adjoints.

(ii) The functor $p^* : \mathsf{m} \to \mathsf{d}_m$ does *not* preserve conjunction so p is not a geometric morphism of triposes hence, presumably, p^* does not lift to a functor between the induced toposes. The proof that p does not preserve conjunction is precisely as the proof in 1.9.

Recall from [vO97] that there are two embeddings, denoted Δ and ∇, of **Set** into **Mod**, with Δ the constant objects functor and ∇ the inclusion of $\neg\neg$-sheaves with left adjoint the global sections functor. Composing with G^*, we also get two embeddings at the tripos level (both full and faithful since G^* is also so) $\Delta_{\mathsf{d}_m} = G^* \circ \Delta$ and $\nabla_{\mathsf{d}_m} = G^* \circ \nabla$ of **Set** into d_m. When there is no risk of confusion, we often leave out the subscripts of Δ_{d_m} and ∇_{d_m}. Δ_{d_m} is indeed the constant-objects functor associated with the modified Dialectica tripos.

Explicitly, the constant-objects functor $\Delta\colon \mathbf{Set} \to \mathbf{Dia}_m$ is, as usual, given by $\Delta X = (X, \exists_{\delta(X)}(\top)) = (X, \approx_\Delta)$, where $\delta(X)$ is the diagonal $\langle id_X, id_X\rangle\colon X \to X \times X$, and where

$$x \approx_\Delta x' = \begin{cases} (\{0,1\},\{0\},\{(1,0)\}) & \text{if } x = x', \\ (\{0\},\{0\},\emptyset) & \text{if } x \neq x'. \end{cases}$$

At the level of the underlying triposes, Δ is a indexed functor from the subset-tripos to d_m, given over I by

$$\Delta(I' \subseteq I)(i) = \begin{cases} (\{0,1\},\{0\},\{(1,0)\}) & \text{if } i \in I' \\ \bot & \text{otherwise.} \end{cases}$$

Since Δ does not preserve $\wedge$, it does not have a left adjoint. Further, since Δ does not preserve $\vee$, it does not have a right adjoint (if it had a right adjoint, it would have been strange indeed, since then $\mathbf{Dia}_m$ would have been localic over $\mathbf{Set}$).

At the level of the underlying triposes, ∇ is a indexed functor from the subset-tripos, $\mathbf{Set}(-,2)$ to d_m, given over I by

$$\nabla(I' \subseteq I)(i) = \begin{cases} (\mathbb{N}, \{0\}, \mathbb{N} \times \{0\}) & \text{if } i \in I' \\ (\mathbb{N}, \{0\}, \emptyset) & \text{otherwise.} \end{cases}$$

or, equivalently:

$$\nabla(I' \subseteq I)(i) = \begin{cases} (\{0,1\}, \{0\}, \{(1,0)\}) & \text{if } i \in I' \\ (\{0,1\}, \{0\}, \emptyset) & \text{otherwise.} \end{cases}$$

Notice that though Δ and ∇ are pointwise equivalent, they are not so uniformly. ∇ has an indexed right adjoint, $F : \mathsf{d}_m \to \mathbf{Set}(-,2)$ given by

$$F(X_i, Y_i, A_i)(i) = \begin{cases} 1 & \text{if } \exists x \in \bigcap_{i \in I} X_i . \forall y \in Y_i . A_i(x,y) \\ 0 & \text{otherwise.} \end{cases}$$

The adjunction is not a geometric morphism of triposes since ∇ does not preserve $\wedge$.

So far, we have the following diagram of fibred adjunctions, where, however, only some give rise to geometric morphisms:

$$\begin{tikzcd}
 & & & \mathsf{d}_m \arrow[r, shift left, "F"] \arrow[lld, shift left, "p_*"] \arrow[d, shift left, "H_*"] & \mathbf{Set} \arrow[l, shift left, "\nabla"] \\
 & \mathsf{m} \arrow[rru, shift left, "p^*"] \arrow[rr, shift left, "v_*"] \arrow[ld, shift left, "\Gamma"] & & \mathsf{e} \arrow[u, shift left, "H^*"] \arrow[ll, shift left, "v^*"] \arrow[ld, shift left, "\Gamma"] \arrow[rd, shift left, "j^*"] & \\
\mathbf{Set} \arrow[ru, shift left, "\Delta"] \arrow[rr, "\cong"] & & \mathbf{Set} \arrow[ru, shift left, "\Delta"] & & \mathsf{d} \arrow[lu, shift left, "j_*"]
\end{tikzcd}$$

where H, v, are connected geometric morphisms, so they lift to surjective geometric morphisms on the induced toposes, and j is an open geometric inclusion, so it lifts to an open geometric inclusion. The left adjoints of the adjunctions $! \dashv id$, $\nabla \dashv F$ and $p^* \dashv p_*$ are all full and faithful. At topos level we have the following geometric morphisms

$$\begin{tikzcd}
 & & & \mathbf{Dia}_m \arrow[d, "H"] & \\
 & \mathbf{Mod} \arrow[rr, "v"] & & \mathbf{Eff} \arrow[rd, hook, "j"] & \\
\mathbf{Mod}_{\neg\neg} \arrow[ru, hook] \arrow[rr, "\cong"] & & \mathbf{Eff}_{\neg\neg} \arrow[ru, hook] & & \mathbf{Dia}
\end{tikzcd}$$

28

3 The Diller-Nahm Tripos

According to Gödel's original Dialectica Interpretation from 1958 a proposition is a pair of types X and Y together with a *decidable* relation $A \subseteq X \times Y$ between them. For our purposes we assume that types are subsets of $\mathbb{N}$ and (constructive) functionals between them are total recursive functions between sets of natural numbers. Entailment between propositions (X, Y, A) and (U, V, B) is given by a pair of (constructive) functionals $f\colon X \to U$ and $F\colon X \times V \to Y$ such that

$$\forall x \in X. \forall v \in V.\ A(x, F(x, v)) \Rightarrow B(f(x), v)\,.$$

However, if relations are not required to be decidable[1] one usually considers the *Diller-Nahm variant* of the Dialectica interpretation where entailment is defined in a somewhat different way: $(X, Y, A) \vdash (U, V, B)$ if there are (constructive) functionals $f\colon X \to U$ and $g\colon X \times V \to \mathsf{P_f}(Y)$ such that

$$\forall x \in X. \forall v \in V.\ [\forall y \in g(x, v). A(x, y)] \Rightarrow B(f(x), v)\,.$$

Here $m \in n$ stands for $m \in e_n$ (where e is some standard Gödel numbering of finite sets of natural numbers) and $\mathsf{P_f}(Y)$ is a shorthand for $\{n \in \mathbb{N} \mid e_n \subseteq Y\}$.

Motivated by these considerations we are now going to define the *Diller-Nahm Dialectica tripos* dn over **Set**. Let

$$\Sigma_{\mathsf{dn}} = \{(X, Y, A) \in \mathcal{P}(\mathbb{N})^2 \times \mathcal{P}(\mathbb{N} \times \mathbb{N}) \mid A \subseteq X \times Y\}$$

be the "set of truth values of dn". If $p = (X, Y, A) \in \Sigma_{\mathsf{dn}}$ we write p^+, p^-, $p(x, y)$ for X, Y, $A(x, y)$, respectively. For $I \in$ **Set** the fibre dn^I is defined as the preorder $({\Sigma_{\mathsf{dn}}}^I, \vdash_I)$, where ${\Sigma_{\mathsf{dn}}}^I$ is the set of all functions from I to Σ_{dn} and $\varphi \vdash_I \psi$ if there exist

$$e^+ \in \bigcap_{i \in I} [\varphi_i^+ \Rightarrow \psi_i^+] \quad \text{and} \quad e^- \in \bigcap_{i \in I} [\varphi_i^+ \otimes \psi_i^- \Rightarrow \mathsf{P_f}(\varphi_i^-)]$$

such that

$$\forall i \in I.\, \forall a \in \varphi_i^+, b \in \psi_i^-.\, [\forall c \in e^- \langle a, b\rangle.\, \varphi_i(a, c)] \supset \psi_i(e^+ a, b)\ \ .$$

Terminal Object
The terminal object is given by $\top = (\{0\}, \emptyset, \emptyset)$. Notice that $\top \cong (\{0\}, \{0\}, \{\langle 0, 0\rangle\})$.

Products
The conjunction $p \wedge q$ is given by

(1) $(p \wedge q)^+ = p^+ \otimes q^+$

(2) $(p \wedge q)^- = p^- \oplus q^-$

(3) $(p \wedge q)(\langle n, m\rangle, \langle i, k\rangle)$ iff $(i = 0 \wedge p(n, k)) \vee (i = 1 \wedge q(m, k))$.

[1]Which is impossible as for quantification we have to take into account *arbitrary* unions and intersections of relations.

Exponentials
The implication $p \to q$ is given by

(1) $(p \to q)^+ = (p^+ \Rightarrow q^+) \otimes (p^+ \otimes q^- \Rightarrow \mathsf{P_f}(p^-))$
(2) $(p \to q)^- = p^+ \otimes q^-$
(3) $(p \to q)(\langle e^+, e^- \rangle, \langle a, b \rangle)$ iff $\big[\forall c \in e^- \langle a, b \rangle .\, p(a, c)\big] \supset q(e^+ a, b)$.

Next we consider quantification in dn. In the following we write $[i = j]$ for $\{0 \mid i = j\}$.

Universal Quantification
For $u \colon I \to J$ and $\varphi \in \mathsf{dn}^I$ we construct $\forall_u(\varphi) \in \mathsf{dn}^J$ as follows

(1) $\forall_u(\varphi)^+_j = \bigcap_{i \in I} [u(i) = j] \Rightarrow \varphi^+_i$
(2) $\forall_u(\varphi)^-_j = \bigcup_{i \in u^{-1}(j)} \varphi^-_i$
(3) $\forall_u(\varphi)_j(a, b)$ iff $\forall i \in u^{-1}(j).\, \big(b \in \varphi^-_i \supset \varphi_i(a \cdot 0, b)\big)$.

Notice that in case $u \colon I \to J$ is epic one may simplify the construction of $\forall_u(\varphi)$ by putting $\forall_u(\varphi)^+_j = \bigcap_{i \in u^{-1}(j)} \varphi^+_i$ and

$$\forall_u(\varphi)_j(a, b) \text{ iff } \forall i \in u^{-1}(j).\, \big(b \in \varphi^-_i \supset \varphi_i(a, b)\big) \ .$$

Existential Quantification
For $u : I \to J$ and $\varphi \in \mathsf{dn}^I$ existential quantification is given by

(1) $\exists_u(\varphi)^+_j = \bigcup_{i \in u^{-1}(j)} \varphi^+_i$
(2) $\exists_u(\varphi)^-_j = \bigcap_{i \in u^{-1}(j)} [\varphi^+_i \Rightarrow \mathsf{P_f}(\varphi^-_i)]$
(3) $\exists_u(\varphi)_j(a, b) \Leftrightarrow \exists i \in u^{-1}(j).\, \big(a \in \varphi^+_i \wedge \forall c \in ba.\, \varphi_i(a, c)\big)$.

Notice that in case $u \colon I \to J$ is epic one may simplify the construction of $\exists_u(\varphi)$ by putting $\exists_u(\varphi)^-_j = \bigcap_{i \in u^{-1}(j)} \mathsf{P_f}(\varphi^-_i)$ and

$$\exists_u(\varphi)_j(a, b) \text{ iff } \exists i \in u^{-1}(j).\, \big(a \in \varphi^+_i \wedge \forall c \in b.\, \varphi_i(a, c)\big) \ .$$

Generic Predicate
The generic predicate is given by the identity function on Σ_{dn} considered as an element of $\mathsf{dn}(\Sigma_{\mathsf{dn}})$.

Disjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their disjunction $A_0 \vee A_1$ is given by $(A_0 \vee A_1)^+ = A^+_0 \oplus A^+_1$, $(A_0 \vee A_1)^- = A^-_0 \oplus A^-_1$ and $(A_0 \vee A_1)(i, \langle k, n \rangle, \langle j, m \rangle) \equiv k = j \ \supset \ A_k(i, n, m)$.

Falsity
Falsity is given by $\bot = (\emptyset, \emptyset, \emptyset)$. Notice that $\bot \cong (\emptyset, \{0\}, \emptyset)$.

Negation
For a proposition p its negation $\neg p$ is given by

$$\neg p = \begin{cases} (\mathbb{N}, \emptyset, \emptyset) & \text{if } p^+ = \emptyset \\ (\emptyset, \emptyset, \emptyset) & \text{otherwise.} \end{cases}$$

Accordingly, double negation is given by

$$\neg\neg p = \begin{cases} (\emptyset, \emptyset, \emptyset) & \text{if } p^+ = \emptyset \\ (\mathbb{N}, \emptyset, \emptyset) & \text{otherwise.} \end{cases}$$

As $(\mathbb{N}, \emptyset, \emptyset) \cong \top$ we have that $\neg p$ is $\top$ if $p^+ = \emptyset$ and $\bot$ otherwise.

Lawvere Equality
For a set I the equality predicate $\mathsf{eq}_I \in \mathsf{dn}^{I \times I}$ is given by

$$\mathsf{eq}_I(i,j)^+ = \{0 \mid i = j\} \quad \text{and} \quad \mathsf{eq}_I(i,j)^- = \emptyset$$

for $i, j \in I$. One easily shows that $\mathsf{eq}_I \dashv\vdash \exists_{\delta(I)}(\top_I)$, i.e. that eq_I coincides with Lawvere's notion of equality.

As dn is a tripos one may consider the associated *Diller-Nahm topos* $\mathbf{DN} = \mathbf{Set}[\mathsf{dn}]$ obtained by the tripos-to-topos construction. Due to the particular nature of (double) negation the $\neg\neg$–sheaves of dn are equivalent to $\mathbf{Set}$. However, the sheafification functor for the $\neg\neg$-topology is not given by the global sections functor. The reason is that dn is not $\exists$-standard as can be seen from the following counterexample:

Proposition 3.1. dn *is not* $\exists$*-standard.*

Proof. Consider $p \in \mathsf{dn}^{\mathbb{N}}$ as given by $p_n^+ = (\{0\}, \{0, n+1\}, \{\langle 0, 0\rangle\})$ all whose items p_n are not valid whereas $\exists_{\mathbb{N}}(p) = (\{0\}, \{0\} \Rightarrow \mathsf{P_f}(\{0\}), \{0\} \times (\{0\} \Rightarrow \mathsf{P_f}(\{0\})))$ is valid. $\square$

3.1 Relation of DN to $\mathbf{Eff}_2$

Just as for d we have a connected geometric morphism from dn to e_2 which, moreover, is given by the same recipe as $p : \mathsf{d} \to \mathsf{e}_2$.

Proposition 3.2. *There is a connected geometric morphism* $q : \mathsf{dn} \to \mathsf{e}_2$ *given by*

$$q^*(A^a, A^p) = \big(A^p, \{0\}, A^a \times \{0\}\big)$$
$$q_*(X, Y, R) = \big(\{x \in X \mid \forall y \in Y.\, R(x,y)\}\,,\, X\big) \ .$$

As already mentioned, the subtoposes $\mathbf{Eff}$ and $\mathbf{Mod}$ of $\mathbf{Eff}_2$ are induced by the local operators $o_U(P) = U \to P$ and $c_U(P) = U \vee P$ where $U = (\emptyset, \{0\})$. Since pullbacks along geometric morphisms preserve open and closed subtoposes (Lemma C.1.2.10 of [Joh02]) the pullback of $e : \mathsf{e} \hookrightarrow \mathsf{e}_2$ along q is given by the local operator $o_{q^*(U)}$ and the pullback of $m : \mathsf{m} \hookrightarrow \mathsf{e}_2$ along q is given by the local operator $c_{q^*(U)}$, where, obviously, $q^*(U) = (\{0\}, \{0\}, \emptyset)$.

The local operators $o_{q^*(U)}$ and $c_{q^*(U)}$, give rise to the open subtripos dn_e and the closed subtripos dn_m of dn, respectively. We denote the corresponding open and closed subtoposes of **DN** by $\mathbf{DN}_e$ and $\mathbf{DN}_m$, respectively. For more details see [Bie08, Chapter 1]. We the pullback diagrams:

$$\begin{tikzcd}
\mathbf{DN}_e \arrow[r] \arrow[d, hook] & \mathbf{Eff} \arrow[d, hook] \\
\mathbf{DN} \arrow[r, "q"'] & \mathbf{Eff}_2
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathbf{DN}_m \arrow[r] \arrow[d, hook] & \mathbf{Mod} \arrow[d, hook] \\
\mathbf{DN} \arrow[r, "q"'] & \mathbf{Eff}_2
\end{tikzcd}$$

Notice that $q^*(U) \vdash_I A$ iff $\bigcap_{i \in I} A_i^+ \neq \emptyset$. Thus, up to isomorphism we may define dn_m as canonically presented via $\Sigma_{\mathsf{dn}_m} = \{(U, X, R) \in \Sigma_{\mathsf{dn}} \mid 0 \in U\}$. The logical operations for dn_m are defined as for dn, only for $\bot$, $\vee$ and $\exists$ we have to correct the constructions by finally applying the local operator $c_{q^*(U)}$.

3.2 Relation of DN to Number Realizability

Next we will show that dn contains as subtripos the number realizability tripos e from which the effective topos $\mathbf{Eff} = \mathbf{Set}[\mathsf{e}]$ arises via the tripos-to-topos construction. Recall that $\Sigma_{\mathsf{e}} = \mathcal{P}(\mathbb{N})$ and $\varphi \vdash_I \psi$ iff $\bigcap_{i \in I}[\varphi_i \Rightarrow \psi_i]$ is inhabited.

Proposition 3.3. *There is an injective geometric morphism $i : \mathsf{e} \to \mathsf{dn}$ and, accordingly,* **Eff** *is a subtopos of* **DN**. *This geometric inclusion arises from the* open *topology $u \to (-)$ on* **DN** *where $u = (\{0\}, \{0\}, \emptyset)$. Moreover, one obtains* **Set** *as the subtopos of $\neg\neg$-sheaves of* **DN**.

Proof. We define $i : \mathsf{e} \to \mathsf{dn}$ as the fibred adjunction

$$i^*(X, Y, R) = X \qquad \text{and} \qquad i_*(X) = (X, \emptyset, \emptyset)$$

and one easily checks that i^* preserves $\top$ and also conjunction in each fibre.

One easily checks that $i_* i^*(X, Y, R) = (X, \emptyset, \emptyset) \cong u \to (X, Y, R)$ from which it follows that the geometric inclusion i is induced by the topology $u \to (-)$ on **DN**.

It is well known that $\mathbf{Set} \simeq \mathbf{Eff}_{\neg\neg}$. Thus, since $\neg\neg(u \to (X, Y, R)) = \neg\neg(X, Y, R)$ it follows that $\mathbf{Set} \simeq \mathbf{DN}_{\neg\neg}$. □

In general direct image parts of geometric morphisms do not preserve natural numbers objects. But in the particular case $i\colon \mathbf{Eff} \to \mathbf{DN}$ one readily checks that $i_*(N) = (\mathbb{N}, E_N)$ with $E_N(n, m) = (\{k \in \mathbb{N} \mid n = k = m\}, \emptyset, \emptyset)$ happens to be a natural numbers object in **DN**.

Proposition 3.4. *The inverse image i^* preserves first-order logic and the finite type structure of the natural number object in* **DN**.

Proof. As the proof of proposition 1.13.

□

3.3 First Order Logic in DN

Section 1.3 applies here as well.

4 Modified Diller-Nahm Tripos

We now investigate the subtopos of **DN** which is the complement of **Eff**, i.e. the subtopos of **DN** induced by the topology $u \vee (-)$.

Notice that $p^*(U) \vdash_I A$ iff $\bigcap_{i\in I} A_i^+ \neq \emptyset$. Thus, up to isomorphism we may define the tripos dn_m as canonically presented via $\Sigma_{\mathsf{dn}_m} = \{(U,X,R) \in \Sigma_{\mathsf{dn}} \mid 0 \in U\}$. The subtopos $\mathbf{DN}_m$ of **DN** arises via the tripos-to-topos construction from the tripos dn_m. The logical operations for dn_m are defined as for dn, only for $\bot$, $\vee$ and $\exists$ we have to correct the constructions by finally applying the local operator $c_{p^*(U)}$.

The set of truth values of the *Modified Diller-Nahm tripos* dn_m over **Set** is thus

$$\Sigma_{\mathsf{dn}_m} = \{(X,Y,A) \in \mathcal{P}_0(\mathbb{N}) \times \mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}\times\mathbb{N}) \mid A \subseteq X \times Y\}$$

where $\mathcal{P}_0(\mathbb{N})$ is the set of all subsets of $\mathbb{N}$ containing 0 as an element.

For $I \in$ **Set** the fibre ${\mathsf{dn}_m}^I$ is defined as the preorder $(\Sigma^I_{\mathsf{dn}_m}, \vdash_I)$, where Σ^I_{DmT} the set of all functions from I to Σ_{dn_m} and $\varphi \vdash_I \psi$ iff there exist $e^+ \in \bigcap_{i\in I}[\varphi_i^+ \Rightarrow \psi_i^+]$ and $e^- \in \bigcap_{i\in I}[\varphi_i^+ \times \psi_i^- \Rightarrow \mathsf{P_f}(\varphi_i^-)]$ such that

$$\forall i \in I.\, \forall a \in \varphi_i^+, b \in \psi_i^-.\, [\forall c \in e^-\langle a,b\rangle.\, \varphi_i(a,c)] \supset \psi_i(e^+a,b) \ \ .$$

In many cases the verification of the tripos requirement is the same as for dn together with the observation that 0 shows up in the positive part of propositions. Sometimes, however, the 0 has to be added after "shifting by 1".

Existential Quantification
For $u\colon I \to J$ and $\varphi \in {\mathsf{dn}_m}^I$ existential quantification is given by

(1) $\exists_u(\varphi)_j^+ = \{0\} \cup succ\big(\bigcup_{i\in u^{-1}(j)} \varphi_i^+\big)$

(2) $\exists_u(\varphi)_j^- = \bigcap_{i\in u^{-1}(j)} [\varphi_i^+ \Rightarrow \mathsf{P_f}(\varphi_i^-)]$

(3) $\exists_u(\varphi)_j(a+1,b)$ iff $\exists i \in u^{-1}(j).\, \big(a \in \varphi_i^+ \wedge \forall c \in ba.\, \varphi_i(a,c)\big)$ and $\exists_u(\varphi)_j(0,b)$ never holds.

Notice that $\exists_u(\varphi)_j^+$ contains 0 by construction and $\exists_u(\varphi)_j^-$ contains 0 as all $\mathsf{P_f}(\varphi_i^-)$ and thus all $[\varphi_i^+ \Rightarrow \mathsf{P_f}(\varphi_i^-)]$ contain 0.

Notice that in case $u : I \to J$ is epic the construction of $\exists_u(\varphi)$ can be simplified by putting $\exists_u(\varphi)_j^+ = \bigcup_{i\in u^{-1}(j)} \varphi_i^+$ and

$$\exists_u(\varphi)_j(a,b) \text{ iff } \exists i \in u^{-1}(j).\, \big(a \in \varphi_i^+ \wedge \forall c \in b.\, \varphi_i(a,c)\big) \ \ .$$

Generic Predicate
The generic predicate is given by the identity function on Σ considered as an element of ${\mathsf{dn}_m}^\Sigma$.

The structure exhibited so far guarantees dn_m to be tripos.

Disjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their disjunction $A_0 \vee A_1$ is given by $(A_0 \vee A_1)^+ = A_0^+ \oplus A_1^+$, $(A_0 \vee A_1)^- = A_0^- \oplus A_1^-$ and $(A_0 \vee A_1)(i, \langle k, n\rangle, \langle j, m\rangle) \equiv k = j \supset A_k(i, n, m)$.

Falsity
Falsity is given by $\bot = (\{0\}, \{0\}, \emptyset)$.

Negation
For a proposition p its negation $\neg p$ is given by

$$(\neg p)^+ = p^+ \Rightarrow \mathsf{P_f}(p^-) \qquad (\neg p)^- = p^+ \quad (\neg p)(a,b) \quad \text{iff} \quad \exists c \in ab.\, \neg p(b,c)$$

Accordingly, double negation of p is given by

$$(\neg\neg p)^+ = [p^+ \Rightarrow \mathsf{P_f}(p^-)] \Rightarrow \mathsf{P_f}(p^+) \qquad (\neg\neg p)^- = p^+ \Rightarrow \mathsf{P_f}(p^-)$$

$$(\neg\neg p)(a,b) \quad \Leftrightarrow \quad \exists c \in ab.\, \forall d \in bc.\, p(c,d)$$

because $(\neg\neg p)(a,b) \Leftrightarrow \exists c \in ab.\, \neg(\neg p)(b,c) \Leftrightarrow \exists c \in ab.\, \neg\big(\exists d \in bc.\, \neg p(c,d)\big)$.

Lawvere Equality
For a set I the equality predicate $\mathsf{eq}_I \in {\mathsf{dn}_m}^{I\times I}$ is given by

$$\mathsf{eq}_I(i,j)^+ = \{0\} \cup \{1 \mid i = j\} \qquad \mathsf{eq}_I(i,j)^- = \{0\}$$

$$\mathsf{eq}_I(i,j)(a,0) \quad \text{iff} \quad (a = 1 \wedge i = j)$$

for all $i, j \in I$ as one easily shows that $\mathsf{eq}_I \dashv\vdash \exists_{\delta(I)}(\top)$. Notice that, alternatively, one may define eq_I as $\mathsf{eq}_I(i,j)^+ = \{0\} = \mathsf{eq}_I(i,j)^-$ and $\mathsf{eq}_I(i,j) \equiv i = j$.

Definition 4.1. The *modified Dialectica Topos* $\mathbf{DN}_m$ is defined as $\mathbf{Set}[\mathsf{dn}_m]$, the topos obtained from dn_m by the tripos-to-topos construction.

Lemma 4.2. *The tripos dn_m is not 2-valued, i.e. there are propositions which are neither true nor false. Consequently, the topos $\mathbf{DN}_m$ is not 2–valued either.*

Proof. A proposition p is true iff $\exists a \in p^+. \forall b \in p^-.\, p(a,b)$ and p is false iff $\exists e \in p^+ \to \mathsf{P_f}(p^-). \forall a \in p^+. \exists b \in e \cdot a.\, \neg p(a,b)$.

Let f be a function growing faster than any total recursive function[2]. Obviously, the proposition $p = (\mathbb{N}, \mathbb{N}, \{\langle n, m\rangle \mid f(n) \neq m\})$ is neither true nor false. The latter follows since for sufficiently big $a \in \mathbb{N}$ all elements of $e \cdot a$ are $< f(a)$. $\square$

For $\mathbf{DN}_m$ this is *not* the case as its $\neg\neg$-sheaves are not equivalent to $\mathbf{Set}$. Actually, it is not clear how to provide a sufficiently simple characterization of the $\neg\neg$–sheaves of $\mathbf{DN}_m$ avoiding the intricacies of double negation.

[2]Let $(f_n \mid n \in \mathbb{N})$ be some enumeration of all total recursive functions and define $f(n) = 1 + \max_{i \leq n} f_i(n)$.

4.1 Relation of $\mathbf{DN}_m$ to Number Realizability

We will define a connected geometric morphism from dn_m to m. Let the maps $q^* \colon \Sigma_{\mathsf{m}} \to \Sigma_{\mathsf{dn}_m}$ and $q_* \colon \Sigma_{\mathsf{dn}_m} \to \Sigma_{\mathsf{m}}$ be defined as

$$q^*(A^a, A^p) = \big(A^p, \{0\}, A^a \times \{0\}\big)$$
$$q_*(X, Y, R) = \big(\{x \in X \mid \forall y \in Y.\, R(x,y)\}\,,\, X\,\big)\ .$$

composing q with the connected geometric morphism $v : \mathsf{m} \to \mathsf{e}$ yields a connected geometric morphism from dn_m to e.

Natural Number Object Since inverse image parts of geometric morphisms preserve natural number objects the nno of $\mathbf{DN}_m$ is given by $N_{\mathbf{DN}_m} = q^*v^*(N_{\mathbf{Eff}}) = (\mathbb{N}, \approx_N)$ where $[n \approx_N m] = (\{0\} \cup \{n+1 \mid n = m\}, \{0\}, \{(n+1, 0) \mid n = m\})$.

Proposition 4.3. *Maps $F : q^*v^*(X, \sim) \to q^*v^*(\mathbb{N}, \sim)$ in $\mathbf{DN}_m$ are in bijective correspondence with maps from $(X, \sim)$ to $(\mathbb{N}, \sim)$ in* **Eff**.

The proof is similar to that of Corollary 2.15.

4.2 First Order Logic in $\mathbf{DN}_m$

We are going to compare the interpretation of formulas ϕ of first order logic over the natural numbers in $\mathbf{DN}_m$ with the Diller-Nahm translated (see [DN74]) formula ϕ^{DN} in the standard model **HRO**. The formula ϕ^{DN} has the form $\exists u \forall x \phi_{DN}(n, u, x)$, where $n : N^k$ and ϕ_{DN} is an quantifier-free formula of HA. The interpretation of ϕ^{DN} in **HRO** is denoted $\exists u : \mathsf{tp}_1(\phi) \forall x : \mathsf{tp}_2(\phi).R_{DN}(\phi)(n)(u, x)$, where $R_{DN}(\phi)(n)(u, x) \subseteq \mathsf{tp}_1(\phi) \times \mathsf{tp}_2(\phi)$, and is defined inductively as follows:

$\phi \equiv t = s$: $\mathsf{tp}_1(\phi) = \mathsf{tp}_2(\phi) = \{0\}$, $R_{DN}(\phi)(n)(0, 0)$ iff $t(n) = s(n)$.

$\phi \equiv \alpha \wedge \beta$: $\mathsf{tp}_1(\phi) = \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_1(\beta)$, $\mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \oplus \mathsf{tp}_2(\beta)$, $R_{DN}(\phi)(n)((u, v), \kappa(x))$ iff $R_{DN}(\alpha)(n)(u, x)$ and $R_{DN}(\phi)(n)((u, v), \kappa'(y))$ iff $R_{DN}(\beta)(n)(v, y)$.

$\phi \equiv \alpha \to \beta$:

$$\begin{array}{lcl} \mathsf{tp}_1(\phi) & = & \mathsf{tp}_1(\alpha) \Rightarrow \mathsf{tp}_2(\beta) \\ \mathsf{tp}_2(\phi) & = & \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) \\ R_{DN}(\phi)(n)(f, F, a, b) & \equiv & (\forall x \in F(a, b).R_{DN}(\alpha)(n)(a, x)) \supset R_{DN}(\beta)(n)(fa, b) \end{array}$$

$\phi \equiv \exists z.\alpha(z)$: $\mathsf{tp}_1(\phi) = N \otimes \mathsf{tp}_1(\alpha)$, $\mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha)$, $R_{DN}(\phi)(n)(k, a, b) \equiv R_{DN}(\alpha)(n, k)(a, b)$

$\phi \equiv \forall z.\alpha(z)$: $\mathsf{tp}_1(\phi) = N \Rightarrow \mathsf{tp}_1(\alpha)$, $\mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \otimes N$, $R_{DN}(\phi)(n)(h, b, k) \equiv R_{DN}(\alpha)(n, k)(h(k), b)$.

Lemma 4.4. *Any map $G : N^k_{\mathbf{DN}_m} \to N_{\mathbf{DN}_m}$ in $\mathbf{DN}_m$ has the form $q^*v^*(F)$ for $F : N^k_{\mathbf{Eff}} \to N_{\mathbf{Eff}}$ in* **Eff**. *And G is represented by $q^*v^*([n = n] \wedge [s(n) = m])$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function.*

Proof. By Proposition 4.3 we have that q^*v^* is fully faithful. Maps $F : N^k_{\mathbf{Eff}} \to N_{\mathbf{Eff}}$ in **Eff** are represented by $[n = n] \wedge [s(n) = m]$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function. Thus G is represented by $q^*v^*([n = n] \wedge [s(n) = m])$ where $s : \mathbb{N}^k \to \mathbb{N}$ is a function. □

Recall that a formula $\phi(x_1, \ldots, x_k)$ with $x^k : N^k$ is interpreted as a subobject of $N^k_{\mathbf{DN}_m}$, i.e., a strict predicate over $(\mathbb{N}^k, \approx)$. We denote this interpretation by $\|\phi\|(\bar{n}) = (\|\phi\|^+(\bar{n}), \|\phi\|^-(\bar{n}), \|\phi\|(\bar{n}))$.

Proposition 4.5. *For every formula $\phi(x_1, \ldots, x_k)$ of HA and for every k-tuple $\bar{n} = n_1, \ldots, n_k$, there are primitive recursive functions*

$$\begin{array}{ll} t_\phi(\bar{n}) : \mathsf{tp}_1(\phi) \to \|\phi\|^+, & T_\phi(\bar{n}) : \mathsf{tp}_1(\phi) \times \|\phi\|^- \to \mathcal{P}_f(\mathsf{tp}_2(\phi)) \\ s_\phi(\bar{n}) : \|\phi\|^+ \to \mathsf{tp}_1(\phi), & S_\phi(\bar{n}) : \|\phi\|^+(\bar{n}) \times \mathsf{tp}_2(\phi) \to \mathcal{P}_f(\|\phi\|^-(\bar{n})) \end{array}$$

such that the following holds:

1. *for all $a \in \mathsf{tp}_1(\phi)$, $x \in \|\phi\|^-(\bar{n})$.*
$$\forall c \in T_\phi(\bar{n})(a, x).\, R_{DN}(\phi)(\bar{n})(a, c)$$
implies
$$\|\phi\|(\bar{n})(t_\phi(\bar{n})(a), x).$$
2. *For all $u \in \|\phi^+(\bar{n})\|$, $b \in \mathsf{tp}_2(\phi)$.*
$$\forall z \in S_\phi(\bar{n})(u, b).\, \|\phi\|(\bar{n})(u, z)$$
implies
$$R_{DN}(\phi)(\bar{n})(s_\phi(\bar{n})(u), b).$$

Proof. Proof is by induction on the structure of ϕ. □

Corollary 4.6. *For all formulas ϕ of HA,*

$$\mathbf{HRO} \models \|\phi^{DN}\| \qquad \textit{iff} \qquad \mathbf{DN}_m \models \|\phi\|.$$

Proof. $\mathbf{HRO} \models \|\phi^{DN}\|$ means that there is a natural number $a \in \mathsf{tp}_1(\phi)$ such that $\forall b : \mathsf{tp}_2.R_{DN}(\phi)(\bar{n})(a, b)$. And $\mathbf{DN}_m \models \|\phi\|$ means that there is a $u \in \|\phi\|^+(\bar{n})$ such that $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(u, x)$. Suppose $\forall b : \mathsf{tp}_2.R_{DN}(\phi)(\bar{n})(a, b)$ then by Proposition 4.5 we have $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(t_\phi(\bar{n})(a), x)$. On the other hand, suppose $\forall x : \|\phi\|^-(\bar{n}).\, \|\phi\|(\bar{n})(u, x)$ then by Proposition 4.5 we have $\forall b : \mathsf{tp}_2.R_{DN}(\phi)(\bar{n})(s_\phi(\bar{n})(u), b)$. □

4.3 Relation of $\mathbf{DN}_m$ to Modified Realizability

Recall that we have already defined a geometric morphism q from dn_m to m:

$$\begin{aligned} q^*(A^a, A^p) &= \big(A^p, \{0\}, A^a \times \{0\}\big) \\ q_*(X, Y, R) &= \big(\,\{x \in X \mid \forall y \in Y.\, R(x, y)\}\,,\, X\,\big)\ . \end{aligned}$$

We also write $q^* \colon \mathsf{m} \to \mathsf{dn}_m$ and $q_* \colon \mathsf{dn}_m \to \mathsf{m}$ for the morphisms of triposes induced by component-wise application.

Proposition 4.7. *The tripos morphisms defined above give rise to a connected geometric morphism $q \colon \mathsf{dn}_m \to \mathsf{m}$ as given by the adjunction $q^* \dashv q_*$ where q^* preserves finite limits and is full and faithful.*

q_* does not have a right adjoint as q_* does not preserve existential quantification.

Proof. It is a straight forward exercise to show that q is a geometric morphism and that q^* is full (it is faithful anyway!).

That q_* does not preserve existential quantification can be seen from the following counterexample. Let $p \in \mathsf{dn}_m{}^{\mathbb{N}}$ be defined as

$$p_n^+ = \{0\} \qquad p_n^+ = \{0, n+1\} \qquad p_n = \{\langle 0,0\rangle\}$$

for $n \in \mathbb{N}$. We write $\exists_N$ for existential quantification along the terminal projection $\mathbb{N} \Rightarrow 1$. We have $\exists_N p = (\{0,1\}, \{0,1\}{\to}\{0\}, \{1\}{\otimes}(\{0,1\}{\Rightarrow}\{0\}))$ and, accordingly, $q_*\exists_N p = (\{1\}, \{0,1\})$. On the other hand $(q_*p)_n = q_*(p_n) = (\emptyset, \{0\})$ for all $n \in \mathbb{N}$ and accordingly $\exists_N q_* p = (\emptyset, \{0,1\})$. Thus, $q_*\exists_N p$ and $\exists_N q_* p$ are not equivalent as the former is true and the latter is false. □

5 Relation Between the Triposes d/dn and $\mathsf{d}_m/\mathsf{dn}_m$

Recall that d_m is the closed subtripos of d that we get from $c_{p^*(U)}(A) = p^*(U) \vee A$, where $p^*(U) = (\{0\}, \{0\}, \emptyset)$, which corresponds to the closed subtripos **Mod** of $\mathbf{Eff}_2$.

$$\begin{tikzcd} \mathbf{Dia}_m \arrow[d, hook] & \mathbf{Mod} \arrow[d, hook] \\ \mathbf{Dia} & \mathbf{Eff}_2 \end{tikzcd}$$

and that dn_m is the closed subtripos of dn from the pullback

$$\begin{tikzcd} \mathbf{DN}_m \arrow[r] \arrow[d, hook] & \mathbf{Mod} \arrow[d, hook] \\ \mathbf{DN} \arrow[r, "q"'] & \mathbf{Eff}_2 \end{tikzcd}$$

We now show that there is a fibred adjunction between d and dn and between d_m and dn_m.

For the fibred adjunction between the modified Diller Nahm tripos and the modified Dialectica tripos we first note that the underlying generic sets Σ_{dn_m} and Σ_{d_m} are the same. Over 1, the fibred adjunction is given by the pair of maps

$$\begin{aligned} id\colon \Sigma_{\mathsf{d}_m} \to \Sigma_{\mathsf{dn}_m} \qquad & (X,Y,A) \mapsto (X,Y,A)\,, \\ !\colon \Sigma_{\mathsf{dn}_m} \to \Sigma_{\mathsf{d}_m} \qquad & (X,Y,A) \mapsto !(X,Y,A) = (X, \mathsf{P_f}Y, !A) \end{aligned}$$

where $!A(x,s)$ if $\forall y \in s.\ A(x,y)$ (here we assume a standard encoding of finite sets of natural numbers by natural numbers and just write $\in$ for the encoded membership operation).

Lemma 5.1. *The above two maps induce a fibred adjunction between the modified Diller-Nahm tripos* dn_m *and the modified Dialectica tripos* d_m*, with* $! \dashv id$*. Moreover,* $!(p \wedge q) \dashv\vdash !p \otimes !q$ *for all* $p, q \in \Sigma$

Proof. Straightforward verification. □

To sum up, we have the following diagram of fibred adjunctions, where, however, only some give rise to geometric morphisms:

$$\begin{tikzcd}
 & \mathsf{dn}_m \arrow[rr, shift left, "!"] \arrow[d, shift left, "q_*"] & & \mathsf{d}_m \arrow[ll, shift left, "id"] \arrow[rr, shift left, "F"] \arrow[d, shift left, "H_*"] \arrow[dll, shift left, "p_*"] & & \mathbf{Set} \arrow[ll, shift left, "\nabla"] \\
 & \mathsf{m} \arrow[u, shift left, "q^*"] \arrow[rr, shift left, "v_*"] \arrow[urr, shift left, "p^*"] \arrow[dl, shift left, "\Gamma"] & & \mathsf{e} \arrow[ll, shift left, "v^*"] \arrow[u, shift left, "H^*"] \arrow[rr, shift left, "i_*"] \arrow[dl, shift left, "\Gamma"] \arrow[dr, shift left, "j^*"] & & \mathsf{dn} \arrow[ll, shift left, "i^*"] \\
\mathbf{Set} \arrow[ur, shift left, "\Delta"] \arrow[rr, "\cong"] & & \mathbf{Set} \arrow[ur, shift left, "\Delta"] & & \mathsf{d} \arrow[ul, shift left, "j_*"] &
\end{tikzcd}$$

where H, v, and q all are connected geometric morphisms, so they lift to surjective geometric morphisms on the induced toposes, and i and j are open geometric inclusions, so they lift to open geometric inclusions. The left adjoints of the adjunctions $! \dashv id$, $\nabla \dashv F$ and $p^* \dashv p_*$ are all full and faithful. At topos level we have the following geometric morphisms

$$\begin{tikzcd}
 & \mathbf{DN}_m \arrow[d, "q"] & & \mathbf{Dia}_m \arrow[d, "H"] & & \\
 & \mathbf{Mod} \arrow[rr, "v"] & & \mathbf{Eff} \arrow[rr, hook, "i"] \arrow[dr, hook, "j"] & & \mathbf{DN} \\
\mathbf{Mod}_{\neg\neg} \arrow[ur, hook] \arrow[rr, "\cong"] & & \mathbf{Eff}_{\neg\neg} \arrow[ur, hook] & & \mathbf{Dia} &
\end{tikzcd}$$

with i, j open inclusions.

6 A Fibration for the Standard Interpretation of Dialectica

We have presented four triposes/toposes, and we have seen that first order logic of $\mathbf{DN}_m$ corresponds exactly to the Diller-Nahm interpretation, but the logic of $\mathbf{Dia}_m$ is that of the Copenhagen interpretation, and *not* that of Dialectica. The reason is that when we drop the requirement of decidability, a different conjunction and implication are forced upon us. And if we want to have a tripos (which we would like because it is higher order) we have to drop decidability, it seems. So we argue that: If we want a tripos we have to drop decidability. If we drop decidability it has already been made clear that conjunction and implication are different from those of Gödel. In this section we briefly describe what kind of fibration would correspond exactly to the Dialectica interpretation.

We have previously defined the standard interpretations $R_C(\phi)$ for the Copenhagen interpretation and $R_{DN}(\phi)$ for the Diller-Nahm interpretation and showed how they correspond to first order Heyting arithmetic in the toposes $\mathbf{Dia}_m$ and

$\mathbf{DN}_m$. The question is: is there a similar topos for the standard interpretation R_D (which we define in Section 6.2) of the Dialectica interpretation? We haven't got an answer for this question, but in the following we argue that R_D does not correspond to a subtopos or fragment of $\mathbf{Dia}_m$ as one might have thought. We then go on and describe a fibration that provides a first order structure for R_D (but not higher order, as a tripos is, and not even higher typed).

It is a condition of the Dialectica interpretation that atomic formulas must be recursively decidable, one might therefore try to find a subtripos of d_m that satisfies this. The obvious thing to try is

$$\Sigma_{\mathsf{d}_{m,rec}} = \{(U, X, R) \in \Sigma_{\mathsf{d}} \mid 0 \in U,\ R \text{ recursive}\}$$

however, it is not hard to see that that the quantifiers do not preserve the property that the predicates are decidable. Hence, we do not get a subtripos in this way. One could define a tripos by $FinSet(-, \Sigma_{\mathsf{d}_{m,rec}})$ but this does not have a natural numbers object. Since we are interested in the logic of nno, we will now take a closer look at this part of the topos.

6.1 Logic of the Natural Numbers Object in $\mathbf{Dia}_m$

Lemma 6.1. *For a strict predicate $A_n = (A^+, A^-, A)$, if A is recursive then $\wedge \equiv \otimes$ hence $\rightarrow \equiv \multimap$.*

Proof. We need to show that $\otimes$ has both projections and diagonal, then $\otimes \equiv \wedge$ and since adjoints are unique up to iso, we then get $\rightarrow \equiv \multimap$. Diagonal: Let $f : A_n^+ \Rightarrow A_n^+ \otimes A_n^+$. be $f(u) = (u, u)$ and define $F : A_n^+ \otimes A_n^- \otimes A_n^- \Rightarrow A_n^-$ by

$$F(u, x, x') = \begin{cases} \delta(u) & \text{if} \quad \mu(u) = 0 \\ x & \text{if} \quad \mu(u) = n+1 \text{ and } \neg A_n(u, x) \\ x' & \text{if} \quad \mu(u) = n+1 \text{ and } A_n(u, x) \end{cases}$$

where δ and μ are the functions defined in Proposition 2.11.

Projections: We show that $A \otimes B \vdash A$. Define $f : A_n^+ \otimes B_n^+ \Rightarrow A_n^+$ by $f(u, v) = u$, and define $F : A_n^+ \otimes B_n^+ \otimes A_n^- \Rightarrow A_n^- \otimes B_n^-$ by $F(u, v, x) = (x, \delta(0))$. Notice that we use the fact that A is strict to get a dummy element in A_n^-. □

Proposition 6.2. *Recursiveness for strict predicates over N is preserved by $\wedge, \rightarrow, \vee$ and $\forall$.*

Proof. Clearly recursiveness is preserved by the propositional connectives, we just need to show $\forall$. Suppose A, B are recursive, then $A \rightarrow B \equiv A \multimap B$. Suppose A is a strict predicate over N^{k+1} then

$$\begin{array}{lcl} \|\forall y : N.A(\bar{n}, y)\| & = & [\bar{n} = \bar{n}] \wedge \forall y \in \mathbb{N}.[y = y] \multimap A(\bar{n}, y) \quad = \\ {[\bar{n} = \bar{n}]} & \wedge & \forall y : \mathbb{N}.\big(\{(f, F) : \{0, y+1\} \Rightarrow A^+(\bar{n}, y) \otimes \{0, y+1\} \times A^-(\bar{n}, y) \Rightarrow \{0\}\}, \\ & & \{0, y+1\} \times A^-(\bar{n}, y), \\ & & u = y+1 \supset A(\bar{n}, y)(fu, x)\big) \end{array}$$

so the predicate part is

$$\gamma(\bar{n}, y)(f, F, u, x) \equiv \forall y \in \mathbb{N}.(u = y + 1 \supset A(\bar{n}, y)(fu, x))$$

which is clearly recursive. □

Remark 6.3. Recursiveness for strict predicates over N does not seem to be preserved by $\exists$.

Remark 6.4. If A and B are decidable predicates over N in $\mathbf{Dia}_m$, then $A \wedge B, A \to B, A \vee B$ are also decidable (by a purely logical argument). However, it does not seem to be the case that decidability is preserved by $\forall, \exists$ in the topos.

6.2 A First order Fibration

We are going to compare the interpretation of formulas ϕ of first order logic over the natural numbers in a fibration Dial_{rec} with the Dialectica translated formulas ϕ^D in the standard model **HRO**. The formulas ϕ^D has the form $\exists u \forall x \phi_D(n, u, x)$, where $n : N^k$ and ϕ_D is quantifier-free formula of HA. The interpretation of ϕ^D in **HRO** is denoted $\exists u : \mathsf{tp}_1(\phi) \forall x : \mathsf{tp}_2(\phi). R_D(\phi)(n)(u, x)$, where $R_D(\phi)(n)(u, x) \subseteq \mathsf{tp}_1(\phi) \times \mathsf{tp}_2(\phi)$ is a *recursive subset*. We define ϕ^D inductively as follows:

$\phi \equiv t = s$**:** $\mathsf{tp}_1(\phi) = \mathsf{tp}_2(\phi) = \{0\}$, $R_D(\phi)(n)(0,0)$ iff $t(n) = s(n)$.

$\phi \equiv \alpha \wedge \beta$**:**

$$\begin{array}{lll} \mathsf{tp}_1(\phi) = \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_1(\beta), & & \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \oplus \mathsf{tp}_2(\beta), \\ R_C(\phi)(n)((u,v), \kappa(x)) & \text{iff} & R_C(\alpha)(n)(u,x) \text{ and} \\ R_C(\phi)(n)((u,v), \kappa'(y)) & \text{iff} & R_C(\beta)(n)(v,y). \end{array}$$

Notice that conjunction may also be constructed in a different way corresponding exactly to Gödel's Dialectica interpretation:
$\mathsf{tp}_1(\phi) = \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_1(\beta)$, $\quad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \otimes \mathsf{tp}_2(\beta)$,
$R_D(\phi)(n)((u,v),(x,y))$ iff
$R_D(\alpha)(n)(u,x)$ and $R_D(\beta)(n)(v,y)$.

$\phi \equiv \alpha \to \beta$**:**

$$\begin{array}{lll} \mathsf{tp}_1(\phi) & = & \{h, H : (\mathsf{tp}_1(\alpha) \Rightarrow \mathsf{tp}_1(\beta)) \otimes (\mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) \Rightarrow \mathsf{tp}_2(\alpha)\} \\ \mathsf{tp}_2(\phi) & = & \mathsf{tp}_1(\alpha) \otimes \mathsf{tp}_2(\beta) \\ R_D(\phi)(n)(h, H, a, b) & \equiv & R_D(\alpha)(n)(a, H(a,b)) \supset R_D(\beta)(n)(fa, b) \end{array}$$

$\phi \equiv \exists z.\alpha(z)$**:**

$$\begin{array}{l} \mathsf{tp}_1(\phi) = N \otimes \mathsf{tp}_1(\alpha), \qquad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha), \\ R_D(\phi)(n)(k, a, b) \equiv R_D(\alpha)(n, k)(a, b). \end{array}$$

$\phi \equiv \forall z.\alpha(z)$**:**

$$\begin{array}{l} \mathsf{tp}_1(\phi) = N \Rightarrow \mathsf{tp}_1(\alpha), \qquad \mathsf{tp}_2(\phi) = \mathsf{tp}_2(\alpha) \otimes N, \\ R_D(\phi)(n)(h, a, k) \equiv R_D(\alpha)(n, k)(h(k), a). \end{array}$$

The following is already described in [Hyl02]. We now define the fibration Dial_{rec} over $\mathcal{P}(\mathbb{N})_0$. In the fibre over I, we have objects: triples (U, X, α) with $0 \in U, X \subseteq \mathbb{N}$ and $\alpha \subseteq I \times U \times X$, where α is recursive. Maps from (U, X, α) to (V, Y, β) are diagrams in the simple slice category of $\mathcal{P}(\mathbb{N})_0$ over I: $f, F \in \mathbb{N}$ such that

$$f : I \otimes U \Rightarrow V, \quad F : I \otimes U \otimes Y \Rightarrow X$$

and such that for all $i : I, u : U, y : Y$,

$$\alpha(i, u, F(i, u, y)) \supset \beta(i, f(i, u), y)$$

In case there exist such $f, F \in \mathbb{N}$ we say that $(U, X, \alpha) \vdash (V, Y, \beta)$, and we have a preordered fibration.

Truth and Falsity
The terminal object and the initial object are given by

$$\top = \big(\{0\}, \{0\}, I \otimes \{(0,0)\}\big) \quad \text{and} \quad \bot = \big(\{0\}, \{0\}, \emptyset\big)$$

respectively.

Conjunction
For $A_0, A_1 \in \mathrm{Dial}_{rec}(I)$ their conjunction $A_0 \wedge A_1$ is given by $(A_0 \wedge A_1)^+ = A_0^+ \otimes A_1^+$, $(A_0 \wedge A_1)^- = A_0^- \otimes A_1^-$ and $(A_0 \wedge A_1)(i, \langle n_0, n_1\rangle, \langle m_0, m_1\rangle) \equiv A_0(i, n_0, n_1) \wedge A_1(i, n_1, m_1)$.

Implication
For $A, B \in \mathrm{Dial}_{rec}(I)$ their implication $A \to B$ is given by

$$(A \to B)^+ = \big\{\langle f, F\rangle \in (A^+ \Rightarrow B^+) \otimes (A^+ \otimes B^- \Rightarrow (A^-)\big\}$$
$$(A \to B)^- = A^+ \otimes B^-$$
$$(A \to B)(i, \langle f, F\rangle, \langle u, y\rangle) \equiv A(i, u, F(u, y) \supset B(i, f(u), y)$$

Disjunction
For $A_0, A_1 \in \Sigma_{\mathsf{d}}$ their disjunction $A_0 \vee A_1$ is given by $(A_0 \vee A_1)^+ = A_0^+ \oplus A_1^+$, $(A_0 \vee A_1)^- = A_0^- \oplus A_1^-$ and $(A_0 \vee A_1)(i, \langle k, n\rangle, \langle j, m\rangle) \equiv k = j \ \supset\ A_k(i, n, m)$.

Universal Quantification
We only have quantifiers for projections. Suppose $A \in \mathrm{Dial}_{rec}(I \times J)$ and $\pi \colon I \times J \to J$ in $\mathcal{P}(\mathbb{N})_0$. Then $\forall_\pi(A) \in \mathrm{Dial}_{rec}(J)$ is given by

$$\forall_\pi(A)^+ = I \Rightarrow A^+$$
$$\forall_\pi(A)^- = I \otimes A^-$$
$$\forall_\pi(A)(j, f, i, x) \equiv A(i, j, f(i), x)$$

Existential Quantification
Suppose $A \in \mathrm{Dial}_{rec}(I \times J)$ and $\pi \colon I \times J \to J$ in $\mathcal{P}(\mathbb{N})_0$. Existential quantification of A along π is given by

$$\exists_\pi(A)^+ = I \otimes A^+$$
$$\exists_\pi(A)^- = A^-$$
$$\exists_\pi(A)(j, (i, u), x) \equiv A(i, j, u, x)$$

Equality
We have for each $I, J \in \mathcal{P}(\mathbb{N})_0$,

$$\mathrm{Eq}_{I,J} \dashv \delta(I,J)^*$$

where $\delta(I,J) = \langle id, \pi' \rangle : I \otimes J \to (I \otimes J) \otimes J$ and where $\mathrm{Eq}_{I,J} : \mathrm{Dial}_{rec}(I \otimes J) \to \mathrm{Dial}_{rec}((I \otimes J) \otimes J)$ is defined by

$$\mathrm{Eq}_{I,J}(A) = (A^+, A^-, \hat{A}(i,j,j',u,x))$$

where

$$\hat{A}(i,j,j',u,x) = \begin{cases} A(i,j,u,x) & \text{if} \quad j = j' \\ \bot & \text{if} \quad j \neq j'. \end{cases}$$

Notice that $\hat{A}$ is recursive because equality on $\mathbb{N}$ is recursive, i.e., this is intensional equality. It is straight forward to verify that Eq satisfy the Beck-Chevalley condition: For each map $h : K \to I$

$$\mathrm{Eq}_{K,J}(h \times J)^* \cong ((h \times J) \times J)^* \mathrm{Eq}_{I,J}$$

and also the Frobenius property:

$$\mathrm{Eq}_{I,J}(\delta^*(A) \wedge B) \cong A \wedge \mathrm{Eq}_{I,J}(B)$$

for all $A \in \mathrm{Dial}_{rec}((I \otimes J) \otimes J)$ and $B \in \mathrm{Dial}_{rec}(I \otimes J)$. Now for a type $I \in \mathcal{P}(\mathbb{N})_0$ we derive the equality predicate $\mathsf{eq}_I \in \mathrm{Dial}_{rec}(I \otimes I)$:

$$\mathsf{eq}_I(i,j) = \mathrm{Eq}_{1,I}(\top) = \begin{cases} \top & \text{if} \quad i = j \\ \bot & \text{if} \quad i \neq j \end{cases}$$

Proposition 6.5. *For all formulas ϕ of HA we have $\|\phi^D\|$ in* **HRO** *is precisely $\|\phi\|$ in* Dial_{rec}.

Proof. By induction on the structure of ϕ. ☐

References

[BBTS05] Lars Birkedal Bodil Biering and Noah Torp-Smith. Bi hyperdoctrines, higher-order separation logic, and abstraction. *Proceedings of European Symposium on Programming (ESOP'05)*, pages 233–247, 2005. 1.2, 2.2, 2.2

[Bie07a] Bodil Biering. An analysis of Cartesian closure in Dialectica categories. *Unpublished draft*, 2007. (document), 1

[Bie07b] Bodil Biering. The Copenhagen interpretation. 2007. Submitted. 1, 2.3

[Bie08] Bodil Biering. *Dialectica Interpretations: a Categorical Analysis.* PhD thesis, IT-University of Copenhagen, 2008. 1.1, 1.2, 2.1, 2.1, 3.1

[Bir99] L. Birkedal. *Developing Theories of Types and Computability.* PhD thesis, School of Computer Science, Carnegie Mellon University, December 1999. 2.1

[DN74] J. Diller and W. Nahm. Eine Variante zur Dialectica–Interpretation der Heyting–Arithmetik endlicher Typen. *Archiv*, 16:49–66, 1974. (document), 4.2

[dP89] V.C.V. de Paiva. The Dialectica categories. In *Contemporary Mathematics*, volume 92, 1989. 1, 2.2

[Fre07] Jonas Frey. A universal characterization of the tripos-to-topos construction. Technical report, Technische Universität Darmstadt, 2007. 2.1

[Göd58] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958. (document)

[HJP80] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980. (document), 1.3, 1, 2.5, 2.1

[Hyl82] J.M.E. Hyland. The effective topos. In A.S. Troelstra and D. Van Dalen, editors, *The L.E.J. Brouwer Centenary Symposium*, pages 165–216. North Holland Publishing Company, 1982. (document), 1.3

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999). 6.2

[Joh02] Peter T. Johnstone. *Sketches of an elephant: a topos theory compendium.*, volume 44 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, Oxford, 2002. 3.1

[vO97] J. van Oosten. The modified realizability topos. *Journal of Pure and Applied Algebra*, 116:273–289, 1997. 1.1, 2.1, 2.4

[vO08] J. van Oosten. *Realizability - An Introduction to its Categorical Side*. Studies in Logic, 152, 2008. 2.1, 2.3

Chapter 3

A Unified View on the Dialectica Triposes

This short note is only the beginning of a story. The initial ideas are due to Martin Hyland. It can be read independently, though it refers to the triposes introduced in Chapter 2.

In Chapter 2 we described realizability triposes for various functional interpretations and their relationships via geometric morphisms or indexed adjunctions. We use these adjunctions to define comonads on Girard categories, and we show that each of the triposes in Chapter 2 is isomorphic to a Kleisli category for a comonad on a Girard category. We also show that if the comonad is Girardian, then the Kleisli category is a tripos.

The Girard categories were first introduced in [dP91]. In [Oli08], Paulo Oliva presents a unifying framework for functional interpretations which has some similarities with our work. In Chapter 4 the non-Girardian comonad is studied in a more general setting.

References

[dP91] V.C.V de Paiva. *The Dialectica Categories, Technical Report 213 from Computer Laboratory (Thesis)*. PhD thesis, University of Cambridge, 1991.

[Oli08] Paulo Oliva. Functional interpretations of linear and intuitionistic logic. *special issue of Information and Computation*, 2008.

A Unified View on the Dialectica Triposes

Bodil Biering

This is a brief note in which we explain how to get a unified view of the realizability triposes introduced in [Bie08, Chapter 2] via comonads on indexed Girard categories. We show how this setting may be used to prove that the triposes from [Bie08, Chapter 2] indeed are triposes, without actually calculating the structure. In [dP91] it is explained how to get model of linear logic with modality by using a (Girardian) comonad on a Dialectica category. In this note we start our with a Girard category instead of a Dialectica category, and we give several Girardian comonads and one non-Girardian comonad, and thus we get several different realizability models of linear logic with modality. The idea of unifying functional interpretations via comonads on a Girard category has been introduced independently and in a very different (syntactic) setting in [Oli08].

1 Indexed Preordered Girard Categories

We define an indexed preordered Girard category G over **Set** as follows. In the fibre over I, the objects are the objects of $\mathsf{d}(I)$ that is, I-indexed triples $A_i = (U_i, X_i, A_i)$, where $A_i \subseteq U_i \times X_i \subseteq \mathbb{N} \times \mathbb{N}$. The order is defined by

$$(U_i, X_i, A_i) \vdash (V_i, Y_i, B_i) \qquad \text{iff}$$

there exists $f, F \in \mathbb{N}$ with $f \in \bigcap_i (U_i \Rightarrow V_i)$ and $F \in \bigcap_i (Y_i \Rightarrow X_i)$ such that for all $i \in I, u \in U_i, y \in Y_i$ we have

$$A_i(u, F(y)) \supset B_i(f(u), y).$$

We also define a modified version of the indexed Girard category G, which we call G_m. G_m is the subcategory of G satisfying $0 \in U_i$ for all objects (U_i, X_i, A_i).

We now give the closure properties of G and G_m.

Proposition 1.1. *All fibres of the indexed preorders G and G_m have finite limits and are symmetric monoidal closed, i.e., we have $\otimes, \multimap, \wedge, \top$. Furthermore G and G_m have products ($\forall$).*

Proof. All of the structure that we are interested in is defined similarly in G and in G_m.

Symmetric Monoidal Closed Structure

The tensor is defined pointwise as

$$(U, X, A) \otimes (V, Y, B) = (U \otimes V, X^V \otimes Y^U, A \otimes B)$$

where $A \otimes B(u, v, f, g) \equiv A(u, fv)$ and $B(v, gu)$. The unit I is

$$(\{0\}, \{0\}, \{(0,0)\})$$

The tensor product has a right adjoint $\multimap$ defined by

$$(V, Y, B) \multimap (Z, W, C) = (Z^V \otimes Y^W, V \otimes W, B \multimap C)$$

where $(B \multimap B)(f, F, v, w) \equiv B(v, F(w)) \supset C(f(v), w)$.

Conjunction

$$(U, X, A) \wedge (V, Y, B) = (U \otimes V, X \oplus Y, A \wedge B)$$

where $A \wedge B(u, v, (i, z)) \equiv \begin{cases} A(u, z) & \text{if} \quad i = 0 \\ B(v, z) & \text{if} \quad i = 1 \end{cases}$

Truth
We have $\top = (\{0\}, \emptyset, \emptyset)$.

Universal Quantification
For a function $u : I \to J$ we have

$$(\forall_u A)_j = (\bigcap_{i \in u^{-1}(j)} A_i^+, \bigcup_{i \in u^{-1}(j)} A_i^-, \tilde{A}_j)$$

where $\tilde{A}_j(a, b)$ iff $\forall i \in u^{-1}(j). b \in A_i^- \Rightarrow A_i(a, b)$. □

2 Comonads and Triposes

Let $P = \mathbf{Set}(-\Sigma)$ be a canonically presented preordered fibration. An indexed comonad on P then amounts to a comonad $\langle L, \mu, \epsilon \rangle$ on $P(1)$, *i.e.*, a functor $L\colon P(1) \to P(1)$ such that, for all $p \in P(1)$

$$Lp \vdash p \qquad \text{and} \qquad Lp \vdash LLp$$

hold in $P(1)$.

Theorem 2.1. *Let $P = \mathbf{Set}(-\Sigma)$ be a canonically presented preordered fibration. Suppose P is a fibred smcc with fibred finite limits and products. let L be a comonad on P satisfying*

$$L(p \wedge q) \dashv\vdash Lp \otimes Lq$$

in $P(1)$. Then P_L, the Kleisli category for the comonad L is a canonically presented **Set***-tripos.*

Proof. It suffices to show that P_L has $\top, \wedge, \to, \forall$ and a generic object. $\top, \wedge$ in P_L are inherited from P because $p \vdash^{P_L} \top$ iff $Lp \vdash^P \top$. And $p \vdash^{P_L} q \wedge r$ iff $Lp \vdash^P q \wedge r$ iff $Lp \vdash^P q$ and $Lp \vdash^P r$ iff $p \vdash^{P_L} q$ and $p \vdash^{P_L} r$.

$\to^{P_L}$ is given by $q \to^{P_L} r := Lq \multimap r$ since $p \wedge q \vdash^{P_L} r$ iff $L(p \wedge q) \vdash^P r$ iff $Lp \otimes Lq \vdash^P r$ iff $Lp \vdash^P Lq \multimap r$ iff $p \vdash^{P_L} Lq \multimap r$. Universal quantifications in P_L is as in P since $u^*p \vdash^{P_L} q$ iff $L(u^*p) \vdash^P q$ iff $u^*(Lp) \vdash^P q$ iff $Lp \vdash^P \forall_u q$ iff $p \vdash^{P_L} \forall_u q$. The generic object is id_Σ. □

Theorem 2.2 (Factorization Theorem)**.** *Suppose $f = (f^*, f_*) : P \to Q$ is an adjunction of canonically presented indexed preorders and f^* is full and faithful. Then $L = f^*f_*$ is a comonad on P with Kleisli category P_L and f can be factored as*

$$\begin{tikzcd} & P_L \arrow[dr, "K"] & \\ P \arrow[ur, "g"] \arrow[rr, "f"'] & & Q \end{tikzcd}$$

where $g^* = L$, $g_* = id$, and the comparison functor K is an iso, so we get $Q \cong P_L$.

Proof. It is well-known that when $f^* \dashv f_*$, f^*f_* defines a comonad. From the definition of a Kleisli category for a comonad we know that there is a diagram

$$\begin{tikzcd} & P_L \arrow[dr, "K"] & \\ P \arrow[ur, "g"] \arrow[rr, "f"'] & & Q \end{tikzcd}$$

where $g = (g^*, g_*)$ is an L-adjunction, i.e., $g^*g_* = L$, and K is unique with the properties

$$Kg_* = f_*, \qquad f^*K = g^*$$

We also know that on objects g is defined by $g^*(A) = LA$, $g_*(A) = A$ We now show that K has an inverse $K^{-1} = g_*f^*$. First we notice that $g_*g^* \cong id_{P_L}$, since $g_*g^*A = LA$ and in P_L we have $A \dashv\vdash LA$. Now using the above equations for K we get

$$\begin{array}{rcccl} Kg_*f^* & = & f_*f^* & \cong & id_Q, \\ g_*f^*K & = & g_*g^* & \cong & id_{P_L}, \end{array}$$

showing that K is an iso. □

Corollary 2.3. *Let P be an indexed preorder satisfying the requirements of Theorem 2.1. And let $f = (f^*, f_*) : P \to Q$ be an adjunction with f^* full and faithful and such that $L = f^*f_*$ satisfy $L(p \wedge q) \dashv\vdash Lp \otimes Lq$, then Q is a tripos and $Q \cong P_L$.*

From [Bie08, Chapter 2] we the following indexed adjunctions and geometric morphisms:

$$\begin{tikzcd} & \mathbf{Set}(-,2) \arrow[d, shift left, "\nabla"] & \\ \mathsf{dn} \arrow[r, shift left, "!"] & \mathsf{d} \arrow[u, shift left, "F"] \arrow[l, shift left, "id"] \arrow[r, shift left, "S"] \arrow[dl, shift left, "r_*"] \arrow[d, shift left, "p_*"] & G \arrow[l, shift left, "I"] \\ \mathsf{e} \arrow[ur, shift left, "r^*"] & \mathsf{e}_2 \arrow[u, shift left, "p^*"] & \end{tikzcd} \qquad (1)$$

For convenience we give the definition of the adjunctions her:

- $S \dashv I$. Where $I : G \to \mathsf{d}$ is the identity on objects and $I(f, F) = (f, F\pi)$. And the functor $S : \mathsf{d} \to G$ is defined by $S(U, X, A) = (U, (X + 1)^U, SA)$, where $SA(u, h)$ iff $\hat{A}(u, hu)$, and

$$\hat{A}(u,x) \equiv \begin{cases} A(u,x) & \text{if} \quad x \in X \\ \top & \text{if} \quad x \in 1. \end{cases}$$

  For morphisms we have $S(f, F) = (f, G)$, where $G : (Y+1)^V \to (X+1)^U$ is given by $G(h) = \lambda u.F(u, h(g(u)))$. The functor S is full and faithful.

- $p^* \dashv p_*$.

  $p^*(A, B) = (B, \{0\}, A\times\{0\}) \qquad p_*(A) = (\{u \in A^+ \mid \forall x \in A^-.A(u, x)\}, A^+)$

  The functor p^* is full and faithful.

- $r^* \dashv r_*$.

$$r^*(A) = (A, \{0\}, A \times \{0\}) \qquad r_*(A) = \{u \in A^+ \mid \forall x \in A^-.A(u, x)\}.$$

  r is a connected geometric morphism from d to e.

- $! \dashv id$.

$$!(A) = (A^+, \mathcal{P}_f(A^-), !A) \quad \text{where} \quad !A(u, s) \equiv \forall x \in s.A(u, x).$$

  The functor ! is full and faithful.

- $\nabla \dashv F$.

$$\nabla(I' \subseteq I)(i) = \begin{cases} (\{0,1\}, \{0\}, \{(0,1)\}) & \text{if} \qquad i \in I \\ (\{0,1\}, \{0\}, \emptyset) & \text{o.w.} \end{cases}$$

$$F(A_i)(i) = \begin{cases} 1 & \text{if} \qquad \exists u \in \cap_i A_i^+.\forall x.A_i^-.A(u,x) \\ 0 & \text{o.w.} \end{cases}$$

  The functor ∇ is full and faithful.

Composing the adjunctions we get five adjunctions going from G to each of the other indexed categories. Now, each of these adjunctions give rise to a comonad on G. We give the definitions of the five comonads:

- $L_{\mathsf{d}}(U, X, A) = SI(U, X, A) = (U, (X + 1)^U, SA)$
  where $SA(u, h) \equiv \hat{A}(u, hu)$.

- $L_s(U, X, A) = S\nabla FI(U, X, A) = (\{0, 1\}, (\{0\} + 1)^{\{0,1\}}, L_sA)$
  $\equiv (\{0, 1\}, \{0\}, L_sA)$
  where $L_sA(n, 0)$ iff $(n, 0) = (1, 0)$ and $\exists u \in U.\forall x \in X.A(u, x)$.

- $L_{\mathsf{dn}}(U, X, A) = (S!idI(U, X, A) = (U, (\mathcal{P}_f(X)+1)^U, L_{\mathsf{dn}}(A)) \equiv (U, \mathcal{P}_f(X)^U, L_{\mathsf{dn}}(A))$
  where $L_{\mathsf{dn}}(A)(u, h) \equiv \forall x \in h(u).A(u, x)$.

- $L_{\mathsf{e}}(U, X, A) = Sr^*r_*I(U, X, A) =$
  $(\{u \in U \mid \forall x \in X.A(u, x)\}, (\{0\} + 1)^{\{u\in U|\forall x\in X.A(u,x)\}}, L_{\mathsf{e}}(A))$
  $\equiv (\{u \in U \mid \forall x \in X.A(u, x)\}, \{0\}, \{u \in U \mid \forall x \in X.A(u, x)\} \otimes \{0\})$.

- $L_{\mathsf{e}_2}(U,X,A) = Sp^*p_*I(U,X,A) = (U,(\{0\}+1)^U, L_{\mathsf{e}_2}(A))$
  $\equiv (U,\{0\},\{u \in U \mid \forall x \in X.A(u,x)\} \otimes \{0\})$

We now turn to the modified Girard category G_m, for which we have a similar situation. Notice that instead of e_2 we have m and for dn we have dn_m. This is exactly as one would expect, since m is the "modified version" of e_2 and dn_m the modified version of dn. We have the following diagram of indexed adjunctions and geometric morphisms:

$$\begin{tikzcd}
& \mathbf{Set}(-,2) \arrow[d, shift left, "\nabla"] & \\
\mathsf{dn}_m \arrow[r, shift left, "!"] & \mathsf{d}_m \arrow[u, shift left, "F"] \arrow[l, shift left, "id"] \arrow[r, shift left, "S"] \arrow[dl, shift left, "H_*"] \arrow[d, shift left, "p_*"] & G_m \arrow[l, shift left, "I"] \\
\mathsf{e} \arrow[ur, shift left, "H^*"] & \mathsf{m} \arrow[u, shift left, "p^*"] &
\end{tikzcd} \tag{2}$$

Most of functors of this diagram have been defined above, we just need to add the following definitions: $H^* \dashv H_*$.

$H^*(A) = (succ(A) \cup \{0\}, \{0\}, succ(A) \otimes \{0\})$ $\qquad H_*(A) = \{u \in A^+ \mid \forall x \in X.A(u,x)\}$

H is a connected geometric morphism.

Again we can compose the adjunctions to get five comonads on G_m, the comonads are defined as follows:

- $K_{\mathsf{d}_m}(U,X,A) = SI(U,X,A) = L_{\mathsf{d}}(U,X,A) = (U,(X+1)^U, SA)$
  where $SA(u,h) \equiv \hat{A}(u,h(u))$.

- $K_s(U,X,A) = S\nabla FI(U,X,A) = L_s(U,X,A) = (\{0,1\},\{0\},K_sA)$
  where $K_sA(n,0)$ iff $(n,0)=(1,0)$ and $\exists u \in U.\forall x \in X.A(u,x)$.

- $K_{\mathsf{dn}_m}(U,X,A) = (S!idI(U,X,A) = L_{\mathsf{dn}}(U,X,A) = (U,(\mathcal{P}_f(X)+1)^U, K_{\mathsf{dn}_m}(A)) \equiv (U,\mathcal{P}_f(X)^U, K_{\mathsf{dn}_m}(A))$
  where $K_{\mathsf{dn}_m}(A)(u,h) \equiv \forall x \in h(u).A(u,x)$.

- $K_{\mathsf{e}}(U,X,A) = SH^*H_*I(U,X,A) =$
  $(succ(\{u \in U \mid \forall x \in X.A(u,x)\}) \cup \{0\}, \{0\}, K_{\mathsf{e}}(A))$
  where $K_{\mathsf{e}}(A)(u,0)$ iff $u \neq 0$.

- $K_{\mathsf{m}}(U,X,A) = Sp^*p_*I(U,X,A) = L_{\mathsf{e}_2}(U,X,A) =$
  $(U,\{0\},\{u \in U \mid \forall x \in X.A(u,x)\} \otimes \{0\})$

Proposition 2.4. *All the comonads $L_{(-)}$ and $K_{(-)}$ except for L_{d} and K_{d_m} satisfy the condition $L(p \wedge q) \dashv\vdash Lp \otimes Lq$.*

Proof. Easy verification. $\square$

Proposition 2.5. *The Kleisli category for each comonad L_X and K_Y, where $X \in \{\mathsf{d}, s, \mathsf{dn}, \mathsf{e}, \mathsf{e}_2\}$ and $Y \in \{\mathsf{d}_m, s, \mathsf{dn}_m, \mathsf{e}, \mathsf{m}\}$ is a tripos, and it is isomorphic to X or Y respectively.*

Proof. By Proposition 2.4 and Corollary 2.3 and using that the comonads are defined form the adjunctions in the diagrams (1) and (2). For the comonads L_d and K_{d_m} we can not use Theorem 2.1 (nor Corollary 2.3) to conclude that the Kleisli categories are triposes. This must be shown by hand, but it follows from Theorem 2.2 that the Kleisli category for L_d is d and for K_{d_m} is d_m. □

We collect all the results in a neat table.

G			G_m		
comonad	Kleisli	$L(p \wedge q) \equiv Lp \otimes Lq$	comonad	Kleisli	$K(p \wedge q) \equiv Kp \otimes Kq$
L_d	d	No	$K_{\mathsf{d}_m} = L_\mathsf{d}$	d_m	No
L_s	$\mathbf{Set}(-,2)$	Yes	$K_s = L_s$	$\mathbf{Set}(-,2)$	Yes
L_dn	dn	Yes	$K_{\mathsf{dn}_m} = L_\mathsf{dn}$	dn_m	Yes
L_e	e	Yes	K_e	e	Yes
L_{e_2}	e_2	Yes	$K_\mathsf{m} = L_{\mathsf{e}_2}$	m	Yes

Remark 2.6. Instead of working with the indexed categories, one may take on an internal view of things, and show that all of the indexed categories are externalizations of internal categories in **Eff**. The comonads and Kleisli constructions could then also be done internally. Seeing the indexed categories as externalizations of internal categories (or preorders) of **Eff** also explains why the ordering in the fibres of the Girard categories and of the triposes is uniform.

Future Work In [Bie08, Chapter 2] there is a precise relationship between d and d_m (d_m is a closed subtripos of d) and similarly between dn and dn_m. At the moment it is not clear if there is a similar relationship between G and G_m.

In [Oli08], Paulo Oliva treats Stein's interpretation and Bounded functional interpretation in a setting that seems to correspond to our comonadic setting. It is very likely that those two interpretations also fit into our comonadic framework.

The whole story of Girard categories and comonads could be lifted from the preordered setting to categories.

References

[Bie08] Bodil Biering. *Dialectica Interpretations: a Categorical Analysis.* PhD thesis, IT-University of Copenhagen, 2008. (document), 2, 2

[dP91] V. de Paiva. The Dialectica categories. Technical Report 213, University of Cambridge, 1991. (document)

[Oli08] Paulo Oliva. Functional interpretations of linear and intuitionistic logic. *special issue of Information and Computation*, 2008. (document), 2

Chapter 4

Cartesian Closed Dialectica Categories

This paper has been submitted for publication in Annals of Pure and Applied Logic. Most of the research for this paper was conducted during my stay in Cambridge in the Spring of 2006 with Martin Hyland as advisor. The only background that is really needed for this paper, is some basic knowledge about fibrations, as can be found in [Jac99]. It may also be nice to have a look in [Hyl02].

In this paper we extend the Dialectica categories to cloven fibrations, and we analyse how to obtain weakly Cartesian closed and Cartesian closed (variants of) Dialectica categories.

The idea for the exponent construction arose from the Dialectica tripos in Chapter 2, and the same constructions has since been used to define a new variant of the Dialectica interpretation, called *the Copenhagen Interpretation* (see Chapter 5). The work in this paper is also related to the work on comonads in Chapter 3.

References

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

[Jac99] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

Cartesian Closed Dialectica Categories

Bodil Biering, IT University of Copenhagen,
biering@itu.dk

Abstract

When Gödel developed his functional interpretation, also known as the Dialectica interpretation, his aim was to prove (relative) consistency of first order arithmetic by reducing it to a quantifier-free theory with finite types. Like other functional interpretations (e.g. Kleene's realizability interpretation and Kreisel's modified realizability) Gödel's Dialectica interpretation gives rise to category theoretic constructions that serve both as new models for logic and semantics and as tools for analysing and understanding various aspects of the Dialectica interpretation itself.

Gödel's Dialectica interpretation gives rise to the Dialectica categories (described by V. de Paiva in [dP89] and J.M.E. Hyland in [Hyl02]). These categories are symmetric monoidal closed and have finite products and weak coproducts, but they are not Cartesian closed in general. We give an analysis of how to obtain weakly Cartesian closed and Cartesian closed Dialectica categories, and we also reflect on what the analysis might tell us about the Dialectica interpretation.

1 Introduction

In this paper we analyse how to obtain Cartesian Closed Dialectica categories. The inspiration for exponent construction that we will give comes from a structure closely related to the Dialectica categories, namely the Dialectica tripos [BBLBCB07] (which is actually an indexed, preordered reflection of a Dialectica category). In order to do this analysis and also to find out whether the construction of the exponential in the tripos can be carried over to the Dialectica categories to give some sort of exponential in these, we first generalise the original Dialectica categories to include fibrations for type theory. In [dP89], Valeria de Paiva explores the Dialectica categories using the subobject fibration, Martin Hyland generalises the definition of a Dialectica category in [Hyl02] to include other preordered fibrations. In this paper we also include fibrations for type theory, that is, fibrations where the fibres are general categories instead of preorders. We will focus on a case study, namely the codomain fibration. The main reason for considering Dialectica categories over general fibrations is that when we start out with more structure, we are forced to be less flexible and the nature of the structures we are studying will reveal themselves. As a spin-off we get a whole new class of Dialectica categories. The analysis shows that both the original Dialectica categories and the Dialectica categories for type theory have a weak exponential, so together with the Cauchy completion we get Cartesian closed Dialectica categories.

Outline of the paper: We start by recalling the definition and closure properties of V. de Paiva's and J.M.E. Hyland's Dialectica categories. We then indicate three different approaches to obtain classes of Cartesian closed Dialectica categories one of which will be studied in this paper. Next we define a generalised version of Dialectica categories and show that they have products. In Section 4 we study monads leading to comonads on Dialectica categories, and we describe the Kleisli category in the general setting. If a comonad is Girardian, we automatically get a Cartesian closed Kleisli category for the comonad. In the most technical part of the paper we show that for a particular non-Girardian comonad, L^+ applied to a Dialectica category Dial(cod($\mathcal{C}$)), gives a Kleisli category, Dial$^+$ with weak exponentials. This result also holds for the original Dialectica categories, Dial(Sub($\mathcal{C}$)) (Dialectica categories over the subobject fibration). This implies that the Cauchy completion of Dial$^+$ is Cartesian closed and also that the preordered reflection of Dial$^+$ is a Heyting algebra. Finally, we spell out the details of an example that might be of particular interest since it corresponds to an extensional version of Dialectica.

2 The Dialectica Categories

In this section we recall the definition of Dialectica categories and their closure properties as given in [dP89] and [Hyl02]. The following is quoted from [Hyl02]: Suppose that we have a category $\mathcal{T}$ which we can think of as interpreting some type theory; and suppose that over the category $\mathcal{T}$ we have a preordered fibration $p : \mathcal{E} \to \mathcal{T}$, which we can regard as providing for each $I \in \mathcal{T}$ a preordered collection of (possibly non-standard) predicates $\mathcal{E}(I) = (\mathcal{E}(I), \vdash)$. Starting with this data we construct a new category Dial(p) which we regard as a category of propositions and proofs.

We do this as follows.

- The objects A of Dial(p) are $U, X \in \mathcal{T}$ together with $\alpha \in \mathcal{E}(U \times X)$. We write this as $A = U \xleftarrow{\alpha} X$. Our understanding of the predicate α is not symmetric as regards U and X: we read $U \xleftarrow{\alpha} X$ as $\exists u.\forall x.\alpha(u,x)$, in accord with the form of propositions in the image of the Dialectica interpretation.

- Maps of Dial(p) from $A = U \xleftarrow{\alpha} X$ to $B = V \xleftarrow{\beta} Y$ are diagrams of the form

$$\begin{tikzcd} U \arrow[dd, "f"'] \arrow[dr, dash] & & X \arrow[ll, "\alpha"', "\shortmid" marking] \\ & {} \arrow[ur, "F"'] \arrow[dr, dash] & \\ V & & Y \arrow[ll, "\beta", "\shortmid" marking] \end{tikzcd}$$

with $\alpha(u, F(u,y)) \vdash \beta(f(u), y)$ in $\mathcal{E}(U \times Y)$.

Thus maps $A \to B$ of Dial(p) consists of maps $f : U \to V$ and $F : U \times Y \to X$ in $\mathcal{T}$ such that $\alpha(u, F(u,y)) \vdash \beta(f(u), y)$ holds in $\mathcal{E}(U \times Y)$.

Proposition 2.1. Dial(p) *forms a category*

The original Dialectica categories described in [dP89] were defined only with p being the subobject fibration. And here comes the closure properties:

Proposition 2.2. *If $p : \mathcal{E} \to \mathcal{T}$ is a product fibration, i.e., $\mathcal{T}$ has finite products and the fibres $\mathcal{E}(I)$ have finite products preserved by reindexing then* $\mathrm{Dial}(p)$ *is a symmetric monoidal category.*

Proposition 2.3. *If $\mathcal{T}$ is ccc and p is fibered Cartesian closed then* $\mathrm{Dial}(p)$ *is symmetric monoidal closed.*

Proposition 2.4. *If $\mathcal{T}$ has finite, distributive coproducts and $\mathcal{E}(0) \cong 1$ and the injections $X \to X + Y$ and $Y \to X + Y$ induce an equivalence $\mathcal{E}(X + Y) \equiv \mathcal{E}(X) \times \mathcal{E}(Y)$ then* $\mathrm{Dial}(p)$ *has finite products.*

2.1 Cartesian Closed Dialectica Categories

We now describe three different approaches to obtain Cartesian closed Dialectica categories. We have seen that the natural structure of the category $\mathrm{Dial}(p)$ is smcc with finite products. One way to obtain Cartesian closure is by adding structure that will make $\otimes$ a product, that is, making sure we get projections and diagonals for $\otimes$. This approach has been studied briefly in [Hyl02]. Another way of obtaining Cartesian closed Dialectica categories is by altering the definition slightly to get variations like the Diller-Nahm Dialectica category (see [dP89]). There are actually several variants constructed in the same manner as the Diller-Nahm category, that is, by a Girardian comonad on the Dialectica category. First steps in this direction has been made in [Bie08], and also implicitly in [Oli08]. The third approach that one might think of is to add enough structure to define an exponent (without making $\otimes = \wedge$). The rest of this paper is devoted to analysing under what circumstances we can get a Cartesian closed Dialectica category using the third approach.

3 Dialectica Categories for Cloven Fibrations

In this section we define a more general version of the Dialectica categories and show that they have finite products.

Given a cloven fibration $p : \mathcal{E} \to \mathcal{T}$, if $\mathcal{T}$ has binary products, we can construct the Dialectica category for types over p, written $\mathrm{Dial}(p)$ as follows:

- Objects are triples (U, X, α), where $U, X \in \mathcal{T}$ and $\alpha \in \mathcal{E}$ is an object in the fibre over $U \times X$.
- A map from (U, X, α) to (V, Y, β) is a triple (f, F, φ), where $f : U \to V$, $F : U \times Y \to X$ and $\varphi(u, y) : \alpha(u, F(u, y)) \to \beta(f(u), y)$ in the fibre over $U \times Y$.

We can think of $\mathcal{T}$ as our types and $\alpha \in \mathcal{E}(U \times X)$ as a dependent type over $U \times X$. Maps are written

$$\begin{tikzcd}
U \arrow[d, "f"'] \arrow[dr, dash, ""{name=M}] & X \arrow[l, "\alpha"', "\shortmid"{marking}] \\
V & Y \arrow[l, "\beta", "\shortmid"{marking}]
\arrow[from=M, to=1-2, "F"']
\end{tikzcd}
\qquad \phi(u, y) : \alpha(u, F(u, y)) \to \beta(fu, y)$$

This forms a category, with

- identity arrows: $(\mathrm{id}_U, \pi_2, \mathrm{id}_{\alpha(u,x)})$.
- Composition works as follows: Given $(f, F, \phi(u,y)) : (U, X, \alpha) \to (V, Y, \beta)$ and $(g, G, \psi(v, w)) : (V, Y, \beta) \to (Z, W, \gamma)$, the composite is
$$(gf, F(u, G(f(u), w)), \psi(f(u), w) \circ \phi(u, G(f(u), w))),$$
where $F(u, G(f(u), w))$ is the arrow $U \times W \xrightarrow{\langle U, f\times W\rangle} U \times (V \times W) \xrightarrow{U\times G} U \times Y \xrightarrow{F} X$, and $\phi(u, G(f(u), w))$ is reindexing of $\phi(u, y)$ along the arrow $(U \times G) \circ \langle U, f \times W\rangle$.

Since reindexing in a fibration is up to isomorphism, we need the fibration to be cloven in order for composition to be associative. Associativity is then a a consequence of the coherence conditions for a cloven fibration. Note that this only regards the third component of the arrows in Dial(p).

3.1 Products in Dial(p)

Definition 3.1. *A category $\mathcal{C}$ has finite, distributive coproducts if it has finite coproducts and products, and the product functor preserves coproducts, that is, for objects A, X, Y there is a natural isomorphism $\delta : A \times (X + Y) \cong A \times X + A \times Y$ such that the following diagram commutes:*

$$\begin{tikzcd}
A \times X \arrow[r, "A\times\iota_X"] \arrow[dr, "\iota_{A\times X}"'] & A \times (X+Y) \arrow[d, "\delta"] & A \times Y \arrow[l, "A\times\iota_Y"'] \arrow[dl, "\iota_{A\times Y}"] \\
 & A \times X + A \times Y &
\end{tikzcd}$$

Fact: In a distributive category, 0 is a strict initial object.

Of course, any Cartesian closed category with finite coproducts is distributive.

Let $q : \mathcal{E} \to \mathcal{T}$ be a preordered fibration and let $\mathcal{E}(X)$ denote the fibre over X. From [Hyl02] we know that, if $\mathcal{T}$ has finite, distributive coproducts, and it also holds that $\mathcal{E}(\mathbf{0}) \cong \mathbf{1}_{\mathcal{E}}$ and that the injections $X \to X+Y$ and $Y \to X+Y$ induce an equivalence $\mathcal{E}(X+Y) \equiv \mathcal{E}(X)\times\mathcal{E}(Y)$, then Dial($q$) has finite products. We now show that this also holds in the general case, where the fibres $\mathcal{E}(X)$ are not preorders, but categories.

Proposition 3.2. *Let $p : \mathcal{E} \to \mathcal{T}$ be a cloven fibration.*

1. *Suppose $\mathcal{T}$ has finite, distributive coproducts and products, and that the injections $X \to X + Y$ and $Y \to X + Y$ induce an equivalence $\mu : \mathcal{E}(X) \times \mathcal{E}(Y) \equiv \mathcal{E}(X + Y)$, natural in X, Y, then* Dial(p) *has binary products.*
2. *Suppose that $\mathcal{E}(0) \cong \mathbf{1}$, then* Dial($p$) *has a terminal object.*

Proof: This is a straightforward generalization of the proof found in [Hyl02]. First note that since the equivalence μ is induced by the injections, we have $\mu^{-1} = \langle \iota_X^*, \iota_Y^*\rangle$ and so

$$\iota_X^* \mu(\phi, \psi) \cong \phi \tag{1}$$

for $\phi \in \mathcal{E}(X)$ and $\psi \in \mathcal{E}(Y)$. The product $A \times B$ of $A = (\, U \xleftarrow{\alpha} X \,)$ and $B = (\, V \xleftarrow{\beta} Y \,)$ is

$$A \times B = (\, U \times V \xleftarrow{\alpha\&\beta} X + Y \,)$$

where $\alpha\&\beta \in \mathcal{E}(U \times V \times (X+Y)) \overset{\mu^{-1}\circ(\delta^{-1})^*}{\cong} \mathcal{E}(U \times V \times X) \times \mathcal{E}(U \times V \times Y)$ is given by $\delta^*\mu(\alpha(\pi_U(u,v),x),\beta(\pi_V(u,v),y))$. The projections are $(\pi_U, \kappa_X \circ \pi_X, \mathrm{id}_{\alpha(\pi_U(u,v),x)})$ and $(\pi_V, \kappa_Y \circ \pi_Y, \mathrm{id}_{\beta(\pi_V(u,v),y)})$.

Let C be the object $(Z \xleftarrow{\gamma} W)$. Given morphisms

$$(f,F,\phi(z,x)) : C \to A \text{ and}$$
$$(g,G,\psi(z,y)) : C \to B,$$

the universal map from C to $A \times B$ is

$$(\langle f,g\rangle, [F,G], \delta^*\mu(\phi(z,x),\psi(z,y))),$$

where $\delta^*\mu(\phi(z,x),\psi(z,y))) \in \mathcal{E}(Z \times (X+Y))$ is reindexing of $\mu(\phi(z,x),\psi(z,y))) \in \mathcal{E}(Z \times X + Z \times Y)$ along the isomorphism $\delta : Z \times (X+Y) \to Z \times X + Z \times Y$.

The composite of

$$(\langle f,g\rangle, [F,G], \delta^*\mu(\phi(z,x),\psi(z,y)))$$

with the projection

$$(\pi_U, \iota_X \circ \pi_X, \mathrm{id}_{\alpha(\pi_U(u,v),x)})$$

is the arrow

$$(\pi_u \circ \langle f,g\rangle, [F,G](z, \iota_X\pi_X(\langle f,g\rangle z,x)), \mathrm{id}_{\alpha(\pi_U(u,v),x)}(\langle f,g\rangle z,x) \circ (\langle z, \iota_X\pi_X(\langle f,g\rangle z,x)\rangle)^*(\delta^*\mu(\phi,\psi))).$$

Strictly speaking, we must also compose $\delta^*\mu(\phi,\psi)$ with the appropriate coherence maps and the (unique) iso which is part of the equivalence μ, but the notation is already heavy, so we leave those implicit.

Now $\langle z, \iota_X\pi_X(\langle f,g\rangle z,x)\rangle = Z \times \iota_X(x) : Z \times X \to Z \times (X+Y)$; by (1), and again keeping the coherence maps that are part of the composition implicit, we get,

$$(Z \times \iota_X)^*(\delta^*\mu(\phi,\psi)) = \iota^*_{Z\times X}(\mu(\phi,\psi)) = \phi,$$

moreover

$$\mathrm{id}_{\alpha(\pi_U(u,v),x)}(\langle f,g\rangle z,x) = \mathrm{id}_{\alpha(\pi_U(fz,gz),x)} = \mathrm{id}_{\alpha(fz,x)},$$

since reindexing is functorial, so the composite is (f,F,ϕ) as needed. Uniqueness is clear by inspection.

The terminal object of Dial(p) is $(0 \xleftarrow{!} 1)$, where ! is the unique object of $\mathcal{E}(0)$. □

The product functor works as follows: Given

$$(f,F,\phi(u,x')) : \quad A = (U \xleftarrow{\alpha} X) \to A' = (U' \xleftarrow{\alpha'} X')$$
$$(g,G,\psi(v,y')) : \quad B = (V \xleftarrow{\beta} Y) \to B' = (V' \xleftarrow{\beta'} Y')$$

The product $(f,F,\phi(u,x')) \times (g,G,\psi(v,y')) : A \times B \to A' \times B'$ is

$$(f \times g, F(\pi_U(u,v),x') + G(\pi_V(u,v),y'), \delta^*\mu[\phi(\pi_U(u,v),x'),\psi(\pi_V(u,v),y')]).$$

Example 3.3. *Examples of fibrations satisfying Proposition 3.2 (which are actually equivalent to codomain fibrations) are the split fibrations* Fam(Set) $\to$ Set *For set-indexed families of sets we have* $\mu((A_i)_{i\in I},(B_i)_{i\in I}) = (C_z)_{z\in X+Y}$*, where*

$$C_z = \begin{cases} A_x & \text{if } z=(0,x) \\ B_y & \text{if } z=(1,y) \end{cases}$$

and UFam(PER) $\to$ PER *(for a description of this fibration, see Section 5). For per-indexed families of pers,* $\mu((A_{[n]})_{[n]\in\mathbb{N}/R},(B_{[m]})_{[m]\in\mathbb{N}/S}) = (C_{[k]})_{[k]\in N/R+S}$*, where*

$$(C_{[k]})_{[k]\in N/R+S} = \begin{cases} A_{[n]} & \text{if } pk=0 \text{ and } [p'k]=[n] \\ B_{[m]} & \text{if } pk=1 \text{ and } [p'k]=[m] \end{cases}$$

We now show under which conditions our case study, the codomain fibration, satisfies the assumptions of Proposition 3.2

Definition 3.4. *A category is said to have* stable *coproducts, if for* $A=\coprod_{i\in I}A_i$ *and any map* $f: B\to A$, $\coprod_{i\in I} f^{-1}(A_i)\cong B$*, where* $f^{-1}(A_i)$ *is the pullback of the injection* $A_i\to\coprod_{i\in I}A_i$ *along* f.

Definition 3.5. *A category is said to have* disjoint *coproducts if each of the injections* $\iota_i : A_i\to A$ *is mono and for each pair of distinct injections* ι_i,ι_j*, the pullback of the two is the initial object.*

The following fact is well known (see, e.g., [CLW93])

Proposition 3.6. *A category that has finite limits and finite, stable, disjoint coproducts is distributive.*

Now let $\mathcal{C}$ be a category with finite products and coproducts, and assume that the coproducts are stable and disjoint, then cod($\mathcal{C}$) is a cloven fibration with an equivalence $\mu:\mathcal{C}/X\times\mathcal{C}/Y\to\mathcal{C}/(X+Y)$ given by $\mu(\alpha,\beta)=\alpha+\beta$ and $\mu^{-1}(\gamma)=(\iota_X^*(\gamma),\iota_Y^*(\gamma))$. This means that Dial(cod($\mathcal{C}$)) has products.

We sum up the results in the following proposition:

Proposition 3.7. *Let* $\mathcal{C}$ *be a category with finite limits and finite coproducts, and assume that the coproducts are stable and disjoint, then* Dial(cod($\mathcal{C}$)) *has finite products.*

Martin Hyland came up with the following construction, which we shall see gives us a variant of a Dialectica category with a weak exponent.

Let $\mathcal{C}$ be a category with finite products and stable, disjoint coproducts. The functor $-+1:\mathcal{C}\to\mathcal{C}$ together with families of maps $\iota: X\to X+1$ and $X+(1+1)\to X+1$ is a monad on $\mathcal{C}$. In the same way that the monad defined from the free commutative monoid monad gives us a comonad $!$: Dial(Sub($\mathcal{C}$)) $\to$ Dial(Sub($\mathcal{C}$)) (see [dP89]), we can define a comonad L^+ on Dial(Sub($\mathcal{C}$)) using the monad $-+1$ as follows. Let α be a subobject of $U\times X$ in $\mathcal{C}$, then

$$L^+(\alpha\rightarrowtail U\times X)=\hat{\alpha}\rightarrowtail U\times(X+1)$$

where $\hat{\alpha}$ is reindexing of α along the arrow

$$U\times X+!_U : U\times(X+1)\cong U\times X+U\to(U\times X)+1.$$

Using the internal language, this means that

$$\hat{\alpha}(u,x) = \begin{cases} \alpha(u,x) & \text{if} \quad x \in X \\ \top & \text{if} \quad x \in 1. \end{cases}$$

And

$$L^+(f,F) = (f,\hat{F}),$$

where $\hat{F}$ is the composite

$$\begin{tikzcd} U \times (Y+1) \arrow[r, "(U\times Y)+!_U"] & (U \times Y + 1) \arrow[r, "F+1"] & X+1 \end{tikzcd}$$

In the internal language this becomes

$$\hat{F}(u,y) = \begin{cases} F(u,y) & \text{if} \quad y \in Y \\ * \in 1 & \text{if} \quad y \in 1. \end{cases}$$

Now, it is the Kleisli category Dial_{L^+} for this comonad that we are really interested in.

In the case of the free commutative monoid monad, there is an isomorphism

$$X^* \times Y^* \cong (X+Y)^* \tag{2}$$

which induces an isomorphism

$$!(A \times B) \cong !A \otimes !B \tag{3}$$

in Dial(Sub($\mathcal{C}$)). A comonad L satisfying $L(A \times B) \cong LA \otimes LB$ will be called *Girardian.* If a comonad is Girardian, the isomorphism in 3 gives us a Cartesian closed structure on the Kleisli category by the following string of equivalences:

$$\begin{aligned} &\mathrm{Hom}_{\mathrm{Dial}_!}(A \times B, C) = \mathrm{Hom}_{\mathrm{Dial}}(!(A \times B), C) \cong \mathrm{Hom}_{\mathrm{Dial}}(!A \otimes !B, C) \\ &\cong \mathrm{Hom}_{\mathrm{Dial}}(!A, [!B, C]_{\mathrm{Dial}}) = \mathrm{Hom}_{\mathrm{Dial}_!}(A, [!B, C]_{\mathrm{Dial}}) = \mathrm{Hom}_{\mathrm{Dial}_!}(A, [B, C]_{\mathrm{Dial}_!}) \end{aligned} \tag{4}$$

Now, for the monad $-+1$ we do *not* have such an isomorphism, because

$$(X+1) \times (Y+1) \cong X+Y+1+(X \times Y) \neq X+Y+1$$

So the monad $-+1$ does not satisfy the distributive law in 2, and one readily sees that the comonad L^+ does not satisfy the distributive law in 3. However, we shall see that what we *do* have is a natural retraction

$$\mathrm{Hom}_{\mathrm{Dial}}(L^+(A \times B), C) \to \mathrm{Hom}_{\mathrm{Dial}}(L^+(A), B \supset C),$$

so $B \supset C$ is the weak exponent, that we will define in the next section. Notice that $B \supset C$ is *not* simply $[LB, C]_{\mathrm{Dial}}$. Hence for the Kleisli category Dial_{L^+}, we will have a natural retraction

$$\mathrm{Hom}_{\mathrm{Dial}_{L^+}}(A \times B, C) \to \mathrm{Hom}_{\mathrm{Dial}_{L^+}}(A, B \supset C).$$

And then the Cauchy completion (see Appendix A) will give a Cartesian closed category.

Definition 3.8 (Weak exponential)**.** *Let $\mathcal{C}$ be a category with finite products. $\mathcal{C}$ has* weak exponentials $[B,C]$, *if there is a retraction*

$$\begin{tikzcd} \mathcal{C}(A \times B, C) \arrow[r, shift left, "I"] & \mathcal{C}(A, [B,C]) \arrow[l, shift left, "R"] \end{tikzcd}$$

onto $\mathcal{C}(A \times B, C)$ (that is, $RI = \mathrm{id}$), natural in A.

4 A non-Girardian comonad on Dial(p)

In this section we study which conditions are needed on the monad and the fibration to give a well-defined comonad on Dial(p).

Recall the definition of a strong monad on a category with finite products.

Definition 4.1. *Let $\mathcal{C}$ be a category with finite products. A* strong monad *on $\mathcal{C}$ is a monad (T, η, μ) together with a natural transformation*

$$C_{X,Y} : X \times TY \to T(X \times Y)$$

called strength*, such that the diagrams*

$$\begin{tikzcd}
X \times Y \arrow[r, "X\times\eta_Y"] \arrow[dr, "\eta_{X\times Y}"'] & X \times TY \arrow[d, "C_{X,Y},"] \\
 & T(X \times Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
X \times T^2Y \arrow[r, "C_{X,TY}"] \arrow[d, "X\times\mu_Y"'] & T(X \times TY) \arrow[r, "T(C_{X,Y})"] & T^2(X \times Y) \arrow[d, "\mu_{X\times Y}"] \\
X \times TY \arrow[rr, "C_{X,Y}"'] & & T(X \times Y)
\end{tikzcd}$$

$$\begin{tikzcd}
(X \times Y) \times TZ \arrow[rr, "C_{X\times Y,Z}"] \arrow[d, "\cong"'] & & T((X \times Y) \times Z) \arrow[d, "\cong"] \\
X \times (Y \times TZ) \arrow[r, "X\times C_{Y,Z}"'] & X \times T(Y \times Z) \arrow[r, "C_{X,Y\times Z}"'] & T(X \times (Y \times Z))
\end{tikzcd}$$

commute for all objects X, Y and Z.

We aim to use a monad to define a comonad on Dial(p) for p a cloven fibration, for that we require the monad to be fibred. A fibred monad (T, T') on a fibration $p : \mathcal{E} \to \mathcal{T}$ is a morphism of fibrations (T,T'):

$$\begin{tikzcd}
\mathcal{E} \arrow[r, "T"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{T} \arrow[r, "T'"'] & \mathcal{T}
\end{tikzcd}$$

together with 2-cells $\mu = (\mu, \mu')$ and $\eta = (\eta, \eta')$:

$$\begin{tikzcd}
\mathcal{E} \arrow[r, bend left, "\mathrm{Id}"] \arrow[r, bend right, "T"'] \arrow[r, phantom, "\Downarrow\eta"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{T} \arrow[r, bend left, "\mathrm{Id}"] \arrow[r, bend right, "T'"'] \arrow[r, phantom, "\Downarrow\eta'"] & \mathcal{T}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{E} \arrow[r, bend left, "T^2"] \arrow[r, bend right, "T"'] \arrow[r, phantom, "\Downarrow\mu"] \arrow[d, "p"'] & \mathcal{E} \arrow[d, "p"] \\
\mathcal{T} \arrow[r, bend left, "(T')^2"] \arrow[r, bend right, "T'"'] \arrow[r, phantom, "\Downarrow\mu'"] & \mathcal{T}
\end{tikzcd}$$

where η is above η', that is, for $X \in \mathcal{E}$, $p(\eta_X) = \eta'_{pX}$, and similar for μ. (T, T') commutes with reindexing in the sense that for $u : X \to Y$ in $\mathcal{T}$, we have $T \circ u^* \cong (T'u)^* \circ T$ as fibred functors from $\mathcal{E}(Y) \to \mathcal{E}(T'X)$. For more details, see e.g. [Jac99].

Suppose we have a fibred monad (T, T'), where T' is strong, then we are able to define a comonad L on Dial(p) by

- $L(U, X, \alpha) = (U, T'X, C^*_{U,X}(T\alpha) = \hat{\alpha})$ and
- $L(f, F, \phi) = (f, T'(F) \circ C_{U,Y}, C^*_{U,Y}(T\phi)) = (f, \hat{F}, \hat{\phi})$.

For the arrow part of the functor L to be well-defined we must show that $C^*_{U,Y}T$ commutes with reindexing, that is,

$$C^*_{U,Y}T(\alpha(u, F(u, y))) = (C^*_{U,Y}T(\alpha))(u, \hat{F}(u, y)) = \hat{\alpha}(u, \hat{F}(u, y)). \tag{5}$$

and

$$C^*_{U,Y}T(\beta(fu, y)) = (C^*_{V,Y}T(\beta))(fu, y) = \hat{\beta}(fu, y). \tag{6}$$

Equation (6) holds because (T, T') is a morphism of fibrations and therefore commutes with reindexing, and because of naturality of C:

$$\begin{aligned} C^*_{U,Y}T(\beta(fu, y)) &= C^*_{U,Y}T((f \times Y)^*\beta) \\ &\cong C^*_{U,Y}(T'(f \times Y))^*(T\beta) \\ &= (f \times T'Y)^* C^*_{V,Y}(T\beta) \\ &= \hat{\beta}(fu, y). \end{aligned}$$

To see that (5) holds, consider

$$\begin{aligned} C^*_{U,Y}T(\alpha(u, F(u, y))) &= C^*_{U,Y}T(\langle \pi_U, F\rangle^*(\alpha)) \\ &\cong C^*_{U,Y}(T'(\langle \pi_U, F\rangle)^*(T(\alpha))) && \text{T commutes with reindexing} \\ &= (T'(\langle \pi_U, F\rangle) \circ C_{U,Y})^*(T\alpha) \\ &= (C_{U,X} \circ \langle \pi_U, T'(F) \circ C_{U,Y}\rangle)^*(T\alpha) && \text{see diagram below} \\ &= \hat{\alpha}(u, \hat{F}(u, y)) \end{aligned}$$

We must show that the following diagram commutes in $\mathcal{T}$:

$$\begin{tikzcd} U \times T'Y \arrow[r, "C_{U,Y}"] \arrow[d, "\langle \pi_U, T'F \circ C_{U,Y}\rangle"'] & T'(U \times Y) \arrow[d, "T'\langle \pi_U, F\rangle"] \\ U \times T'X \arrow[r, "C_{U,X}"'] & T'(U \times X) \end{tikzcd} \tag{7}$$

The diagram (7) can be decomposed as:

$$\begin{tikzcd} U \times T'Y \arrow[r, "C_{U,Y}"] \arrow[d, "\delta_U \times T'Y"'] \arrow[dr, phantom, "(1)"] & T'(U \times Y) \arrow[d, "T'(\delta_U \times Y)"] \\ U \times U \times T'Y \arrow[r, "C_{U \times U,Y}"] \arrow[d, "U \times C_{U,Y}"'] \arrow[dr, phantom, "(2)"] & T'(U \times U \times Y) \arrow[d, "\mathrm{id}"] \\ U \times T'(U \times Y) \arrow[r, "C_{U,U \times Y}"] \arrow[d, "U \times T'F"'] \arrow[dr, phantom, "(3)"] & T'(U \times U \times Y) \arrow[d, "T'(U \times F)"] \\ U \times T'X \arrow[r, "C_{U,X}"'] & T'(U \times X) \end{tikzcd}$$

where (1) and (3) commute by naturality of C, and (2) commutes by properties of strength of T'.

Assume the following extra requirements on (T, η, μ):

$$\begin{array}{rcl} \alpha(u,x) & = & \eta^*_{U\times X}(T\alpha) \quad \text{and} \\ T^2(\alpha) & = & \mu^*_{U\times X}(T\alpha) \end{array} \tag{8}$$

Using the properties of strength one can show that the equations in (8) imply the following equations:

$$\begin{array}{rcl} \hat{\alpha}(u,\eta_X(x)) & = & \alpha(u,x) \quad \text{and} \\ \hat{\alpha}(u,\mu_X(x')) & = & \hat{\hat{\alpha}}(u,x'). \end{array} \tag{9}$$

We now use these assumptions to show that L is a comonad:

L is a comonad on Dial(p)**:** For every (U, X, α) we have a map

$$\begin{tikzcd} U & & TX \arrow[ll, "\hat{\alpha}"', "|" marking] \\ & {} \arrow[ur, "\eta_X\pi_X"'] & \\ U \arrow[from=1-1, d, "\text{id}"] & & X \arrow[ll, "\alpha", "|" marking] \arrow[from=1-1, dash] \end{tikzcd}$$

with

$$\hat{\alpha}(u,\eta(x)) = \alpha(u,x) \xrightarrow{\text{id}} \alpha(u,x).$$

since from (9) we have $\hat{\alpha}(u,\eta(x)) = \alpha(u,x)$. We define

$$\varepsilon_{(U,X,\alpha)} = (\text{id}_U, \eta_X\pi_X, \text{id}_\alpha).$$

And for every (U, X, α) we have a map

$$\begin{tikzcd} U & & TX \arrow[ll, "\hat{\alpha}"', "|" marking] \\ & {} \arrow[ur, "\mu_X\pi"'] & \\ U \arrow[from=1-1, d, "\text{id}"] & & X \arrow[ll, "\hat{\hat{\alpha}}", "|" marking] \arrow[from=1-1, dash] \end{tikzcd}$$

with

$$\hat{\alpha}(u,\mu_X(x')) = \hat{\hat{\alpha}}(u,x') \xrightarrow{\text{id}} \hat{\hat{\alpha}}(u,x'),$$

where $x' : TT(X)$. Again we are using (9) to get $\hat{\alpha}(u,\mu_X(x')) = \hat{\hat{\alpha}}(u,x')$. We define

$$\delta_{(U,X,\alpha)} = (\text{id}_U, \mu_X\pi, \text{id}_{\hat{\hat{\alpha}}}).$$

We have shown the following.

Proposition 4.2. *Let $((T, T'), (\eta, \eta'), (\mu, \mu'))$ be a fibred monad with T' a strong monad on a cloven fibration $p : \mathcal{E} \to \mathcal{T}$, and with $\mathcal{T}$ ccc. If (T, η, μ) also satisfies the equations in (8), we can define a comonad L on* Dial(p) *by*

$$L(U, X, \alpha) = (U, TX, C^*_{U,X}(T\alpha) = \hat{\alpha}) \quad \textit{and} \quad L(f, F, \phi) = (f, TF \circ C_{U,Y}, C^*_{U,Y}(T(\phi))) = (f, \hat{F}, \hat{\phi}).$$

The Comonad L^+ Our leading example is the comonad L^+ on Dial(cod($\mathcal{C}$)) based on the monad $TX = X + 1$, which is strong and induces a fibred monad on cod($\mathcal{C}$). We have

Lemma 4.3. *For a Cartesian closed category $\mathcal{C}$ with finite coproducts, the functor $TX = X+1$ together with the obvious natural transformations $\mu_X : X+1+1 \to X+1$ and $\eta_X : X \to X+1$ is a strong monad. The maps $C_{X,Y}$ are defined by*

$$X \times (Y+1) \cong X \times Y + X \xrightarrow{X \times Y + !} X \times Y + 1$$

Assuming $\mathcal{C}$ has stable, disjoint coproducts, it is not hard to see that the equations in (8) are met by the monad $-+1$. We collect the facts:

Proposition 4.4. *Suppose $\mathcal{C}$ has finite limits and coproducts, and that coproducts are stable and disjoint. Then the monad $-+1$ gives rise to a comonad on* Dial(cod($\mathcal{C}$)).

Some examples:

Example 4.5. *Examples that satisfy Proposition 4.2 are the codomain fibration together with the monads $-+1$ and strings; and the subobject fibration together with the monads multisets, powersets, finite powersets and the free commutative monoid monad. The Kleisli category for the latter is the Diller-Nahm Dialectica category. The comonad based on $-+1$ is the only non-Girardian among these examples.*

4.1 The Kleisli Category Dial$_L(p)$

We now write out some details about the category Dial$_L(p)$ for a comonad L on Dial(p).

Dial$_L$ has products inherited from Dial. For the record:

Proposition 4.6. *The Kleisli category* Dial$_L(p)$ *has products inherited from* Dial(p).

Proof:

$$\begin{aligned} \mathrm{Dial}_L(\alpha,\beta) \times \mathrm{Dial}_L(\alpha,\gamma) &= \mathrm{Dial}(\hat{\alpha},\beta) \times \mathrm{Dial}(\hat{\alpha},\gamma) \\ &\cong \mathrm{Dial}(\hat{\alpha},\beta \times \gamma) \\ &= \mathrm{Dial}_L(\alpha,\beta \times \gamma). \end{aligned}$$

□

Composition in the Kleisli category Dial$_L(p)$ Given two maps $(f, F, \phi(u,y)) : (U, TX, \hat{\alpha}) \to (V, Y, \beta)$ and $(g, G, \psi(v,w)) : (V, TY, \hat{\beta}) \to (Z, W, \gamma)$ the composite is

$$\begin{tikzcd} U \arrow[d, "gf"'] \arrow[rd, dash] & TX \arrow[l, "\hat{\alpha}"', "|" marking] \\ Z & W \arrow[l, "\gamma", "|" marking] \arrow[u, "\mu_X \circ \hat{F}(u,G(fu,w))"'] \end{tikzcd}$$

with

$$\hat{\alpha}(u, \mu_X \circ \hat{F}(u, G(fu,w))) = \hat{\hat{\alpha}}(u, \hat{F}(u, G(fu,w))) \xrightarrow{\hat{\phi}(u,G(fu,w))} \hat{\beta}(fu, G(fu,w)) \xrightarrow{\psi(fu,w)} \gamma(gfu, w).$$

Product functor in the Kleisli category $\mathrm{Dial}_L(p)$ Given maps

$$\begin{tikzcd}
U' \arrow[d, "f"'] \arrow[dr, dash] & TX' \arrow[l, "\hat{\alpha}'"', "+" marking] \\
U & X \arrow[l, "\alpha", "+" marking] \arrow[u, "F"']
\end{tikzcd}$$

$$\phi(u', x) : \hat{\alpha}'(u', F(u', x)) \longrightarrow \alpha(fu', x)$$

and

$$\begin{tikzcd}
V' \arrow[d, "g"'] \arrow[dr, dash] & TY' \arrow[l, "\hat{\beta}'"', "+" marking] \\
V & Y \arrow[l, "\beta", "+" marking] \arrow[u, "G"']
\end{tikzcd}$$

$$\psi(v', y) : \hat{\beta}'(v', F(v', y)) \longrightarrow \alpha(gv', y)$$

The product is

$$\begin{tikzcd}
U' \times V' \arrow[d, "f \times g"'] \arrow[dr, dash] & T(X' + Y') \arrow[l, "\alpha' \hat{\&} \beta'"', "+" marking] \\
U \times V & X + Y \arrow[l, "\alpha \& \beta", "+" marking] \arrow[u, "\mu_{X+Y} \circ (F\pi + G\pi)"']
\end{tikzcd}$$

$$\phi(\pi(u', v'), x) + \psi(\pi'(u', v'), y)$$

For a Girardian comonad, the Kleisli category is automatically Cartesian closed (see (4)). We now show that for the non-Girardian comonad L^+ constructed from $- + 1$, we can define a weak exponent in the Kleisli category. Notice that the weak exponent is *not* simply the usual $[L^+A, B]_{\mathrm{Dial}}$.

4.2 The Kleisli category Dial$^+$ has a weak exponent.

From this point we only consider the comonad L^+ which was defined on the basis of the monad $- + 1$. Let $\mathcal{C}$ be a ccc with stable, disjoint coproducts, and which is locally Cartesian closed. We have already seen that the monad $- + 1$ satisfies the requirements needed to construct a comonad L^+ on Dial(cod($\mathcal{C}$)). Let Dial$^+$ denote the Kleisli category for L^+ on Dial(cod($\mathcal{C}$)). We are going to show that Dial$^+$ has weak exponentials. That is,

Theorem 4.7. *Let $\mathcal{C}$ be a ccc with finite limits, and stable, disjoint coproducts, and which is locally Cartesian closed, then the Dialectica-Kleisli category,* $\mathrm{Dial}_{L^+}(\mathrm{cod}(\mathcal{C}))$*, which we denote by* Dial$^+$*, and also* $\mathrm{Dial}_{L^+}(\mathrm{Sub}(\mathcal{C}))$ *has finite products and weak exponentials.*

First some notation that we shall be using.

Notation Let $\mathcal{C}$ be a ccc with stable, disjoint coproducts and consider a pullback of the form:

$$\begin{tikzcd} F^{-1}(\alpha+\beta) \arrow[r] \arrow[d] & \alpha+\beta \arrow[d, "a+b"] \\ U\times V \arrow[r, "F"'] & X+Y \end{tikzcd}$$

Since we have stable, disjoint coproducts, we have

$$\begin{tikzcd} U\times V \cong F^{-1}(X)+F^{-1}(Y) \arrow[d, "F=F_X+F_Y"] \\ X+Y \end{tikzcd}$$

and pullback preserves coproducts, so

$$F^{-1}(\alpha+\beta)\cong F^{-1}(\alpha)+F^{-1}(\beta)$$

which, because of the stable, disjoint coproducts is the same as

$$F_X^{-1}(\alpha)+F_Y^{-1}(\beta)$$

Note that this holds in any fibration over $\mathcal{C}$ where reindexing preserves coproducts.

We will use the following notation for $F_X^{-1}(\alpha)+F_Y^{-1}(\beta)$ in this situation:

$$F(u,v)\in X.\,\alpha(F(u,v))+F(u,v)\in Y.\,\beta(F(u,v))$$

or sometimes, when convenient:

$$F(u,v)=x\in X.\,\alpha(x)+F(u,v)=y\in Y.\,\beta(y)$$

indicating that α is being reindexed along those (u,v) such that $F(u,v)\in X$ and β along those (u,v) such that $F(u,v)\in Y$.

In case α (or β) is the terminal object of the fibre, in this case if $\alpha=X$, $a=\mathrm{id}$, the pullback will be the type

$$\begin{aligned} F(u,v)\in X.\,\top_X(F(u,v))+F(u,v)\in Y.\,\beta(F(u,v)) \quad &= \\ F(u,v)\in X.\,\top_{F^{-1}(X)}+F(u,v)\in Y.\,\beta(F(u,v)) & \end{aligned}$$

By abuse of notation we will sometimes leave out the first part and just write this type as

$$F(u,v)\in Y.\,\beta(F(u,v)).$$

We are now able to give the proof of the Theorem 4.7:

Proof. First we define an object corresponding to

$$\begin{aligned} \Box = \{&(g,G):(V\Rightarrow W)\times(V\times Z\Rightarrow 1+Y),(v,z):V\times Z, \\ &k(g,G,v,z):G(v,z)=y\in Y.[\beta(v,G(v,z)),\gamma(gv,z)]\} \end{aligned}$$

where $[\beta,\gamma]$ is the fibred exponential, and $G(v,z)=y\in Y.[\beta(v,G(v,z)),\gamma(gv,z)]$ means (informally) that we reindex the fibred exponent $[\beta,\gamma]$ along those (G,g,v,z) such that

$G(v,z) \in Y$, which makes sense since we have disjoint, stable coproducts. In our case study, the codomain fibration, this is the dependent type defined by the pullback:

$$\begin{tikzcd}
& \square \arrow[r] \arrow[d] & W \times V \times Z + [W \times Z \times \beta, V \times Y \times \gamma] \arrow[d, "\mathrm{id}+b"] \\
& (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y) \times V \times Z \arrow[r, "\langle ev, V\times Z\rangle \circ ((\pi \Rightarrow W)\times \mathrm{id})"'] & W \times V \times Z + W \times V \times Z \times Y
\end{tikzcd}$$

where $(\pi \Rightarrow W) \times \mathrm{id} : (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y) \times V \times Z \to (V \times Z \Rightarrow W) \times (V \times Z \Rightarrow 1+Y) \times V \times Z$. Since B is lcc we have a right adjoint to reindexing $\pi^* \dashv \Pi_\pi$. Let $\pi : (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y) \times V \times Z \to (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y)$ then we have

$$\begin{tikzcd}
(\Pi_\pi \square) \times V \times Z \arrow[r, "\varepsilon"] \arrow[dr] & \square \arrow[d] \\
& (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y) \times V \times Z
\end{tikzcd}$$

where ε is the counit for the adjunction $\pi^* \dashv \Pi_\pi$. We have

$$\Pi_\pi \square = \{(g,G) : (V \Rightarrow W) \times (V \times Z \Rightarrow 1+Y), \lambda v,z.K(g,G,v,z) : \Pi v,z.G(v,z) = y \in Y.[\beta(v,G(v,z)), \gamma(g(v),z)]\}$$

Now the exponent in Dial$^+$, which we will denote $\beta \supset \gamma$, is the dependent type over $(\Pi_\pi \square) \times V \times Z$ corresponding to

$$(\beta \supset \gamma)((g,G,k),(v,z)) := G(v,z) \in 1.\gamma(g(v),z)$$

defined by the pullback:

$$\begin{tikzcd}
\beta \supset \gamma \arrow[rr] \arrow[d] & & (\gamma \times V) + [W \times Z \times \beta, V \times Y \times \gamma] \arrow[d, "c+\mathrm{id}"] \\
(\Pi_\pi \square) \times V \times Z \arrow[r, "\varepsilon"'] & \square \arrow[r] & W \times V \times Z + [W \times Z \times \beta, V \times Y \times \gamma]
\end{tikzcd}$$

In the type theoretic language this is

$$\beta \supset \gamma = \{(g, G, \lambda v,z.k(g,G,v,z)) : \Pi_\pi \square, v,z \in V \times Z, h(g,G,\lambda v,z.k,v,z) : G(v,z) \in 1.\gamma(g(v),z)\}$$

Before we show that there is a natural retraction

$$\begin{tikzcd}
\mathrm{Dial}^+(\alpha \times \beta, \gamma) \arrow[r, shift left, "I"] & \mathrm{Dial}^+(\alpha, \beta \supset \gamma), \arrow[l, shift left, "R"]
\end{tikzcd}$$

we will characterize the homsets. A map from $\alpha \& \beta$ to γ in Dial$^+$:

$$\begin{tikzcd}
U \times V \arrow[d, "f"'] \arrow[drr, dash] & & X+Y+1 \arrow[ll, "\alpha\hat{\&}\beta"', "|" marking] \\
W & & Z \arrow[ll, "\gamma", "|" marking] \arrow[u, "F"']
\end{tikzcd}$$

$$\phi(u,v,z) : \alpha\hat{\&}\beta(u,v,F(u,v,z)) \to \gamma(f(u,v),z)$$

Now because we have stable and disjoint coproducts, we can write $\phi(u, v, z)$ as

$$\begin{array}{lllll}
\phi(u,v,z) = & & & & \\
 & \phi_X(u,v,z): & F(u,v,z) = x \in X. & \alpha(u,x) \to \gamma(f(u,v),z) \\
+ & \phi_Y(u,v,z): & F(u,v,z) = y \in Y. & \beta(v,y) \to \gamma(f(u,v),z) \\
+ & \phi_1(u,v,z): & F(u,v,z) = * \in 1. & \top \to \gamma(f(u,v),z)
\end{array}$$

where the maps ϕ_X, ϕ_Y, ϕ_1 are the results of pulling back ϕ along respectively $F^{-1}(X) \to U \times V \times Z$, $F^{-1}(Y) \to U \times V \times Z$, and $F^{-1}(Z) \to U \times V \times Z$.

A map from α to $\beta \supset \gamma$ in Dial$^+$:

$$\begin{tikzcd}
U \arrow[d, "{(f,H,k(f,H))}"'] \arrow[dr, dash] & & X+1 \arrow[ll, "\hat{\alpha}"', "|" marking] \\
 & {} \arrow[ur, "K"'] \arrow[dr, dash] & \\
\Pi\square & & V \times Z \arrow[ll, "\beta\supset\gamma", "|" marking]
\end{tikzcd}$$

$\psi(u,v,z) : \hat{\alpha}(u, K(u,v,z)) \to (\beta \supset \gamma)(f(u), H(u), k(f(u), H(u)), v, z)$

Again we use the coproduct properties to get:

$$\begin{array}{rllll}
\psi(u,v,z) & = & & & \\
 & \psi_X(u,v,z): & K(u,v,z) = x \in X. & \alpha \to (\beta \supset \gamma)(f(u),H(u),k(f(u),H(u)),v,z) \\
+ & \psi_1(u,v,z): & K(u,v,z) = * \in 1. & \top_{K^{-1}(1)} \to (\beta \supset \gamma)(f(u),H(u),k(f(u),H(u)),v,z)
\end{array}$$

Spelling out what this means:

$$\begin{tikzcd}
\hat{\alpha}(u, K(u,v,z)) \arrow[dr, "\psi"] \arrow[ddr, bend right] & & & \\
 & (\beta \supset \gamma)(f(u),H(u),k(f(u),H(u)),v,z) \arrow[rr] \arrow[d] & & \gamma \times V + [\beta,\gamma] \arrow[d] \\
 & U \times V \times Z \arrow[r, "{(f,H,k(f,H))}"'] & \square \arrow[r] & W \times V \times Z + [\beta,\gamma]
\end{tikzcd}$$

where the square is a pullback. Notice that $U \times V \times Z \cong H^{-1}(1) + H^{-1}(Y)$ and

$$(\beta \supset \gamma)(f, H, k(f,H), v, z) \cong \gamma(f(u,v),z) + H^{-1}(Y),$$

so the triangle in the diagram can be written

$$\begin{tikzcd}
\hat{\alpha}(u, K(u,v,z)) \arrow[r, "\psi"] \arrow[dr] & \gamma(f(u,v),z) + H^{-1}(Y) \arrow[d] \\
 & H^{-1}(1) + H^{-1}(Y)
\end{tikzcd}$$

Since pullbacks preserves coproducts and

$$U \times V \times Z \cong (K^{-1}(X) \wedge H^{-1}(1)) + (K^{-1}(X) \wedge H^{-1}(Y)) + (K^{-1}(1) \wedge H^{-1}(1)) + (K^{-1}(1) \wedge H^{-1}(Y)).$$

where $K^{-1}(1) \wedge H^{-1}(1)$ means pullback of $K^{-1}(1) \to U \times V \times Z$ and $H^{-1}(1) \to U \times V \times Z$. We then get

$$\begin{array}{rll} \psi_X = & \psi_{X,1}: & K(u,v,z) = x \in X \wedge H(u,v,z) = * \in 1. \quad \alpha(u,x) \to \gamma(f(u,v),z) \\ + & \psi_{X,Y}: & K(u,v,z) = x \in X \wedge H(u,v,z) = y \in Y. \quad \alpha(u,x) \to \top \\ \psi_1 = & \psi_{1,1}: & K(u,v,z) = * \in 1 \wedge H(u,v,z) = * \in 1. \quad \top \to \gamma(f(u,v),z) \\ + & \psi_{1,Y}: & K(u,v,z) = * \in 1 \wedge H(u,v,z) = y \in Y. \quad \top \to \top \end{array}$$

So when $H(u,v,z) \in Y$ we get no information from ψ, however,

$$(f, \widetilde{H, k(f,H)}) : U \times V \times Z \to \Box$$

and in particular, we have, in the fibre over $H^{-1}(Y)$,

$$\tilde{k}(f(u), H(u), v, z) : H(u,v,z) = y \in Y.[\beta(v, H(u,v,z), \gamma(f(u,v),z))].$$

Now we are ready to give the retraction.

$$I : \mathrm{Dial}^+(\alpha \& \beta, \gamma) \to \mathrm{Dial}^+(\alpha, \beta \supset \gamma)$$

I works as follows. Given f, F, ϕ we get

- $\tilde{f}$, the transposed of f by the Cartesian closure of $\mathcal{C}$.
- $H = U \times V \times Z \xrightarrow{F} X + Y + 1 \longrightarrow Y + 1$,
- $k(\tilde{f}(u), H(u), v, z) = \phi_Y(u,v,z) : F(u,v,z) = H(u,v,z) = y \in Y. \beta(v,y) \to \gamma(f(u,v),z)$,
- $K = U \times V \times Z \xrightarrow{F} X + Y + 1 \longrightarrow X + 1$
- $$\begin{array}{rllll} \psi(u,v,z) = & & & & \\ & \phi_X(u,v,z): & K^{-1}(X) \wedge H^{-1}(1) & = & F^{-1}(X). \quad \alpha(u, F(u,v,z)) \to \gamma(f(u,v),z) \\ + & \phi_1(u,v,z): & K^{-1}(1) \wedge H^{-1}(1) & = & F^{-1}(1). \quad \top \to \gamma(f(u,v),z) \\ + & \mathrm{id}: & K^{-1}(1) \wedge H^{-1}(Y) & = & F^{-1}(Y). \quad \top \to \top. \end{array}$$

Note that $K^{-1}(X) \wedge H^{-1}(Y) = 0$ since $F^{-1}(X) \wedge F^{-1}(Y) = 0$.

$$R : \mathrm{Dial}^+(\alpha, \beta \supset \gamma) \to \mathrm{Dial}^+(\alpha \& \beta, \gamma)$$

Given $(f, H, k(f,H)), K, \psi$, R returns

- $\hat{f}$, the transpose of f.
- $F = H \restriction_{H^{-1}(Y)} + K \restriction_{H^{-1}(1)}$,
- $\phi(u,v,z) : \alpha \hat{\&} \beta(u,v,F(u,v,z)) \to \gamma(f(u,v),z)$ is defined by

  $$\begin{array}{rlll} \phi(u,v,z) = & & & \\ & \psi_{X,1}: & F^{-1}(X) = K^{-1}(X) \wedge H^{-1}(1). & \alpha(u, F(u,v,z)) \to \gamma(f(u,v),z) \\ + & \psi_{1,1}: & F^{-1}(1) = K^{-1}(1) \wedge H^{-1}(1). & \top \to \gamma(f(u,v),z) \\ + & k(f(u), h(u), v, z): & F(u,v,z) = H(u,v,z) = y \in Y. & \beta(v,y) \to \gamma(f(u,v),z) \end{array}$$

  which is the same as saying $\phi = \psi \restriction_{H^{-1}(1)} + k$.

It is now straightforward to verify that $RI = \mathrm{id}$.

Naturality of the retraction Let

$$\begin{tikzcd}
U' \arrow[d, "t"'] \arrow[rd, dash] & X'+1 \arrow[l, "\hat{\alpha'}"', "\shortmid" marking] \\
U & X \arrow[l, "\alpha", "\shortmid" marking] \arrow[u, "T"']
\end{tikzcd}$$

$$\theta(u',x) : \hat{\alpha'}(u,v,F(u',T(u',x)) \to \alpha(t(u'),x)$$

and $(f,F,\phi) : \alpha\hat{\&}\beta \to \gamma$, and $(f,H,k,K,\psi) : \hat{\alpha} \to (\beta \supset \gamma)$. We know that $\mathrm{id}_\beta = (\mathrm{id}_V, \mu_Y \circ \pi_Y, \mathrm{id}_\beta)$ and

$$(t,T,\theta)\&\,\mathrm{id}_\beta = (t \times \mathrm{id}_V, \mu_{X+Y} \circ (T+\pi_Y), \theta(\pi(u',v'),x) + \mathrm{id}_\beta)$$

We must show that

$$I(f,F,\phi) \circ (t,T,\theta) = I((f,F,\phi) \circ ((t,T,\theta)\&\,\mathrm{id}_\beta)) \tag{10}$$

and

$$R((f,H,k,K,\psi) \circ (t,T,\psi)) = R(f,H,k,K,\psi) \circ ((t,T,\theta)\&\,\mathrm{id}_\beta) \tag{11}$$

For (10) consider

$$I(f,F,\phi) = (f,H,k=\phi_Y,K,\psi=\phi_X+\phi_1+\mathrm{id})$$

so the left hand side of (10) becomes

$$I(f,F,\phi) \circ (t,T,\theta) =$$

- $(f,H,\phi_Y) \circ t = (f(t(u'),v), H(t(u'),v,z), \phi_Y(t(u'))) : U' \to \Pi_\pi \Box$,
- $\hat{T}(u',K(t(u'),v,z)) : U' \times V \times Z \to X'+1$,
- the composite

$$\begin{tikzcd}
\hat{\alpha'}(u',\hat{T}(u',K(t(u'),v,z))) \arrow[d, "{\hat{\theta}(u',K(t(u'),v,z))}"] \\
\hat{\alpha}(t(u'),K(t(u'),v,z)) \arrow[d, "{\psi(t(u'),v,z)}"] \\
(\beta \supset \gamma)(f(t(u')),H(t(u')),\phi_Y(t(u')),v,z).
\end{tikzcd}$$

On the other hand

$I((f,F,\phi) \circ (t,T,\theta)\&\,\mathrm{id}_\beta) =$
$I[f(t(u'),v), (\widehat{T+\pi_Y})(u',v,F(t(u'),v,z)), (\phi(t(u'),v,z) \circ (\theta + \mathrm{id}_\beta))(u',v,F(t(u',v,z)))]$

which yields

- $f(t(u'),v)$,

17

- $H = U' \times V \times Z \xrightarrow{\langle u', v, F(t(u), v, z)\rangle} U' \times V \times (X + Y + 1) \xrightarrow{(\widehat{T + \pi_Y})} X' + Y + 1 \longrightarrow Y + 1$
- $k(ft(u'), H(t(u')), v, z) =$

$$\phi_Y(t(u'), v, z) : F(t(u'), v, z) = y \in Y.\hat{\beta}(v, y) \xrightarrow{\text{id}} \hat{\beta}(v, y) \xrightarrow{\phi_Y(t(u'), v, z)} \gamma(t(u'), v, z)$$

- $K = U' \times V \times Z \xrightarrow{\langle u', v, F(t(u), v, z)\rangle} U' \times V \times (X + Y + 1) \xrightarrow{\widehat{T + \pi_Y}} X' + Y + 1 \longrightarrow X' + 1$
- $$\begin{array}{rrl} \psi = & & \\ & \phi_X(t(u'), v, z) \circ \hat{\theta}(u', x) : & F(t(u'), v, z) = x \in X.\, \hat{\alpha}'(u', T(u', x)) \to \hat{\alpha}(t(u'), x) \to \gamma(t(u'), v, z) \\ + & \phi_1(t(u'), v, z) : & F(t(u'), v, z) \in 1.\, \top \to \gamma(t(u'), v, z) \\ + & \text{id} : & F(t(u'), v, z) = y \in Y.\, \top \to \top. \end{array}$$

which is easily seen to be equal.

To see that (11) holds, recall that

$$R(f, H, k, K, \psi) = (f, F = H \restriction_{H^{-1}(Y)} + K \restriction_{H^{-1}(1)}, \psi \restriction_{H^{-1}(1)} + k)$$

Now, the right hand side of 11 is

- $f(t(u'), v)$,
- $\widehat{T + \pi_Y}(u', v, F(t(u'), v, z))$,
- $$\begin{array}{rl} & \psi \restriction_{H^{-1}(1)} (t(u'), v, z) + k(t(u')) \circ \widehat{\theta + \text{id}_\beta}(u', v, F(t(u'), v, z)) \\ = & \psi \restriction_{H^{-1}(1)} (t(u'), v, z) + k(t(u')) \circ \hat{\theta}(u', F_{X+1}(t(u'), v, z) + \text{id}_\beta(v, F_Y(t(u'), v, z))) \\ = & \psi \restriction_{H^{-1}(1)} (t(u'), v, z) + k(t(u')) \circ \hat{\theta}(u', K \restriction_{H^{-1}(1)} (t(u'), v, z) + \text{id}_\beta(v, H_Y(t(u'), v, z))) \\ = & (\psi \restriction_{H^{-1}(1)} (t(u'), v, z) \circ \hat{\theta}(u', K \restriction_{H^{-1}(1)} (t(u'), v, z)) + k(t(u')) \\ = & (\psi \restriction_{H^{-1}(1)} (t(u'), v, z) \circ \hat{\theta}(u', K(t(u'), v, z)) \restriction_{H^{-1}(1)} + k(t(u')) \end{array}$$

On the other hand

$$(f, H, k, K, \psi) \circ (t, T, \phi) = (f(t(u')), H(t(u')), k(t(u')), \hat{T}(u'), K(t(u'), v, z), \psi(t(u'), v, z) \circ \hat{\theta}(t(u'), v, z))$$

applying R to this gives

- $f(t(u'), v)$
- $H(t(u'), v, z) \restriction_{H^{-1}(Y)} + \hat{T}(u', K(t(u'), v, z)) \restriction_{H^{-1}(1)}$
- $(\psi(t(u'), v, z) \circ \hat{\theta}(u', K(t(u'), v, z))) \restriction_{H^{-1}(1)} + k(t(u'))$.

□

It seems clear that this proof can be carried out in the general case of cloven fibrations with the appropriate structure, but we leave that for another occasion.

One way of thinking of a map (f, F, ϕ) in $\mathrm{Dial}(p)(A, B)$ is the following: given a witness u of $\exists u \forall x.\alpha(u, x)$, f provides a witness fu of $\exists v \forall y.\beta(v, y)$, and given a counter example y of $\forall y.\beta(fu, y)$, $F(u, y)$ is a counter example of $\forall x.\alpha(u, x)$, and ϕ is a proof of this. Now, for the Kleisli category Dial^+, the difference is that given a counter example of $\forall y.\beta(fu, y)$, $F(u, y)$ may either give a counter example of $\forall x.\alpha(u, x)$ or raise an exception.

In the same spirit, one may give the following intuitive characterization of the homsets $\mathrm{Dial}^+(A \times B, C)$ and $\mathrm{Dial}^+(A, B \supset C)$: The counter example part of $\mathrm{Dial}^+(A \times B, C)$ gives a counter example of α or β *exclusively* provided a counter example of γ. The counter example part of $\mathrm{Dial}^+(A, B \supset C)$ gives a counter example of α or β or both provided a counter example of γ. This gives some intuition as to why $\mathrm{Dial}^+(A, B \supset C)$ is "bigger" than $\mathrm{Dial}^+(A \times B, C)$.

5 Examples

Examples of fibrations that meet the conditions of Theorem 4.7 are, $\mathrm{cod}(\mathrm{PER}) \to \mathrm{PER}$ (equivalently, the split fibration $\mathrm{UFam}(\mathrm{PER}) \to \mathrm{PER}$), $\mathrm{cod}(\mathrm{Set}) \to \mathrm{Set}$ (equivalently, the split fibration $\mathrm{Fam}(\mathrm{Set}) \to \mathrm{Set}$), and for a topos $\mathcal{C}$, the codomain fibration $\mathrm{cod}(\mathcal{C}) \to \mathcal{C}$, and the subobject fibration $\mathrm{Sub}(\mathcal{C}) \to \mathcal{C}$.

5.1 A modest example

We will now spell out the details of one of the examples, namely the fibration

$$\begin{tikzcd} \mathrm{UFam}(\mathrm{PER}) \arrow[d] \\ \mathrm{PER} \end{tikzcd}$$

This example is important because it may provide us with the insight to give an extensional version of the Dialectica interpretation (corresponding to extensional realizability). Also some readers might like a concrete example.

Objects of UFam(PER) are collections $(A_{[n]})_{[n] \in \mathbb{N}/R}$ of partial equivalence relations (pers) indexed by a per R.

Morphisms from $(A_{[n]})_{[n] \in \mathbb{N}/R}$ to $(B_{[m]})_{[m] \in \mathbb{N}/S}$ are pairs (u, f), where $u : \mathbb{N}/R \to \mathbb{N}/S$ is a morphism in PER (that is, it is tracked by some $e_u \in \mathbb{N}$; $u([n]) = [e_u \cdot n]$) and $f = (f_{[n]} : A_{[n]} \to B_{u([n])})$ which is tracked uniformly, i.e., there is an $e_f \in \mathbb{N}$ such that for all $[n] \in \mathbb{N}/R$ and for all $m \in [n]$, $e_f \cdot m$ tracks $f_{[n]}$: $f_{[n]}([a]) = [(e_f \cdot m) \cdot a]$ for all $m \in [n]$.

We now describe some well-known closure properties for the category PER of partial equivalence relation (also known as the category of modest sets).

The category PER has finite limits. The terminal object is given by $\{(0, 0\})$, the product of two pers R and S is given by

$$R \times S = \{(n, m) \mid \mathrm{p}nR\mathrm{p}m \text{ and } \mathrm{p}'nS\mathrm{p}'m\}.$$

The pullback of

$$\begin{tikzcd} & S \arrow[d, "f"] \\ R \arrow[r, "g"'] & T \end{tikzcd}$$

is

$$\{[n] \in \mathbb{N}/(R \times S) \mid g([\mathrm{p}n]) = f([\mathrm{p}'n])\}$$

The initial object in PER is the empty set. The coproduct of R and S is

$$R + S = \{(n, m) \mid (\mathrm{p}n = \mathrm{p}m = 0 \text{ and } \mathrm{p}'nR\mathrm{p}'m) \text{ or } (\mathrm{p}n = \mathrm{p}m = 1 \text{ and } \mathrm{p}'nS\mathrm{p}'m)\}.$$

Proposition 5.1. *Coproducts in* PER *are stable and disjoint.*

Furthermore, PER has exponentials

$$R \Rightarrow S = \{(n, n') \mid \forall mm'.mRm' \Rightarrow n \cdot mSn' \cdot m'\}$$

We also have simple products, that is, for projections $\pi : I \times J \to I$ in PER, there is a right adjoint Π_π to π^*, it is defined as follows: For $(R_{i,j})_{(i,j)\in I\times J}$,

$$(\Pi_\pi R)_{[i]} = (\bigcap_{j\in|J|} \{c \mid \forall n \in [j].c{\cdot}n \in R_{i,j}\}, \sim), \text{ where } c \sim c' \text{ iff for all } j \in |J|.\forall n \in [j].c{\cdot}nR_{i,j}c'{\cdot}n.$$

Now we will turn to the Dialectica-Kleisli category Dial$^+$(UFam(PER)), for which we will describe the weak exponential in details. The product: $(U, X, \alpha) \times (V, Y, \beta) = (U \times V, X + Y, \alpha\&\beta)$, where for $n \in X + Y$,

$$(\alpha\&\beta)(u, v, n) = \begin{cases} \alpha(u, \mathrm{p}'n) & \text{if} \quad \mathrm{p}n = 0 \\ \beta(v, \mathrm{p}'n) & \text{if} \quad \mathrm{p}n = 1 \end{cases}$$

Now $\square$ in the fibre over $V \Rightarrow W \times V \times Z \Rightarrow 1 + Y \times V \times Z$ is defined by

$$\square(g, G, v, z) = \begin{cases} \beta(v, y) \Rightarrow \gamma(gv, z) & \text{if } G(v, z) = y \in Y \\ \{(g, G, v, z) \mid G(v, z) \in 1\} & \text{if } G(v, z) \in 1 \end{cases}$$

And $\Pi_\pi\square$ in the fibre over $V \Rightarrow W \times V \times Z \Rightarrow 1 + Y$ is

$$\begin{array}{ll} (\Pi_\pi\square)(g, G) = & (\bigcap_{v,z}\{k(g, G) \in \mathbb{N} \mid \forall n \in [(v, z)].k(g, G) \cdot n \in \square(g, G, v, z), \sim) \\ \text{where} & k(g, G) \sim k'(g, G) \text{ iff } \forall v, z \in |V \times Z| \\ & \forall n \in [v, z].k(g, G) \cdot n\square(g, G, v, z)k'(g, G) \cdot n \end{array}$$

And, finally $\beta \supset \gamma$ in the fibre over $(\Pi\square) \times V \times Z$ is

$$(\beta \supset \gamma)(k(g, G) : (\Pi\square)(g, G), v, z) = \begin{cases} \gamma(gv, z) & \text{if} \quad G(v, z) \in 1 \\ \{(k(g, G), v, z) \mid G(v, z) \in Y\} & \text{if} \quad G(v, z) \in Y \end{cases}$$

6 Conclusion and Future Work

We have shown that Dialectica categories can be generalized to cloven fibrations and how, starting with a monad, one can construct comonads on the Dialectica category. We have shown how one particular non-Girardian comonad constructed from a monad gives rise to a weakly Cartesian closed Dialectica-Kleisli category.

The ideas presented in this paper suggest two new Dialectica variants. The first one based on the new exponent as first presented in the tripos version in [BBLBCB07]. One can expand Gödel's system T with stable, disjoint coproducts and subset types and then we can interpret implication as the new exponent. This has the advantage that we do not need the condition that primitive formulas have to be decidable (in the recursion theoretic sense). By now, this variant is described in [Bie07].

The second Dialectica variant that falls out of this work is a type theoretic one: instead of having formulas over Heyting arithmetic (this more or less corresponds to de Paiva's original Dialectica categories) we have dependent types over some type system, and the Dialectica interpretation turns the dependent type system into a lambda calculus without the η-rule.

There seem to be at least two monads that give rise to comonads with interesting Kleisli categories, the free commutative monoid monad gives rise to the Diller-Nahm category and the monad $- + 1$ gives a weakly Cartesian closed Dialectica category. One may ask if there are other comonads on Dialectica categories that gives interesting Kleisli categories. And in fact there is; this is studied in a realizability setting in [Bie08] and in a syntactical setting in [Oli08].

The PER example that we gave in Section 5.1 gives a model for an extensional version of the Dialectica interpretation, it would be interesting to describe this extensional version in details. Also the type theoretic variant of the Dialectica interpretation mentioned above deserves to be studied.

It would also be natural to find out what the closure properties are for the generalised Dialectica categories, to see if they are symmetric monoidal closed like the original ones.

7 Acknowledgement

I wish to express my gratitude to Martin Hyland who kindly hosted my 5 months visit in Cambridge, his advise and ideas has been of immense value to me. I would also like to thank Rasmus Ejlers Møgelberg, Lars Birkedal, Carsten Butz, Jaap van Oosten, Giuseppe (Pino) Rosolini, and Thomas Streicher for interesting, related discussions. I also thank the anonymous referee for many good suggestions.

A Cartesian closure of the Cauchy completion

In this appendix we include a proof of a folklore theorem:

Theorem A.1. *Let $\mathcal{C}$ be a category with finite products and a weak closed structure $[B, C]$, then the Cauchy completion $\bar{\mathcal{C}}$ is Cartesian closed.*

Let $\mathcal{C}$ be a category with finite products and a weak closed structure $[B, C]$, that is, we have a retraction

$$\begin{tikzcd}\mathcal{C}(A \times B, C) \arrow[r, shift left, "I"] & \mathcal{C}(A, [B, C]) \arrow[l, shift left, "R"]\end{tikzcd}$$

onto $\mathcal{C}(A \times B, C)$ (that is, $RI = \mathrm{id}$), natural in A.

In the internal language, having a weak exponent like this corresponds to having λ-calculus with the β-rule, but without the η-rule. The rule corresponding to the morphism R is

$$\frac{\Gamma \vdash N : A \to B}{\Gamma, x : A \vdash Nx : B}\ R$$

and the rule corresponding to I is

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}\ I$$

Consider

$$\frac{\dfrac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}\ I}{\Gamma, x : A \vdash (\lambda x : A.M)x : B}\ R$$

Since $RI = \mathrm{id}$ we get $M : B \overset{\beta}{=} (\lambda x : A.M)x : B$, substituting the free variable x for a term $N : A$ we get the β-rule

$$(\lambda x : A.M)N : B \overset{\beta}{=} M[N/x]$$

On the other hand, consider

$$\frac{\dfrac{\Gamma \vdash N : A \to B}{\Gamma, x : A \vdash Nx : B}\ R}{\Gamma \vdash \lambda x : A.(Nx) : A \to B}\ I$$

Since $IR \neq \mathrm{id}$ we can conclude that

$$N : A \to B \overset{\eta}{\neq} \lambda x : A.(Nx) : A \to B,$$

so we have no η-rule in the internal language. That is, not all terms of function type can be constructed by λ abstraction.

Naturality of I in A is: given $a : A' \to A$, $f : A \times B \to C$,

$$I(f \circ (a \times B)) = I(f) \circ a$$

Naturality of R in A is: given $a : A' \to A$, $g : A \to [B, C]$,

$$R(g \circ a) = R(g) \circ (a \times B).$$

Write $ev : [A,B] \times A \to B$ for $R(\mathrm{id}_{[A,B]})$ and $e : [A,B] \to [A,B]$ for $I(ev) = IR(\mathrm{id}_{[A,B]})$. Note that

$$ev \circ g \times B = R(\mathrm{id}_{[B,C]}) \circ (g \times B) = R(\mathrm{id} \circ g) = R(g)$$

and

$$IR(g) = I(ev \circ g \times B) = I(ev) \circ g = eg$$

It follows that

$$ev \circ (I(f) \times B) = RI(f) = f$$

i.e.,

$$\tilde{f}(a)(b) = f(a,b).$$

Definition A.2 (Notation)**.** *For $u : A_1 \to A$, $x : B \to B_1$, define*

$$[u,x] : [A,B] \to [A_1,B_1]$$

to be

$$I([A,B] \times A_1 \xrightarrow{[A,B]\times u} [A,B] \times A \xrightarrow{ev} B \xrightarrow{x} B_1)$$

Observe that $[\mathrm{id}_A, \mathrm{id}_B] = e : [A,B] \to [A,B]$.

Proposition A.3. *Let $g : A \to [B,C]$, $b : B' \to B$, $c : C \to C'$. Then*

$$\begin{aligned} [b,c] \circ g &= I(c \circ ev \circ [B,C] \times b) \circ g \\ &= I(c \circ ev[B,C] \times b \circ g \times B') \\ &= I(c \circ ev \circ g \times B \circ A \times b) \\ &= I(c \circ R(g) \circ A \times b). \end{aligned}$$

Proposition A.4. *Take $A_2 \xrightarrow{v} A_1 \xrightarrow{u} A$ and $B \xrightarrow{x} B_1 \xrightarrow{y} B_2$. Then*

$$\begin{aligned} [v,y] \circ [u,x] &= I(y \circ R([u,x]) \circ [A,B] \times v) \\ &= I(y \circ ([A,B] \circ u \circ ev \circ x) \circ [A,B] \times v) \quad \textit{as } RI = \mathrm{id} \\ &= I((yx) \circ ev \circ ([A,B] \times uv)) \\ &= [uv, yx]. \end{aligned}$$

We define $\bar{\mathcal{C}}$ to be the category of idempotents in $\mathcal{C}$. The objects are (A,a) with $a = a^2$ an idempotent on A. The maps $(A,a) \to (B,b)$ are the maps $f : A \to B$ with $bfa = f$. This is called the Cauchy completion of $\mathcal{C}$.

$\bar{\mathcal{C}}$ has products. Take $(A,a),(B,b)$ in $\bar{\mathcal{C}}$, and consider $(A \times B, a \times b)$. For (C,c) in $\bar{\mathcal{C}}$ consider $f = (f_1,f_2) : C \to (A \times B)$. We see that

$$a \times b \circ fc = f$$

if and only if

$$af_1c = f_1 \text{ and } bf_2c = f_2.$$

Thus $(A \times B, a \times b)$ is the product as required.

Now take $(B,b),(C,c)$ in $\bar{\mathcal{C}}$. We have

$$[b,c]^2 = [b,c] \circ [b,c] = [b^2,c^2] = [b,c].$$

So $[b,c]$ is an idempotent and $([B,C],[b,c])$ is in $\bar{\mathcal{C}}$.
Suppose $g : A \to [B,C]$ satisfies $g = [b,c] \circ g$. Then

$$eg = e[b,c]g = [\mathrm{id}_B, \mathrm{id}_C][b,c]g = [b,c]g = g.$$

So $g = IR(g)$. Also

$$\begin{aligned} R(g) &= R([b,c]g) \\ &= R([b,c]) \circ g \times B \\ &= (c \circ ev \circ [B,C] \times b) \circ g \times B \\ &= cR(\mathrm{id})(g \times B)(A \times b) \\ &= cR(g)(A \times b) \end{aligned}$$

That is $R(g)$ satisfies $R(g) = cR(g)(A \times b)$. Conversely suppose $f : A \times B \to C$ satisfies

$$cf(A \times b) = f.$$

Then

$$\begin{aligned} [b,c] \circ I(f) &= I(c \circ RI(f) \circ (A \times b)) \\ &= I(cf(A \times b)) \\ &= I(f). \end{aligned}$$

That is, $I(f)$ satisfies $I(f) = [b,c]I(f)$. it follows that I and R induce an isomorphism between maps

$$g : (A, \mathrm{id}_A) \to ([B,C],[b,c])$$

and

$$f : (A, \mathrm{id}_A) \times (B,b) \to (C,c)$$

in $\bar{\mathcal{C}}$. This is natural in A and the extension to an isomorphism between

$$g : (A,a) \to ([B,C],[b,c])$$

and

$$f : (A,a) \times (B,b) \to (C,c)$$

follows from: Suppose $g = [b,c]ga$ then $ga = [b,c]ga^2 = [b,c]ga = g$. It follows that

$$\begin{aligned} R(g) = R(ga) &= cR(ga)(A \times b) \\ &= cR(g)(a \times B)(A \times b) \\ &= cR(g)(a \times b). \end{aligned}$$

and conversely suppose $f = cf(a \times b)$ then $f(a \times B) = cf(A \times b)(a^2 \times B) = cf(A \times b)(a \times B) = f$ and also $f \circ (a \times B) = c(f \circ a \times B)(A \times b)$. It follows that

$$\begin{aligned} I(f) = I(f \circ A \times b) &= [b,c]I(f \circ a \times B) \\ &= [b,c]I(f)a. \end{aligned}$$

Thus $([B,C][b,c])$ is the function space of (B,b) to (C,c) in $\bar{\mathcal{C}}$ and $\bar{\mathcal{C}}$ is Cartesian closed.

References

[BBLBCB07] J.M.E. Hyland & J. van Oosten & G. Rosolini & T. Streicher B. Biering & L. Birkedal & C. Butz. Topos theoretic versions of Dialectica interpretations. 2007. Unpublished draft.

[Bie07] Bodil Biering. The Copenhagen interpretation. 2007. Submitted.

[Bie08] Bodil Biering. *Dialectica Interpretations, a Categorical Analysis.* PhD thesis, IT University of Copenhagen, 2008.

[CLW93] Aurelio Carboni, Stephen Lack, and R. F. C. Walters. Introduction to extensive and distributive categories. *J. Pure Appl. Algebra*, 84(2):145–158, 1993.

[dP89] V. C. V. de Paiva. The Dialectica categories. In *Categories in computer science and logic (Boulder, CO, 1987)*, volume 92 of *Contemp. Math.*, pages 47–62. Amer. Math. Soc., Providence, RI, 1989.

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

[Jac99] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics.* North-Holland Publishing Co., Amsterdam, 1999.

[Oli08] Paulo Oliva. Functional interpretations of linear and intuitionistic logic. *special issue of Information and Computation*, 2008.

Chapter 5

The Copenhagen Interpretation

This paper has been submitted for publication in Annals of Pure and Applied Logic. The reader should have a look at [Göd90] or the more recent survey paper [AF98], and [DN74]. Some readers may find it helpful to first have a look at the constructions in Chapters 2 and 4.

In this paper we present a new functional interpretation called the Copenhagen interpretation. The Copenhagen interpretation is a generalization of Dialectica which is not limited to decidable atomic formulas, thus it soundly interprets higher typed Heyting arithmetic, $\mathbf{HA}^\omega$, whereas the original Dialectica interpretation only interprets first order Heyting arithmetic, **HA**. Generalizing the Dialectica interpretation to higher types has been one of the aims for our research. With the Copenhagen interpretation, we reach this goal, though it is perhaps not as simple as one would have liked. Very recently, we have discovered a much simpler variant, which we believe also interprets $\mathbf{HA}^\omega$. This new variant will be presented in a future paper, but we give a rough sketch of the idea in an appendix of the this paper.

The Copenhagen interpretation is the direct result of the categorical analysis of the Dialectica and Diller-Nahm interpretations in [dP89, Hyl02], and the papers in Chapters 2 and 4. The basic structure was discovered during the research for the material in Chapter 2 and refined by Martin Hyland at a meeting in Copenhagen[1] in 2006, hence the name "Copenhagen interpretation". A thorough analysis of the clause for implication can be found in Chapter 4.

Apart from presenting the precise definition of the Copenhagen interpretation, there are two main results in this paper, namely the Soundness Theorem and the Axiomatization Theorem. Moreover, we state and prove precisely in what sense the Copenhagen interpretation (and in fact also the Diller-Nahm interpretation) is a generalization of Gödel's Dialectica interpretation, and finally we give a classical version of the Copenhagen interpretation.

References

[AF98] Jeremy Avigad and Solomon Feferman. Gödel's functional ("Dialectica") interpretation. In *Handbook of proof theory*, volume 137 of *Stud. Logic Found. Math.*, pages 337–405. North-Holland, Amsterdam, 1998.

[BBLBCB07] J.M.E. Hyland & J. van Oosten & G. Rosolini & T. Streicher B. Biering & L. Birkedal & C. Butz. Topos theoretic versions of Dialectica interpretations. 2007. Unpublished draft.

[DN74] Justus Diller and Werner Nahm. Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen. *Arch. Math. Logik Grundlagenforsch.*, 16:49–66, 1974.

[dP89] V. C. V. de Paiva. The Dialectica categories. In *Categories in computer science and logic (Boulder, CO, 1987)*, volume 92 of *Contemp. Math.*, pages 47–62. Amer. Math. Soc., Providence, RI, 1989.

[1]The authors of [BBLBCB07] have held two "Dialectica meetings" the first in Copenhagen in September 2006, the second in Genoa, June 2007

[Göd90] Kurt Gödel. *Collected works. Vol. II.* The Clarendon Press Oxford University Press, New York, 1990. Publications 1938–1974, Edited and with a preface by Solomon Feferman, pages 217–251.

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

The Copenhagen Interpretation

Bodil Biering

1 Introduction

We present a new functional interpretation, called the Copenhagen interpretation. Like the Diller-Nahm interpretation [DN74], the Copenhagen interpretation generalizes Gödel's Dialectica interpretation [Göd90, AF98, Sch06] from first order Heyting arithmetic **HA** to higher typed Heyting arithmetic $\mathbf{HA}^{\omega}$, or to be precise, the Dialectica interpretation requires atomic formulas to be decidable because otherwise it cannot interpret contraction, the Diller-Nahm and Copenhagen interpretations do not have this limitation. The Diller-Nahm and Copenhagen interpretations are two different ways of coping with the problem. The Diller-Nahm interpretation avoids choosing by using list types, whereas for the Copenhagen interpretation contraction is validated because the Copenhagen interpretation of conjunction is defined differently. Where the Diller-Nahm interpretation uses list types, the Copenhagen interpretation requires subset types, because the new interpretation of conjunction also forces a new interpretation of implication, and for that subset types are needed. Beside their apparent difference, the two generalizations, Diller-Nahm and Copenhagen, differs in interesting ways and this is illustrated by a detailed example.

The Copenhagen interpretation is the direct result of the categorical analysis of the Dialectica and Diller-Nahm interpretations in [dP89, Hyl02, BBLBCB07, Bie07]. The basic structure was presented in [BBLBCB07] and refined by Martin Hyland at a meeting in Copenhagen[1] in 2006, hence the name "Copenhagen interpretation". A analysis of the clause for implication can be found in [Bie07].

Apart from presenting the precise definition of the Copenhagen interpretation, there are two main results in this paper, namely the Soundness Theorem and the Axiomatization Theorem. Moreover, we state and prove precisely in what sense the Copenhagen interpretation (and in fact also the Diller-Nahm interpretation) is a generalization of Gödel's Dialectica interpretation, and finally we give a classical version of the Copenhagen interpretation.

2 Definition of Interpretation

In this section we give a complete description of the logical system that we shall use, and we also provide some models for the system. We then define the Copenhagen interpretation and illustrate the difference between the Dialectica, Diller-Nahm and Copenhagen interpretations by a detailed example.

[1]The authors of [BBLBCB07] have held two "Dialectica meetings" the first in Copenhagen in September 2006, the second in Genoa, June 2007

2.1 Higher Typed Heyting Arithmetic with Sum Types and Subset Types

We now describe the system $\mathbf{HA}^{\omega}_{+}$. It is essentially Gödel's system $\mathbf{T}$ (as described in [AF98] and in [Str01]) with the addition of quantifiers and sum types and subset types. For convenience we give the full definition.

Types: The set $\mathcal{T}$ of types is defined inductively as follows

- $N \in \mathcal{T}$ (type of natural numbers)
- $1 \in \mathcal{T}$ (unit or terminal type)
- $U, X \in \mathcal{T}$ then $U \times X \in \mathcal{T}$.
- $U, X \in \mathcal{T}$ then $U \Rightarrow X \in \mathcal{T}$.
- If X and U are types of $\mathcal{T}$, then $X + U$ is a type.
- If $\alpha(u, x), \beta(v, y)$ are formulas of $\mathbf{HA}^{\omega}_{+}$, then $\{(f, F) : U \Rightarrow V \times (U \times Y \Rightarrow X + 1) \mid \forall u, y.\, \mathtt{case}\, F(u, y) \in X.\alpha_C(u, F(u, y)) \rightarrow \beta_C(f(u), y)\}$ is a type.
- If s, t are closed terms of type U, then $1 + \{* : 1 \mid s =_U t\}$ is a type.

When we write

$$\mathtt{case}\, F(u, y) \in X.\alpha_C(u, F(u, y)) \rightarrow \beta_C(f(u), y)$$

it is short for

$$\begin{pmatrix} \mathtt{case}\, F(u, y) \in X. & \alpha_C(u, F(u, y)) \rightarrow \beta_C(f(u), y) \\ \mathtt{case}\, F(u, y) \in 1. & \top \end{pmatrix}$$

The reason for only allowing certain kinds of subset types is that we want to ensure that all types are non-empty. This rather ugly form of subset types is motivated by the fact that full subset types are not C-interpreted, this restricted form is both C-interpreted and enables us to prove the Axiomatization Theorem 5.1

Language : Countably many variables $x : U$ for each type $U \in \mathcal{T}$. The following constant symbols:

$$\begin{array}{lll} 0 : N, & \mathtt{succ} : N \rightarrow N, & \mathtt{R}^U : U \rightarrow (N \rightarrow U \rightarrow U) \rightarrow N \rightarrow U, \\ \mathtt{p}^{U,X} : U \rightarrow X \rightarrow U \times X, & \mathtt{pr}_0^{U,X} : U \times X \rightarrow U, & \mathtt{pr}_1^{U,X} : U \times X \rightarrow X, \\ \mathtt{inl}^{U,X} : U \rightarrow U + X, & \mathtt{inr}^{U,X} : X \rightarrow U + X, & * : 1. \end{array}$$

An equality predicate $=_U$ for each type U. For each $U, X \in \mathcal{T}$ an application operation $\mathtt{app}^{U,X}$ from $(U \rightarrow X) \times U \rightarrow X$.

Terms

$$t, s ::= \quad x : U \mid f : U \mid \mathtt{app}^{U,X}(t : U \rightarrow X, s : U) : X \mid \lambda x : U.(t : X) : U \rightarrow X$$

where f is a constant symbol, $\mathtt{app}^{U,X}(t, s)$ is often written $t(s)$. We also have term constructors $\mathtt{i}$ and $\mathtt{o}$ for subset types: if $\phi[t/x]$ is provable then $\mathtt{i}(t) : \{x : X \mid \phi\}$ is a term. If $s : \{x : X \mid \phi\}$ is a term, then $\mathtt{o}(t) : X$ is a term.

Formulas $\phi ::= s =_U t \mid \bot \mid \top \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \forall x.\phi \mid \exists x.\phi \mid \begin{pmatrix} \mathtt{case}\, t \in U. & \phi \\ \mathtt{case}\, t \in X. & \phi \end{pmatrix}$

By abuse of notation we will write

$$\left(\begin{array}{ll}\texttt{case}\, z \in W. & \phi(z)\\ \texttt{case}\, z \in R. & \psi(z)\end{array}\right) \quad \text{for} \quad \left(\begin{array}{ll}\texttt{case}\, z = \texttt{inl}(w). & \phi(w)\\ \texttt{case}\, z = \texttt{inr}(r). & \psi(r)\end{array}\right).$$

We notice that for every type U there is a distinguished closed term $0_U : \sigma$, defined as follows:

$$0_N = 0,\quad 0_{U\times X} = \texttt{p}(0_U, 0_X),\quad 0_{U\to X} = \lambda u : U.0_X,\quad 0_{U+X} = \texttt{inl}(0_U),$$
$$0_{\{(f,F):U\Rightarrow V\times(U\times Y\Rightarrow X+1)\mid\forall u,y.\,\texttt{case}\, F(u,y)\in X.\alpha_C(u,F(u,y))\to\beta_C(f(u),y)\}} = \texttt{p}(\lambda u : U.0_V, \lambda u, y.\texttt{inr}(*))$$

The logical system for $\mathbf{HA}^{\omega}_{+}$ is:

Propositional connectives Rules have the form

$$\frac{\text{premiss}}{\text{conclusion}}$$

and

$$\frac{P}{\overline{Q}} \quad \text{is short for} \quad \frac{P}{Q} \quad \text{and} \quad \frac{Q}{P},$$

$$P \equiv Q \quad \text{is short for} \quad \overline{P \vdash Q} \quad \text{and} \quad \overline{Q \vdash P}$$

$$\frac{}{\phi\vdash\phi}\,(1) \qquad \frac{}{\phi\vdash\top}\,(2) \qquad \frac{}{\bot\vdash\phi}\,(3)$$

$$\frac{\phi\vdash\psi \quad \psi\vdash\theta}{\phi\vdash\theta}\,(4) \qquad \frac{}{\phi\wedge\psi\vdash\phi}\,(5) \qquad \frac{}{\phi\wedge\psi\vdash\psi}\,(6)$$

$$\frac{\phi\vdash\psi \quad \phi\vdash\chi}{\phi\vdash\psi\wedge\chi}\,(7) \qquad \frac{}{\phi\vdash\phi\vee\psi}\,(8) \qquad \frac{}{\psi\vdash\phi\vee\psi}\,(9)$$

$$\frac{\phi\vdash\chi \quad \psi\vdash\chi}{\phi\vee\psi\vdash\chi}\,(10) \qquad \frac{\phi\vdash\psi\to\theta}{\overline{\phi\wedge\psi\vdash\theta}}\,(11)$$

$$(12)\quad \exists f : V^U, F : (X+1)^{U\times Y}\forall u,y.\left(\begin{array}{ll}\texttt{case}\, F(u,y)\in X. & \alpha_C(u,F(u,y))\to\beta_C(f(u),y)\\ \texttt{case}\, F(u,y)\in 1. & \beta_C(f(u),y)\end{array}\right)\equiv$$

$$\exists (f,F) : \{f : V^U, F : (X+1)^{U\times Y} \mid \forall u,y.\texttt{case}\, F(u,y)\in X.\alpha_C(u,F(u,y))\to\beta_C(f(u),y)\}$$
$$\forall u,y.\left(\begin{array}{ll}\texttt{case}\, F(u,y)\in X. & \top\\ \texttt{case}\, F(u,y)\in 1. & \beta_C(f(u),y)\end{array}\right)$$

$$(13)\quad \left(\begin{array}{ll}\texttt{case}\, z = \texttt{inl}(w). & \phi(w)\\ \texttt{case}\, z = \texttt{inr}(r). & \psi(r)\end{array}\right)\equiv (z = \texttt{inl}(w)\to\phi(w))\wedge(z=\texttt{inr}(r)\to\psi(r))$$

Quantifiers

$$\frac{\phi\vdash\psi}{\overline{\phi\vdash\forall x.\psi}}\; x\notin \mathrm{FV}(\phi)\,(14) \qquad \frac{\phi\vdash\psi}{\overline{\exists x.\phi\vdash\psi}}\; x\notin\mathrm{FV}(\psi)\,(15)$$

Induction scheme

$$\frac{\vdash \phi(0) \quad \vdash \forall n.(\phi(n) \to \phi(\mathtt{succ}(n)))}{\vdash \forall n.\phi(n)} \quad (16)$$

Defining equations for the constants

$$\begin{array}{lll} \neg 0 = \mathtt{succ}(x), & (\lambda x : \sigma.t)(s) = t[s/x], & \\ \mathtt{pr}_0(\mathtt{p}(x,y)) = x, & \mathtt{pr}_1(\mathtt{p}(x,y)) = y, & \mathtt{p}(\mathtt{pr}_0(z), \mathtt{pr}_1(z)) = z, \\ \mathtt{R}(x,y,0) = x, & \mathtt{R}(x,y,\mathtt{succ}(z)) = y(z, \mathtt{R}(x,y,z)), & \\ \mathtt{io}(t) = t, & \mathtt{oi}(t) = t. & \end{array}$$

We write $z \in X$ for $\exists x : X.z = \mathtt{inl}(x)$, where $z : X + Y$.

$$\frac{}{z : X + Y \vdash \neg(z \in X \wedge z \in Y)}\ , \qquad \frac{}{z : X + Y \vdash z \in X \vee z \in Y}$$

Equality Axioms

$$\frac{}{\vdash x =_U x}\ , \quad \frac{}{x =_U y \vdash y =_U x}\ , \quad \frac{}{x =_U y \wedge y =_U z \vdash x =_U z}\ , \quad \frac{}{s = r \vdash \mathsf{app}(t,s) = \mathsf{app}(t,r)}$$

Substitution [2]

$$\frac{\phi \vdash \psi}{\phi[t/x] \vdash \psi[t/x]}\ t \text{ free for } x \text{ in } \phi, \psi \qquad \frac{t = s \vdash \psi[t/x]}{t = s \vdash \psi[s/x]}\ t, s \text{ free for } x \text{ in } \psi.$$

Models of $\mathbf{HA}^{\omega}_{+}$ Examples of models for the system $\mathbf{HA}^{\omega}_{+}$ are **HRO**, hereditary recursive operations, **HEO**, hereditary effective operations. Any subobject fibration over a category $\mathbb{C}$, where $\mathbb{C}$ is regular $(=, \exists, \wedge)$, ccc $(\times, \Rightarrow)$ and lccc $(\to, \forall)$ and has finite, stable, disjoint coproducts (needed for the `case` construction), and has a natural numbers object. Note that the Dialectica tripos defined in [BBLBCB07] is *not* a model of the system $\mathbf{HA}^{\omega}_{+}$ because it doesn't validate the rule (12).

Definition 2.1 (Copenhagen Interpretation). *Suppose α and β are formulas of $\boldsymbol{HA}^{\omega}$ and $\alpha^C = \exists u \forall x.\alpha_C(u,x)$ and $\beta^C = \exists v \forall y.\beta_C(v,y)$.*

$$\begin{array}{lcl}
\alpha \in \{\top, \bot\}, & & \alpha^C = \alpha_C = \alpha. \\
(s =_U t)^C & = & \exists u : 1 + \{* : 1 \mid s =_U t\}. \begin{pmatrix} \mathtt{case} & u = \mathtt{inl}(*). & \bot \\ \mathtt{case} & u = \mathtt{inr}(*). & \top \end{pmatrix} \\
(\alpha \wedge \beta)^C & = & \exists u, v \forall z : X + Y. \begin{pmatrix} \mathtt{case} & z \in X. & \alpha_C(u,z) \\ \mathtt{case} & z \in Y. & \beta_C(v,z) \end{pmatrix} \\
(\alpha \to \beta)^C & = & \\
 & \exists \langle f, F \rangle : & \{U \Rightarrow V \times (U \times Y \Rightarrow X + 1) \mid \\
 & & \forall u, y.\mathtt{case}\, F(u,y) \in X.\alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\} \\
 & \forall u, y. & \begin{pmatrix} \mathtt{case}\, F(u,y) \in 1. & \beta_C(fu, y) \\ \mathtt{case}\, F(u,y) \in X. & \top. \end{pmatrix} \\
(\forall z.\alpha(z))^C & = & \exists f : Z \to U \forall z, x.\alpha_C(z, f(z), x) \\
(\exists z.\alpha(z))^C & = & \exists z, u \forall x.\alpha_C(z,u,x) \\
(\alpha \vee \beta)^C & = & \exists z \in U + V \forall x, y. \begin{pmatrix} \mathtt{case} & z \in U. & \alpha_C(z,x) \\ \mathtt{case} & z \in V. & \beta_C(z,y) \end{pmatrix} \\
\begin{pmatrix} \mathtt{case} & z \in W. & \alpha(z) \\ \mathtt{case} & z \in R. & \beta(z) \end{pmatrix}^C & = & \exists u, v \forall x, y. \begin{pmatrix} \mathtt{case} & z \in W. & \alpha_C(z,u,x) \\ \mathtt{case} & z \in R. & \beta_C(z,v,y) \end{pmatrix}
\end{array}$$

[2]The second, and somewhat unusual substitution rule is provably equivalent to the standard substitution rule: $t = s \wedge \phi[t/x] \vdash \psi[s/x]$.

If we add other predicate symbols P, these are interpreted as $P^C = P_C = P$.

Note that

$$\mathsf{case}\, F(u,y) \in X.\alpha_C(u, F(u,y)) \to \beta_C(f(u), y)$$

is short for

$$\begin{pmatrix} \mathsf{case}\, F(u,y) \in X. & \alpha_C(u, F(u,y)) \to \beta_C(f(u), y) \\ \mathsf{case}\, F(u,y) \in 1. & \top \end{pmatrix}$$

The Copenhagen interpretation translates a formula α of $\mathbf{HA}^{\omega}_{+}$ into a formula of the form $\exists u \forall x.\alpha_C(u,x)$, where $\alpha_C(u,x)$ is a quantifier-free formula. As for the Dialectica interpretation, the most complicated case is implication. Recall that for the Dialectica interpretation we have

$$(\alpha \to \beta)^D = \exists f : U \Rightarrow V,\ F : (U \times Y) \Rightarrow X \forall u, y.\alpha_D(u, F(u,y)) \to \beta_D(f(u), y)$$

and the intuition is that given a realizer u for $\exists u \forall x.\alpha_D(u,x)$, f provides a realizer $f(u)$ for $\exists v \forall y.\beta_D(v,y)$. At the same time, if y is a counterexample of $\forall y.\beta_D(f(u), y)$ then $F(u,y)$ is a counterexample for $\forall x.\alpha_D(u,x)$.

For the C-interpretation we have a similar situation except here, F can either send a counterexample to a counterexample or $F(u,y) = * \in 1$, in which case we can think of F as raising an exception, and F can do this when we know that the conclusion, $\beta_C(fu, y)$ is true.

Definition 2.2. *A formula ϕ is said to be* C-stable *respectively* D-stable *whenever $\phi^C = \phi$, respectively $\phi^D = \phi$, where $(-)^D$ is the Dialectica interpretation and where $=$ means that they are syntactically equal.*

One important observation to make about the Copenhagen interpretation is that it is *not* the case that quantifier-free formulas are C-stable. The Dialectica interpretation enjoys the property that quantifier-free formulas are D-stable, and one can exploit that to conclude that whenever a formula has the form $\phi = \exists u \forall x.\alpha(u,x)$ where α is quantifier-free, then $\phi^D = \phi$, so in particular the D-interpretation is idempotent. For the C-interpretation we have the following for formulas A, B of the form $A = \alpha_C$ and $B = \beta_C$:

$$\begin{array}{lcl}
(A \wedge B)^C & = & \forall z : 2. \begin{pmatrix} \mathsf{case}\, z = 0. & A \\ \mathsf{case}\, z = 1. & B \end{pmatrix} \\
(A \to B)^C & = & \exists F : \{F : 1 \to 2 \mid \mathsf{case}\, F(*) = 0.A \to B\}. \begin{pmatrix} \mathsf{case}\, F(*) = 0. & \top \\ \mathsf{case}\, F(*) = 1. & B \end{pmatrix} \\
(A \vee B)^C & = & \exists z : 2. \begin{pmatrix} \mathsf{case}\, z = 0. & A \\ \mathsf{case}\, z = 1. & B \end{pmatrix} \\
\begin{pmatrix} \mathsf{case}\, z \in X. & A(z) \\ \mathsf{case}\, z \in Y. & B(z) \end{pmatrix}^C & = & \begin{pmatrix} \mathsf{case}\, z \in X. & A(z), \\ \mathsf{case}\, z \in Y. & B(z) \end{pmatrix} \\
(\neg A)^C & = & \exists F : \{F : 1 \to 2 \mid \mathsf{case}\, F(*) = 0.\neg A\}. \begin{pmatrix} \mathsf{case}\, F(*) = 0. & \top \\ \mathsf{case}\, F(*) = 1. & \bot \end{pmatrix}
\end{array}$$

Proposition 2.3. *The C-interpretation is idempotent, i.e., for all formulas ϕ, $(\phi^C)^C = \phi^C$.*

Proof: By induction on the structure of formulas, we easily show that for all α, where $\alpha^C = \exists u \forall x.\alpha_C(u,x)$, we have $\alpha_C(u,x)^C = \alpha_C(u,x)$, so

$$\begin{array}{rcl}
(\alpha^C)^C & = & (\exists u \forall x.\alpha_C(u,x))^C \\
& = & \exists u \forall x.\alpha_C(u,x)^C \\
& = & \exists u \forall x.\alpha_C(u,x)
\end{array}$$

□

2.2 Example

In the following we give a detailed example of a formula that have different C-, D-, and DN-interpretations.

We shall look at the formula

$$\psi = \forall f : N \to N \exists x : N(P(x) \to P(f(x)))$$

Assuming that P is a new predicate symbol over the type N, which is undecidable, we show that for the D-interpretation we are unable to define a realizer for ψ, and that the two generalizations of Dialectica, namely the Diller-Nahm and the Copenhagen interpretations give very different realizers. The point is not to try to show superiority of one interpretation over the others, but merely to illustrate the differences. Very informally, one might say that this example illustrates that while the Dialectica interpretation forces us to provide a realizer right away - whether needed or not, the Copenhagen interpretation postpones this to the very last moment. One may therefore think of the Copenhagen interpretation as a sort of call-by-name Dialectica, again this is very informal.

Classically, ψ is true, because

- If $P(f(0)) = \top$, then we put $x := 0$
- If $P(f(0)) = \bot$, then we put $x := f(0)$

The negative translation of ψ is

$$\psi^N = \forall f : N \to N \neg\neg\exists x : N(\neg\neg P(x) \to \neg\neg P(f(x)))$$

which is intuitionistically equivalent [3] to

$$\psi^N = \forall f : N \to N \neg\neg\exists x : N\neg\neg(\neg P(x) \lor P(f(x)))$$

so this holds for intuitionistic logic, $\mathbf{HA}^\omega$. Before we go on with the various functional interpretations of ψ^N, we need to consider:

Markov's principle The Dialectica interpretation validates the following version of Markov's Principle:

$$\mathrm{MP}_{\mathbf{Dia}} \qquad (\forall y\theta(y) \to \psi) \to \exists y.(\theta(y) \to \psi)$$

where θ and ψ are quantifier-free. In case θ is $\neg\phi$ and ψ is $\bot$ we have

$$\neg\forall y\neg\phi \to \exists\neg\neg\phi$$

and since intuitionistic logic validates

$$\neg\exists y\phi \leftrightarrow \forall y\neg\phi$$

we get

$$\neg\neg\exists y\phi \to \exists y\neg\neg\phi$$

[3]In classical logic we have $(A \to B) \leftrightarrow (\neg A \lor B)$. Suppose A and B are atomic formulas, then the negative translation of this is $(\neg\neg A \to \neg\neg B) \leftrightarrow \neg(\neg\neg A \land \neg B)$ and the latter is equivalent in **HA** to $\neg\neg(\neg A \lor B)$.

Thus if ϕ is double negation closed, we have

$$\mathbf{HA}^{\omega} + \text{MP}_{\mathbf{Dia}} \vdash \neg\neg\exists y\phi \to \exists y\phi$$

Hence $\text{MP}_{\mathbf{Dia}}$ gives us that

$$\forall f.\exists x.\neg\neg(\neg P(x) \vee P(f(x)))$$

holds in $\mathbf{HA}^{\omega} + \text{MP}_{\mathbf{Dia}}$. The Dialectica interpretation of ϕ^N is thus

$$\exists F : N^N \to N \forall f : N^N.\neg\neg(\neg P(F(f)) \vee P(f(F(f))))$$

Now the term F that one would *like* to define is

$$F(f) = \begin{cases} 0 & \text{if} \quad P(f(0)) \\ f(0) & \text{if} \quad \neg P(f(0)) \end{cases}$$

But since P is not recursive, F is not either.

The Diller-Nahm interpretation[4] validates another version of MP (see [DN74]):

$$\text{MP}_{\mathbf{DN}} : \qquad (\forall y A \to B) \to \exists W : \mathcal{P}_f(Y)(\forall w : W A \to B)$$

Put $B = \bot$ and $A = \neg\phi$ then we get

$$\neg\forall y\neg\phi \to \exists W : \mathcal{P}_f(Y).\neg(\forall w : W\neg\phi)$$

Since W is a finite set, $\forall w : W\neg\phi$ is a conjunction: $\bigwedge_{w\in W} \neg\phi(w)$, and in intuitionistic logic we have

$$\neg(\bigwedge_{w\in W} \neg\phi(w)) \dashv\vdash \neg\neg(\bigvee_{w\in W} \phi(w))$$

We are now able to deduce

$$\neg\neg\exists y\phi \to \exists W : \mathcal{P}_f(Y).\neg\neg(\bigvee_{w\in W} \phi(w))$$

We now turn to the Diller-Nahm interpretation of ϕ^N. Let

$$Q(f,x) = \neg P(x) \vee P(f(x)).$$

We have

$$\begin{aligned} \mathbf{HA}^{\omega} + \text{MP}_{\mathbf{DN}} &\vdash \forall f : N^N \exists W : \mathcal{P}_N.\neg\neg \textstyle\bigvee_{w\in W} \neg\neg Q(f,w) \\ &\vdash \forall f : N^N \exists W : \mathcal{P}_N.\neg\neg \textstyle\bigvee_{w\in W} Q(f,w) \end{aligned}$$

The Diller-Nahm interpretation of this is

$$\exists F : (N^N) \to \mathcal{P}_f(N).\forall f : N^N.\neg\neg \bigvee_{w\in F(f)} Q(f,w).$$

We define the realizer F as

$$F(f) = \{0, f(0)\}$$

[4]We have not defined the type $\mathcal{P}_f(N)$ in our formal system, since it is used only for the Diller-Nahm interpretation, which is not our main focus. $\mathcal{P}_f(N)$ is the inductively defined type consisting of codes of finite sets of natural numbers by natural numbers.

and we have

$$\begin{array}{rll} \mathbf{HA}^{\omega} & \vdash & \forall f : N^N.\neg\neg(\neg P(f(0)) \vee P(f(0))) \\ & \vdash & \forall f : N^N.\neg\neg(\neg P(0) \vee P(f(0)) \vee P(f(0)) \vee Pf(f(0)))) \\ & \dashv\vdash & \forall f : N^N.\neg\neg(Q(f,0) \vee Q(f,f(0))). \end{array}$$

For the Copenhagen interpretation, things get a bit more elaborate, let us first consider the interpretation of $\neg\neg\exists u.\phi_C(u)$.

$$(\exists u.\phi_C(u) \to \bot)^C = \exists F : \{F : U \Rightarrow 2 \mid \forall u.\texttt{case}\, Fu = 0.\neg\phi_C(u)\}\forall u : U.\gamma(F,u)$$

where

$$\gamma(F,u) = \left(\begin{array}{ll} \texttt{case} & Fu = 1.\bot \\ \texttt{case} & Fu = 0.\top \end{array}\right)$$

Let

$$\Box = \{F : U \Rightarrow 2 \mid \forall u.\texttt{case}\, Fu = 0.\neg\phi_C(u)\}$$

Intuitively, $F \in \Box$ is a partial characteristic function for $\phi_C(u)$ in the sense that F might give us some information of ϕ_C, but not necessarily all. Note that, if ϕ_C is recursive, then there is a characteristic function $\chi_{\phi_C} \in \Box$.

$$\begin{array}{ll} (\neg\neg\exists u.\phi_C(u))^C = & \\ \exists G : \{G : \Box \Rightarrow U+1 \mid \forall F \in \Box.\texttt{case}\, G(F) \in U.\neg\gamma(F,G(F)))\} & \forall F \in \Box.\alpha(G,F) = \\ \exists G : \{G : \Box \Rightarrow U+1 \mid \forall F \in \Box.\texttt{case}\, G(F) \in U.F(GF) = 1\} & \forall F \in \Box.\alpha(G,F) \end{array}$$

where

$$\alpha(G,F) = \left(\begin{array}{ll} \texttt{case} & G(F) \in 1.\bot \\ \texttt{case} & G(F) \in U.\top \end{array}\right)$$

Intuitively this holds if there is a $G : \Box \Rightarrow U + 1$ such that for all partial characteristic functions F, G provides a point $u \in U$ that satisfies $Fu = 1$, i.e., u is not a counter example of ϕ_C.

We are now ready to give the Copenhagen interpretation of ψ^N:

$$\begin{array}{l} (\forall f : N^N\neg\neg\exists x : N\neg\neg(\neg P(x) \vee P(f(x))))^C = \\ \exists H : N^N \to \{G : \Box \Rightarrow N+1 \mid \forall F \in \Box.\texttt{case}\, G(F) \in N.F(G(F)) = 1\} \\ \forall f : N^N.\forall F \in \Box.\alpha(H(f),F,f) \end{array}$$

where

$$\alpha(H,F,f) = \left(\begin{array}{ll} \texttt{case} & H(f)(F) \in 1.\bot \\ \texttt{case} & H(f)(F) \in N.\top \end{array}\right)$$

So we to show that this holds, we must find a realizer H which for all $f : N^N$ provides a $G : \Box \Rightarrow N + 1$ satisfying $\forall F \in \Box.F(G(F)) = 1$. We define H by:

$$H(f)(F) = \left\{\begin{array}{lll} 0 & \text{if} & F(0) = 1 \\ f(0) & \text{if} & F(0) = 0 \end{array}\right.$$

To see that this H works we reason as follows: if $F(0) = 1$ we have found our point $G(F) \in N$ with $F(G(F)) = 1$. If $F(0) = 0$, then by definition of the type $\Box$, we have that $\neg(\neg(P(0) \vee P(f(0)))) = \neg Q(f,0)$. Now we have $\mathbf{HA}^{\omega}_{+} \vdash F(n) = 0 \vee F(n) = 1$ for all $n : N$, and by definition of $\Box$, we have $F(f(0)) = 0 \to \neg(\neg(P(f(0)) \vee P(ff(0)))) = \neg Q(f, f(0))$. Recall that we have

$$\mathbf{HA}^{\omega}_{+} \vdash \neg\neg(Q(f,0) \vee Q(f,f(0)))$$

which is equivalent to

$$\mathbf{HA}^{\omega}_{+} \vdash (\neg Q(f,0) \wedge \neg Q(f,f(0))) \to \bot$$

So assuming $F(0) = 0$ we get $\mathbf{HA}^{\omega}_{+} \vdash F(f(0)) = 0 \to \bot$ hence $\mathbf{HA}^{\omega}_{+} \vdash F(f(0)) = 1$.

3 Soundness

The soundness Theorem is the first of our two main results. It states that if a formula ϕ is provable in $\mathbf{HA}^{\omega}_{+}$ then the C-interpreted formula ϕ^C is also provable in $\mathbf{HA}^{\omega}_{+}$, which in turn means that there is a term t (which is actually a primitive recursive function in the standard model) of $\mathbf{HA}^{\omega}_{+}$ such that

$$\mathbf{HA}^{\omega}_{+} \vdash \forall x.\phi_C(t,x).$$

Lemma 3.1. *$\boldsymbol{HA}^{\omega}_{+} \vdash (\alpha \to \beta)^C$ iff there exists terms $g : U \Rightarrow V$, $G : (U \times Y) \Rightarrow X$ in $\boldsymbol{HA}^{\omega}_{+}$ such that $\boldsymbol{HA}^{\omega}_{+} \vdash \forall u, y.\alpha_C(u, G(u,y)) \to \beta_C(gu, y)$.*

Proof: Assume $\mathbf{HA}^{\omega}_{+} \vdash (\alpha \to \beta)^C$ this means that we have terms

$$(f, F) : \{f : V^U, F : (X+1)^{U\times Y} \mid \forall u, y.\mathtt{case}\, F(u,y) \in X.\alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\}$$

such that

$$\mathbf{HA}^{\omega}_{+} \vdash \forall u, y. \begin{pmatrix} \mathtt{case}\, F(u,y) \in X. & \top \\ \mathtt{case}\, F(u,y) \in 1. & \beta_C(fu, y) \end{pmatrix}$$

Let $g = f$ and

$$G(u,y) = \begin{cases} F(u,y) & \text{if} \quad F(u,y) \in X \\ 0_X & \text{if} \quad F(u,y) \in 1 \end{cases}$$

Clearly we have

$$\mathbf{HA}^{\omega}_{+} \vdash \forall u, y.\alpha_C(u, G(u,y)) \to \beta_C(gu, y)$$

On the other hand assume we are given g, G as above, then $f = g, F = \mathtt{inl} \circ G$ have type $\{f : V^U, F : (X+1)^{U\times Y} \mid \forall u, y.\mathtt{case}\, F(u,y) \in X.\alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\}$ and so we get

$$\begin{array}{ll} \mathbf{HA}^{\omega}_{+} \vdash & \exists (f,F) : \{f : V^U, F : (X+1)^{U\times Y} \mid \forall u, y.\mathtt{case}\, F(u,y) \in X.\alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\} \\ & \forall u, y. \begin{pmatrix} \mathtt{case}\, F(u,y) \in X. & \top \\ \mathtt{case}\, F(u,y) \in 1. & \beta_C(fu, y) \end{pmatrix} \end{array}$$

$\Box$

Theorem 3.2 (Soundness). *If $\boldsymbol{HA}^{\omega}_{+} \vdash \alpha$ then $\boldsymbol{HA}^{\omega}_{+} \vdash \alpha^C$*

Proof: By induction on length of proofs in $\mathbf{HA}^{\omega}$. We show a few of the more interesting cases.

Equality We show that the transitivity rule for equality is C-interpreted, i.e., that $\mathbf{HA}^{\omega}_{+} \vdash (x = y \wedge y = z \to x = z)^C$. To do this we must find realizers

$$f : (1 + \{* : 1 \mid x = y\}) \times (1 + \{* : 1 \mid y = z\}) \to (1 + \{* : 1 \mid x = z\})$$

and

$$F : (1 + \{* : 1 \mid x = y\}) \times (1 + \{* : 1 \mid y = z\}) \Rightarrow 2$$

such that for all $u : 1 + \{* : 1 \mid x = y\}$, $v : 1 + \{* : 1 \mid y = z\}$,

$$\left(\begin{array}{ll} \texttt{case}\, F(u,v) = 0. & \left(\begin{array}{ll} \texttt{case}\, u = \texttt{inl}(*). & \bot \\ \texttt{case}\, u = \texttt{inr}(*). & \top \end{array} \right) \\ \texttt{case}\, F(u,v) = 1. & \left(\begin{array}{ll} \texttt{case}\, v = \texttt{inl}(*). & \bot \\ \texttt{case}\, v = \texttt{inr}(*). & \top \end{array} \right) \end{array} \right)$$

implies

$$\left(\begin{array}{ll} \texttt{case}\, f(u,v) = \texttt{inl}(*). & \bot \\ \texttt{case}\, f(u,v) = \texttt{inr}(*). & \top \end{array} \right)$$

This holds for the following definitions of f, F:

$$f(u,v) = \begin{cases} \texttt{inr}(*) & \text{if } u = \texttt{inr}(*) \text{ and } v = \texttt{inr}(*) \\ \texttt{inl}(*) & \text{otherwise} \end{cases} \qquad F(u,v) = \begin{cases} 0 & \text{if } u = \texttt{inl}(*) \\ 1 & \text{otherwise.} \end{cases}$$

(11) We show that the rule

$$\frac{\phi \vdash \psi \to \theta}{\phi \wedge \psi \vdash \theta} \ (11)$$

is C-interpreted.

$$\mathbf{HA}^{\omega}_{+} \vdash (\phi \to (\psi \to \theta))^C \tag{1}$$

holds if and only iff

$$\begin{array}{l} \exists (g_1, g_2) : U \Rightarrow \{(f, F) : W^V \times (V \times Z)^{Y+1} \mid \forall v, z. \texttt{case}\, F(v,z) \in Y. \psi_C(v, F(v,z) \to \theta_C(fv, z))\} \\ \exists G : U \times V \times Z \Rightarrow X \\ \forall u, v, z. \quad \phi_C(u, G(u,v,z)) \to \left(\begin{array}{ll} \texttt{case}\, g_2(u)(v,z) \in 1. & \theta_C(g_1(u)(v), z) \\ \texttt{case}\, g_2(u)(v,z) \in Y. & \top \end{array} \right) \end{array}$$

$$\mathbf{HA}^{\omega}_{+} \vdash (\phi \wedge \psi \to \theta)^C \tag{2}$$

holds if and only iff

$$\begin{array}{ll} \exists h : U \times V \Rightarrow W, & H : U \times V \times Z \Rightarrow X + Y \\ \forall u, v, z. & \left(\begin{array}{ll} \texttt{case}\, H(u,v,z) \in X. & \phi_C(u, H(u,v,z)) \\ \texttt{case}\, H(u,v,z) \in Y. & \psi_C(v, H(u,v,z)) \end{array} \right) \to \theta_C(h(u,v), z). \end{array}$$

Assume (1) holds, then we define

$$h(u,v) = g_1(u)(v), \qquad H(u,v,z) = \begin{cases} g_2(u)(v,z) & \text{if } g_2(u)(v,z) \in Y \\ G(u,v,z) & \text{otherwise.} \end{cases}$$

Assume (2) holds then we define

$$g_1(u)(v) = h(u,v), \qquad g_2(u)(v,z) = \begin{cases} H(u,v,z) & \text{if } H(u,v,z) \in Y \\ * \in 1 & \text{otherwise,} \end{cases}$$

$$G(u,v,z) = \begin{cases} H(u,v,z) & \text{if } H(u,v,z) \in X \\ 0_X & \text{otherwise.} \end{cases}$$

(14) We show that the rule number (14)

$$\frac{\phi \vdash \psi}{\phi \vdash \forall x.\psi}\ x \notin \mathrm{FV}(\phi)$$

is C-interpreted. $\mathbf{HA}^{\omega}_{+} \vdash (\phi \to \forall z.\psi(z))^C$ is equivalent to

$$\exists g : U \Rightarrow V^Z, G : Z \times U \times Y \Rightarrow X.\forall z, u, y.\phi_C(u, G(z, u, y)) \to \psi_C(z, g(u, z), y)$$

and $\mathbf{HA}^{\omega}_{+} \vdash (\phi \to \psi(z))^C$ is equivalent to

$$\exists h : Z \times U \Rightarrow V,\ H : Z \times U \times Y \Rightarrow X.\forall z, u, y.\phi_C(u, H(z, u, y)) \to \psi_C(h(z, u), y, z)$$

and these two are clearly equivalent.

(16) This is the induction scheme. Suppose that we have $u_0 : U$ such that $\forall x.\phi_C(0, u_0, x)$ and

$$\begin{array}{lll} (f, F) : & N \Rightarrow & \{(U \Rightarrow U) \times (U \times X \Rightarrow X + 1) \mid \forall u, x.\mathtt{case}\, F(n)(u, x) \in X. \\ & & \phi_C(n, u, F(n)(u, x)) \to \phi_C(\mathtt{succ}(n), f(n)(u), x)\} \\ \text{such that} & & \\ \forall n, u, x. & & \left(\begin{array}{ll} \mathtt{case}\, F(n)(u, x) \in X. & \top \\ \mathtt{case}\, F(n)(u, x) \in 1. & \phi_C(\mathtt{succ}(n), f(n)(u), x) \end{array}\right) \end{array}$$

We must provide a term $g : N \to U$ such that $\forall n, x.\phi_C(n, g(n), x)$. We define g by recursion as follows:

$$g(0) = u_0, \qquad g(\mathtt{succ}(n)) = f(n)(g(n)),$$

to be precise we use the recursion operator $\mathtt{R}$ to define g:

$$g(n) := \mathtt{R}(u_0, f, n).$$

(7) $\dfrac{\phi \vdash \psi \quad \phi \vdash \chi}{\phi \vdash \psi \wedge \chi}$ Suppose we are given $f : U \Rightarrow V, F : U \times Y \Rightarrow X$ with

$$\forall u, y.\phi_C(u, F(u, y)) \to \psi_C(fu, y)$$

and $g : U \Rightarrow W,\ G : U \times Z \Rightarrow X$ with

$$\forall u, z.\phi_C(u, G(u, z)) \to \chi_C(gv, z)$$

we must define $h : U \Rightarrow V \times W,\ H : U \times (Y + Z) \Rightarrow X$ with

$$\forall u : U, q : Y + Z.\phi_C(u, H(u, q)) \to \left(\begin{array}{ll} \mathtt{case}\, q \in Y. & \phi_C(\pi_1 h(u), q) \\ \mathtt{case}\, q \in Z. & \chi_C(\pi_2 h(u), q) \end{array}\right)$$

Clearly the following realizers satisfy the requirement:

$$h(u) = (f(u), g(u)) \qquad H(u, q) = \left\{ \begin{array}{lll} F(u, y) & \text{if} & q = \mathtt{inl}(y) \\ G(u, z) & \text{if} & q = \mathtt{inr}(z) \end{array}\right.$$

(13) $$\left(\begin{array}{ll}\texttt{case}\, z = \texttt{inl}(w). & \phi(w)\\ \texttt{case}\, z = \texttt{inr}(r). & \psi(r)\end{array}\right) \equiv (z = \texttt{inl}(w) \to \phi(w)) \wedge (z = \texttt{inr}(r) \to \psi(r))$$

$$\left(\begin{array}{ll}\texttt{case}\, z = \texttt{inl}(w). & \phi(w)\\ \texttt{case}\, z = \texttt{inr}(r). & \psi(r)\end{array}\right)^C = \exists u, v. \forall x, y. \left(\begin{array}{ll}\texttt{case}\, z = \texttt{inl}(w). & \phi_C(w,u,x)\\ \texttt{case}\, z = \texttt{inr}(r). & \psi_C(r,v,y)\end{array}\right)$$

and

$$\begin{array}{l}((z = \texttt{inl}(w) \to \phi(w)) \wedge (z = \texttt{inr}(r) \to \psi(r)))^C = \\ \exists (u,F) : \{u : U, F : X \Rightarrow 2 \mid \forall x : X. \texttt{case}\, Fx = 0. z = \texttt{inl}(w) \to \phi_C(w,u,x)\} \\ \exists (v,G) : \{v : V, G : Y \Rightarrow 2 \mid \forall y : Y. \texttt{case}\, Gy = 0. z = \texttt{inr}(r) \to \psi_C(r,v,y)\} \\ \forall q : X + Y. \left(\begin{array}{ll}\texttt{case}\, q \in X. & \left(\begin{array}{ll}\texttt{case}\, Fx = 0. & \top\\ \texttt{case}\, Fx = 1. & \phi_C(w,u,x)\end{array}\right)\\ \texttt{case}\, q \in Y. & \left(\begin{array}{ll}\texttt{case}\, Gy = 0. & \top\\ \texttt{case}\, Gy = 1. & \psi_C(r,v,y)\end{array}\right)\end{array}\right)\end{array}$$

Showing that the former C-implies the latter, we define $h(z)(u,v) = (u,F,v,G)$ where $Fx = 1 = Gy$ for all x, y. And

$$H(z)(u,v,q) = \begin{cases}(x, 0_Y) & \text{if} \quad q = \texttt{inl}(x)\\ (0_X, y) & \text{if} \quad q = \texttt{inr}(y)\end{cases}$$

The other direction: $k(z)(u,F,v,G) = (u,v)$, and

$$K(z)(u,F,v,G,x,y) = \begin{cases}\texttt{inl}(x) & \text{if} \quad z \in W\\ \texttt{inr}(y) & \text{if} \quad z \in R.\end{cases}$$

$$\exists f : V^U, F' : (X+1)^{U \times Y} \forall u, y. \left(\begin{array}{ll}\texttt{case}\, F(u,y) \in X. & \alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\\ \texttt{case}\, F(u,y) \in 1. & \phi_C\end{array}\right) \equiv$$

(12) $$\begin{array}{l}\exists (f,F) : \{f : V^U, F : (X+1)^{U \times Y} \mid \forall u, y. \texttt{case}\, F(u,y) \in X. \alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\} \\ \forall u, y. \left(\begin{array}{ll}\texttt{case}\, F(u,y) \in X. & \top\\ \texttt{case}\, F(u,y) \in 1. & \phi_C\end{array}\right)\end{array}$$

In order to show that

$$\exists f : V^U, F' : (X+1)^{U \times Y} \forall u, y. \left(\begin{array}{ll}\texttt{case}\, F(u,y) \in X. & \alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\\ \texttt{case}\, F(u,y) \in 1. & \phi_C\end{array}\right) \tag{3}$$

is C-equivalent to

$$\begin{array}{l}\exists (h,H) : \{h : V^U, H : (X+1)^{U \times Y} \mid \forall u, y. \texttt{case}\, H(u,y) \in X. \alpha_C(u, H(u,y)) \to \beta_C(h(u), y)\} \\ \forall u, y. \left(\begin{array}{ll}\texttt{case}\, H(u,y) \in X. & \top\\ \texttt{case}\, H(u,y) \in 1. & \beta_C(hu, y)\end{array}\right)\end{array} \tag{4}$$

we first show $\mathbf{HA}^{\omega}_{+} \vdash (3 \to 4)^C$ and then $\mathbf{HA}^{\omega}_{+} \vdash (4 \to 3)^C$. We start by calculating 3^C:

$$\begin{array}{rcl}3^C & = & \exists f : V^U, F : (X+1)^{U \times Y}, G : U \times Y \Rightarrow \{z : 2 \mid \texttt{case}\, z = 0. \alpha_C(u, F(u,y)) \to \beta_C(f(u), y)\} \\ & & \forall u, y. \left(\begin{array}{ll}\texttt{case}\, F(u,y) \in X. & \left(\begin{array}{ll}\texttt{case}\, G(u,y) = 0. & \top\\ \texttt{case}\, G(u,y) = 1. & \beta_C(fu, y)\end{array}\right)\\ \texttt{case}\, F(u,y) \in 1. & \beta_C(fu, y)\end{array}\right)\end{array}$$

$\mathbf{HA}^{\omega}_{+} \vdash (3 \to 4)^C$: We define $K(f,F,G,u,y) = u,y$ and $k(f,F,G) = (h,H)$ where $h = f$ and

$$H(u,y) = \begin{cases} F(u,y) & \text{if} \quad F(u,y) \in X \text{ and } G(u,y) = 0 \\ * \in 1 & \text{otherwise.} \end{cases}$$

$\mathbf{HA}^{\omega}_{+} \vdash (4 \to 3)^C$: Define $J(h,H,u,y) = u,y$ and $j(h,H) = (f,F,G)$ where $f = h, F = H$ and

$$G(u,y) = \begin{cases} 0 & \text{if} \quad F(u,y) \in X \\ 1 & \text{if} \quad F(u,y) \in 1. \end{cases}$$

$\square$

4 Which Principles are Validated

In this section we show some important non-constructive principles which are *validated* by the Copenhagen interpretation (then they are said to be *C-interpreted*). In the next section we will use this to give an *axiomatization* (sometimes called a characterization) of the Copenhagen interpretation.

Definition 4.1. *A formula ϕ in the language of $\mathbf{HA}^{\omega}_{+}$ is said to be* validated *by the Copenhagen interpretation or* C-interpreted *if $\mathbf{HA}^{\omega}_{+} \vdash \phi^C$.*

Independence of Premiss

$$\text{IP} \qquad (\forall y\phi \to \exists u \forall v \psi) \to \exists u(\forall y \phi \to \forall v \psi)$$

where ϕ, ψ are of the form $\phi = \phi_C$ and $\psi = \psi_C$. First consider the premiss and conclusion of the outer $\to$:

$$\begin{aligned}&(\forall y\phi(y) \to \exists u\forall v\psi(u,v))^C = \\ &\exists (f,F) : \square = \{(f,F) : (1 \Rightarrow U) \times (V \Rightarrow Y+1) \mid \forall v \in V.\mathsf{case}\, Fv \in Y.\phi(Fv) \to \psi(f(*),v)\} \\ &\forall v.\alpha(f,F,v)\end{aligned}$$

where

$$\alpha(f,F,v) = \begin{pmatrix} \mathsf{case}\, Fv \in Y. & \top \\ \mathsf{case}\, Fv \in 1. & \psi(f(*),v) \end{pmatrix}$$

And

$$\begin{aligned}&(\exists u(\forall y\phi \to \forall v\psi))^C = \\ &\exists u : U, G : \dagger = \{V \Rightarrow Y+1 \mid \forall v.\mathsf{case}\, GV \in Y.\phi(Gv) \to \psi(u,v)\} \\ &\forall v.\beta(u,G,v)\end{aligned}$$

where

$$\beta(u,G,v) = \begin{pmatrix} \mathsf{case}\, Gv \in Y. & \top \\ \mathsf{case}\, Gv \in 1. & \psi(u,v) \end{pmatrix}$$

Now we can interpret IP:

$$\begin{aligned}&(\text{IP})^C = \\ &\exists (h,H) \quad : \{(h,H) : (\square \Rightarrow U \times \dagger) \times (\square \times V \Rightarrow V+1) \mid \\ &\qquad \forall (f,F), v : \square \times V.\mathsf{case}\, H(f,F,v) \in V.\alpha((f,F),H(f,F,v)) \to \beta(h(f,F),v)\} \\ &\forall (f,F), v : \square \times V. \begin{pmatrix} \mathsf{case}\, H(f,F,v) \in V. & \top \\ \mathsf{case}\, H(f,F,v) \in 1. & \beta(h(f,F),v) \end{pmatrix}\end{aligned}$$

This is realized by

$$h(f, F) = (f(*), F), \qquad H(f, F, v) = v.$$

Axiom of Choice

$$\text{AC} \qquad (\forall x \exists y \phi(x, y) \to \exists F \forall x \phi(x, Fx))$$

where again we assume that ϕ is of the form $\phi = \phi_C$.

$$(\text{AC})^C = (\exists F \forall x \phi(x, Fx)^C \to \exists F \forall x.\phi(x, Fx)^C)^C$$

which obviously holds.

A Generalized Markov Principle $\text{MP}_{\mathbf{Dia}}$ is also validated by the C-interpretation.

$$\text{MP}_C \qquad (\forall \phi(y) \to \psi) \to \exists y(\phi(y) \to \psi)$$

where ϕ, ψ are of the form $\phi = \phi_C$ and $\psi = \psi_C$. First consider

$$(\forall y \phi(y) \to \psi)^C = \exists z : \Box = \{z : Y + 1 \mid \texttt{case}\, z \in Y.\phi(z) \to \psi\}.\alpha(z)$$

where

$$\alpha(z) = \begin{pmatrix} \texttt{case}\, z \in Y. & \top \\ \texttt{case}\, z \in 1. & \psi \end{pmatrix}$$

And

$$\begin{array}{l} ((\forall y \phi(y) \to \psi) \to \exists(\phi(y) \to \psi))^C = \\ \exists(f, F) : \{(f, F) : (\Box \to Y) \times (\Box \to 2) \mid \forall z : \Box.Fz = 0.\alpha(z) \to \phi(fz) \to \psi\} \\ \forall z : \Box. \begin{pmatrix} \texttt{case}\, Fz = 0. & \top \\ \texttt{case}\, Fz = 1. & \phi(fz) \to \psi \end{pmatrix} \end{array}$$

This is realized by

$$fz = \begin{cases} z & \text{if} \quad z \in Y \\ 0 & \text{if} \quad z \in 1 \end{cases}, \qquad Fz = 0,$$

since, if $z \in 1$, we have $\alpha(z) = \psi$ and $\psi \to \phi(0) \to \psi$, and if $z \in Y$, then because $z : \Box$, we have $\phi(z) \to \psi$ and $\alpha(z) = \top$, so $\alpha(z) \to \phi(fz) \to \psi$.

5 Axiomatization

The axiomatization Theorem is our second main result. It states under certain non-constructive principles (but still in a system much more constructive than classical logic) any formula ϕ of $\mathbf{HA}^\omega_+$ is provably equivalent to it's C-interpretation ϕ^C. Moreover, ϕ is provable in the stronger system if and only if ϕ^C is provable in $\mathbf{HA}^\omega_+$. The combination of the two results of the axiomatization Theorem tells us that if a formula ϕ then we can find a realizer for the equivalent formula ϕ^C.

We have the following *axiomatization* of the functional interpretation C:

Theorem 5.1 (Axiomatization). *Let $\mathcal{X} = \{\text{MP}_{\mathbf{Dia}}, \text{AC}, \text{IP}\}$, then*

1. *$\mathbf{HA}^\omega_+ + \mathcal{X} \vdash \phi \leftrightarrow \phi^C$ for all formulas ϕ of the language of $\mathbf{HA}^\omega_+$.*

2. $\boldsymbol{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi$ *iff* $\boldsymbol{HA}^{\omega}_{+} \vdash \phi^C$.

Proof: We show 1 by induction on formulas. For $\phi \in \{\alpha \text{ atomic }, (\alpha \wedge \beta), \alpha \vee \beta, \forall z.\alpha, \exists z.\alpha\}$ it is not hard to see that $\mathbf{HA}^{\omega}_{+} \vdash \phi \leftrightarrow \phi^C$. For example, let us show that

$$\mathbf{HA}^{\omega}_{+} \vdash \alpha \wedge \beta \leftrightarrow (\alpha \wedge \beta)^C.$$

By induction we have $\mathbf{HA}^{\omega}_{+} \vdash \alpha \leftrightarrow \alpha^C$ and $\mathbf{HA}^{\omega}_{+} \vdash \beta \leftrightarrow \beta^C$, so $\mathbf{HA}^{\omega}_{+} \vdash \alpha \wedge \beta \leftrightarrow \alpha^C \wedge \beta^C$. And clearly

$$\begin{aligned} \mathbf{HA}^{\omega}_{+} \ \vdash \ & \exists u, v \forall \alpha_C(u, x) \wedge \beta_C(v, y) \\ & \leftrightarrow \exists u, v \forall z : X + Y. \begin{pmatrix} \texttt{case} & z \in x. & \alpha_C(u, z), \\ \texttt{case} & z \in Y. & \beta_C(v, z) \end{pmatrix} \end{aligned}$$

For $\phi = \begin{pmatrix} \texttt{case} & z \in W. & \alpha(z), \\ \texttt{case} & z \in R. & \beta(z) \end{pmatrix}^C$ we must use the principle IP as well:

$$\begin{array}{ll}
\begin{pmatrix} \texttt{case} & z \in W. & \alpha(z), \\ \texttt{case} & z \in R. & \beta(z) \end{pmatrix} & \equiv \\
\begin{pmatrix} \texttt{case} & z \in W. & \exists u \forall x. \alpha_C(z, u, x), \\ \texttt{case} & z \in R. & \exists v \forall y. \beta_C(z, v, y) \end{pmatrix} & \equiv \\
(z \in W \to \exists u \forall x. \alpha_C) \wedge (z \in R \to \exists v \forall y. \beta_C) & \equiv \text{IP} \\
\exists u (z \in W \to \forall x. \alpha_C) \wedge \exists v (z \in R \to \forall y. \beta_C) & \equiv \\
\exists u \forall x (z \in W \to \alpha_C) \wedge \exists v \forall y (z \in R \to \beta_C) & \equiv \\
\exists u, v \forall x, y. (z \in W \to \alpha_C) \wedge (z \in R \to \beta_C) & \equiv \\
\exists u, v \forall x, y. \begin{pmatrix} \texttt{case} & z \in W. & \alpha_C(z, u, x), \\ \texttt{case} & z \in R. & \beta_C(z, v, y) \end{pmatrix} &
\end{array}$$

And for $\phi = \alpha \to \beta$ we have the following string of equivalences:

$$\begin{array}{lll}
\exists u \forall x \alpha_C(u, x) \to \exists v \forall y. \beta_C(v, y) & \equiv & \mathbf{HA}^{\omega}_{+} \\
\forall u (\forall x \alpha_C(u, x) \to \exists v \forall y. \beta_C(v, y)) & \equiv & \text{IP} \\
\forall u \exists v (\forall x \alpha_C(u, x) \to \forall y. \beta_C(v, y)) & \equiv & \mathbf{HA}^{\omega}_{+} \\
\forall u \exists v \forall y (\forall x \alpha_C(u, x) \to \beta_C(v, y)) & \equiv & \text{MP}_{\mathbf{Dia}} \\
\forall u \exists v \forall y \exists x (\alpha_C(u, x) \to \beta_C(v, y)) & \equiv & \text{AC} \\
\exists f : V^U, F : X^{U \times Y} \forall u, y. \alpha_C(u, F(u, y)) \to \beta_C(f(u), y) & \equiv & \mathbf{HA}^{\omega}_{+} \\
\exists f : V^U, F' : (X+1)^{U \times Y} \forall u, y. \begin{pmatrix} \texttt{case}\, F(u, y) \in X. & \alpha_C(u, F(u, y)) \to \beta_C(f(u), y) \\ \texttt{case}\, F(u, y) \in 1. & \beta_C(fu, y) \end{pmatrix} & \equiv & \mathbf{HA}^{\omega}_{+}
\end{array}$$

$$\exists (f, F) : \{ f : V^U, F : (X+1)^{U \times Y} \mid \forall u, y. \texttt{case}\, F(u, y) \in X. \alpha_C(u, F(u, y)) \to \beta_C(f(u), y) \}$$
$$\forall u, y. \begin{pmatrix} \texttt{case}\, F(u, y) \in X. & \top \\ \texttt{case}\, F(u, y) \in 1. & \beta_C(fu, y) \end{pmatrix}$$

Where the last two equivalences follows from 3.1.

In the previous section we showed that $\mathbf{HA}^{\omega}_{+} \vdash (\text{MP}_{\mathbf{Dia}})^C, (\text{IP})^C, (\text{AC})^C$, and by the Soundness Theorem 3.2, we have $\mathbf{HA}^{\omega}_{+} \vdash \phi$ implies $\mathbf{HA}^{\omega}_{+} \vdash \phi^C$, so it follows that

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi \Rightarrow \mathbf{HA}^{\omega}_{+} \vdash \phi^C$$

To see that

$$\mathbf{HA}^{\omega}_{+} \vdash \phi^C \Rightarrow \mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi,$$

assume $\mathbf{HA}^{\omega}_{+} \vdash \phi^C$ then also $\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi^C$, so together with 1 of Theorem 5.1 we get $\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi$. □

5.1 Copenhagen Interpretation Generalizes Dialectica Interpretation

We now show that the C-interpretation is a generalization of the D-interpretation in the sense that if require atomic formulas to be recursively decidable, then they have the same axiomatization. Moreover, ϕ^C and ϕ^D are both C- and D-equivalent, i.e., within the interpretations we can't tell the difference, assuming atomic formulas are recursively decidable.

Lemma 5.2.

$$\boldsymbol{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi^D \leftrightarrow \phi^C$$

for all formulas ϕ in the language of $\boldsymbol{HA}^{\omega}_{+}$.

Proof: We have

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi \leftrightarrow \phi^C$$

by Theorem 5.1. And

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi \leftrightarrow \phi^D.$$

□

Theorem 5.3. *Assuming that atomic formulas are recursively decidable, the D- and C-interpretations have the same axiomatization.*

Proof: First we must expand the D-interpretation to $\mathbf{HA}^{\omega}_{+}$ by adding one clause, namely:

$$\begin{pmatrix} \texttt{case} & z \in W. & \alpha(z) \\ \texttt{case} & z \in R. & \beta(z) \end{pmatrix}^D = \exists u, v \forall x, y. \begin{pmatrix} \texttt{case} & z \in W. & \alpha_D(z,u,x) \\ \texttt{case} & z \in R. & \beta_D(z,v,y) \end{pmatrix}$$

We have (by axiomatization of Dialectica):

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi \leftrightarrow \phi^D$$

and by soundness of D, under the assumption that atomic formulas are recursive,

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash \phi \Rightarrow \mathbf{HA}^{\omega}_{+} \vdash \phi^D.$$

Notice that soundness of D usually is shown only for the system $\mathbf{HA}^{\omega} + \mathcal{X}$, i.e., without the rules (12) and (13) and it only holds under the assumption that atomic formulas are recursively decidable. Rule (12) is validated by D iff atomic formulas are recursive.

□

This theorem tells us that in the system $\mathbf{HA}^{\omega}_{+}$ we might as well define the Dialectica interpretation (still under assumption that atomic formulas are recursive) as the C-interpretation, since it has the same axiomatization.

Corollary 5.4. *1.*

$$\boldsymbol{HA}^{\omega}_{+} \vdash (\phi^D \leftrightarrow \phi^C)^D$$

when atomic formulas are recursively decidable.

2.

$$\boldsymbol{HA}^{\omega}_{+} \vdash (\phi^D \leftrightarrow \phi^C)^C.$$

Proof: By Theorem 5.3

$$\mathbf{HA}^{\omega}_{+} + \mathrm{IP} + \mathrm{AC} + \mathrm{MP}_{\mathbf{Dia}} \vdash \phi^D \leftrightarrow \phi^C$$

so it follows from soundness of the two functional interpretations that

$$\mathbf{HA}^{\omega}_{+} \vdash (\phi^D \leftrightarrow \phi^C)^D \quad \text{and} \quad \mathbf{HA}^{\omega}_{+} \vdash (\phi^D \leftrightarrow \phi^C)^C$$

where we recall that soundness of D-interpretation only holds under the assumption that all atomic formulas are recursively decidable. □

6 A Classical Version of the Copenhagen Interpretation

A classical version of the C-interpretation is the double negation translation followed by the C-interpretation and then re-arranged in a way so that it becomes more simple. The classical version that we give here builds on a classical version of the D-interpretation given in [SK07]. We admit that not all of the clauses look simple, but others really are simplifications of double negation followed by C-interpretation.

We first define a negative translation $(-)'$, which was presented in [SR98] and in [SK07].

Definition 6.1. *The negative translation $A' = \neg A^*$, where A^* is defined inductively as follows:*

$$\begin{array}{lcl} P^* & = & \neg P \quad \textit{if } P \textit{ is prime} \\ (\neg A)^* & = & \neg A^* \\ (A \vee B)^* & = & A^* \wedge B^* \\ (\forall x A)^* & = & \exists x A^* \\ (A \to B)^* & = & A' \wedge B^* \\ (\exists x A)^* & = & \neg \exists x \neg A^* \\ (A \wedge B)^* & = & A^* \vee B^* \end{array}$$

Notice that the last three clauses can be defined by the others since the source is classical (**PA**). Shoenfield introduced a functional interpretation for Peano arithmetic **PA**, associating to every formula A a formula $A^S = \forall u \exists x. A_S(u,x)$, with A_S quantifier-free. Shoenfield's interpretation is defined inductively as follows:

Definition 6.2.

$$\begin{array}{lcl} P^S & = & P = P_S \quad \textit{for } P \textit{ atomic} \\ (\neg A)^S & = & \forall f \exists u. \neg A_S(u, fu) \\ (A \vee B)^S & = & \forall u, v \exists x, y A_S(u,x) \vee B_S(v,y) \\ (\forall z. A)^S & = & \forall z, u \exists x. A_S(z,u,x) \\ (A \to B)^S & = & \forall f, v \exists u, y A_S(u, f(u)) \to B_S(v,y) \\ (\exists z A)^S & = & \forall U \exists z, f. A_S(z, U(z,f), f(U(z,f))) \\ (A \wedge B)^S & = & \forall z : U + V \exists x, y. \begin{pmatrix} \mathsf{case}\, z \in U. & A_S(z,x) \\ \mathsf{case}\, z \in V. & B_S(z,y) \end{pmatrix} \end{array}$$

Let $(-)'$ be the negative translation defined in [SK07], then

17

Theorem 6.3. *$\boldsymbol{HA}^{\omega}_{+} + \mathcal{X} \vdash (A')^C \leftrightarrow (A^S)^C$, and $\boldsymbol{PA} \vdash A \Rightarrow \boldsymbol{HA}^{\omega}_{+} \vdash (A^S)^C$.*

Proof: From [SK07] we know that $(A')^D = \exists f \forall u A'_D(f,u)$, where $A'_D(f,u) \leftrightarrow A_S(u, f(u))$ in $\mathbf{HA}^{\omega}$. So

$$\mathbf{HA}^{\omega} \vdash (A')^D \leftrightarrow \exists f \forall u A_S(u, f(u))$$

Notice that $\exists f \forall u A_S(u, f(u)) = (A^S)^D$, so in fact

$$\mathbf{HA}^{\omega} \vdash (A')^D \leftrightarrow (A^S)^D$$

By Lemma 5.2 we have for all formulas A of $\mathbf{HA}^{\omega}_{+}$,

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash A^D \leftrightarrow A^C$$

so

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash (A')^C \leftrightarrow (A')^D \quad \text{and} \quad \mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash (A^S)^C \leftrightarrow (A^S)^D$$

it follows that

$$\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash (A^S)^C \leftrightarrow (A')^C$$

and $(A^S)^C$ is of course C-stable. Now from [SK07] we know that [5]

$$\mathbf{PA} \vdash A \Rightarrow \mathbf{HA}^{\omega} \vdash (A^S)^D$$

hence, by the above equivalences, $\mathbf{HA}^{\omega}_{+} + \mathcal{X} \vdash (A^S)^C$ so it follows from Theorem 5.1 that

$$\mathbf{PA} \vdash A \Rightarrow \mathbf{HA}^{\omega}_{+} \vdash (A^S)^C$$

because $((A^S)^C)^C = (A^S)^C$. $\square$

7 Acknowledgement

As mentioned in the introduction, this work relies heavily on results in [BBLBCB07] and discussions at the two "Dialectica meetings" in Copenhagen and in Genoa. The author therefore owes many, many thanks to (in alphabetic order) Lars Birkedal, Carsten Butz, Martin Hyland, Jaap van Oosten, Pino Rosolini and Thomas Streicher for encouragement, support, answering and posing of questions, endless discussions and very pleasant company. I would also like to thank Ulrich Kohlenbach for pointing out the example in Section 2.2, and Hongseok Yang for many good suggestions and discussions, and Paulo Oliva for finding a severe bug in a previous version of the paper.

A Sketch of a Simple, Higher Typed Variant of Dialectica

Just to indicate the ideas, we now give a rough sketch of a simpler variant of Dialectica, which soundly interprets $\mathbf{HA}^{\omega}$. The details will be worked out in a future paper.

In the setting of Definition 2.1, if we allow only closed atomic formulas, e.g., $s =_U t$ with s, t closed terms, then we may give the following higher typed variant of Dialectica:

[5]notice that to show this result one uses the fact that atomic formulas of **PA** are recursive, since atomic formulas of **PA** are equality between terms of type N.

Definition A.1. *Suppose α and β are formulas of $\boldsymbol{HA}^\omega$ and $\alpha^C = \exists u \forall x.\alpha_C(u,x)$ and $\beta^C = \exists v \forall y.\beta_C(v,y)$.*

$$\begin{array}{lcl}
\alpha \in \{\top, \bot\}, & & \alpha^C = \alpha_C = \alpha. \\
\alpha \text{ atomic, } \alpha^C & = & \exists u : 1 + \{* : 1 \mid \alpha\}. \begin{pmatrix} \texttt{case} & u = \texttt{inl}(*). & \bot \\ \texttt{case} & u = \texttt{inr}(*). & \top \end{pmatrix} \\
(\alpha \wedge \beta)^C & = & \exists u, v \forall x, y.\alpha_C(u,x) \wedge \beta_C(v,y) \\
(\alpha \to \beta)^C & = & \exists f : V^U,\ F : X^{U \times Y}.\forall u, y.\alpha_C(u, F(u,y)) \to \beta_C(fu, y) \\
(\forall z.\alpha(z))^C & = & \exists f : Z \to U \forall z, x.\alpha_C(z, f(z), x) \\
(\exists z.\alpha(z))^C & = & \exists z, u \forall x.\alpha_C(z,u,x) \\
(\alpha \vee \beta)^C & = & \exists z \in U + V \forall x, y. \begin{pmatrix} \texttt{case} & z \in U. & \alpha_C(z,x) \\ \texttt{case} & z \in V. & \beta_C(z,y) \end{pmatrix} \\
\begin{pmatrix} \texttt{case} & z \in W. & \alpha(z) \\ \texttt{case} & z \in R. & \beta(z) \end{pmatrix}^C & = & \exists u, v \forall x, y. \begin{pmatrix} \texttt{case} & z \in W. & \alpha_C(z,u,x) \\ \texttt{case} & z \in R. & \beta_C(z,v,y) \end{pmatrix}
\end{array}$$

That is, it works precisely like the original Dialectica interpretation except for the interpretation of atomic formulas, and the use of subset types. Notice that the quantifier-free part α_C is recursive by construction! This version of Dialectica works without the assumption of decidable atomic formulas, hence equality at higher types does not have to be intensional.

If we want to allow open atomic formulas $P(x)$ as well, types will become dependent, e.g., $\{x : X \mid P(x)\}$ depends on the free variable $x : X$, so if $\alpha(z)$ has a free variable $z : Z$, then

$$(\alpha(z))^C = \exists u : U(z) \forall x : X(z).\alpha_C(z,u,x)$$

and

$$(\forall z.\alpha(z))^C = \exists f : Z \to U(z).\forall z : Z, x : X(z).\alpha_C(z, f(z), x).$$

References

[AF98] Jeremy Avigad and Solomon Feferman. Gödel's functional ("Dialectica") interpretation. In *Handbook of proof theory*, volume 137 of *Stud. Logic Found. Math.*, pages 337–405. North-Holland, Amsterdam, 1998.

[BBLBCB07] J.M.E. Hyland & J. van Oosten & G. Rosolini & T. Streicher B. Biering & L. Birkedal & C. Butz. Topos theoretic versions of Dialectica interpretations. 2007. Unpublished draft.

[Bie07] Bodil Biering. Cartesian closed Dialectica categories. 2007. Submitted.

[DN74] Justus Diller and Werner Nahm. Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen. *Arch. Math. Logik Grundlagenforsch.*, 16:49–66, 1974.

[dP89] V. C. V. de Paiva. The Dialectica categories. In *Categories in computer science and logic (Boulder, CO, 1987)*, volume 92 of *Contemp. Math.*, pages 47–62. Amer. Math. Soc., Providence, RI, 1989.

[Göd90] Kurt Gödel. *Collected works. Vol. II.* The Clarendon Press Oxford University Press, New York, 1990. Publications 1938–1974, Edited and with a preface by Solomon Feferman, pages 217–251.

[Hyl02] J. M. E. Hyland. Proof theory in the abstract. *Ann. Pure Appl. Logic*, 114(1-3):43–78, 2002. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

[Sch06] Helmut Schwichtenberg. Proof theory. *lecture notes*, 2006.

[SK07] Thomas Streicher and Ulrich Kohlenbach. Shoenfield is Gödel after Krivine. *MLQ Math. Log. Q.*, 53(2):176–179, 2007.

[SR98] Th. Streicher and B. Reus. Classical logic, continuation semantics and abstract machines. *J. Funct. Programming*, 8(6):543–572, 1998.

[Str01] Thomas Streicher. Introduction to constructive logic and mathematics. *lecture notes*, 2000/2001.

Chapter 6

BI Hyperdoctrines and Higher Order Separation Logic

The following paper was published in ACM Transactions on Programming Languages and Systems, Volume 29, Issue 5, Article 24 (2007), Special Issue ESOP'05. Sections 5,6 and 7 were written by Birkedal and Torp-Smith. Basic knowledge of tripos theory or hyperdoctrines is assumed, see e.g. [Pit02]. An introduction to separation logic can be found in [Rey02].

The main results in this paper are: The simple definition of predicate BI (logic of bunched implications). The introduction of BI hyperdoctrines as sound and complete models for first- and higher-order predicate BI. A precise correspondence between separation logic and predicate BI, which leads to an easy extension of first-order separation logic to higher-order separation logic. The result that any BI hyperdoctrine which satisfies the rules for full subset types is trivial. This rules out all topos models (since the subobject fibration has full subset types).

The Dialectica tripos is an instance of a BI-hyperdoctrine, hence it is a model of separation logic, which is very different from the standard models of separation logic. Following the idea that the structure of the tripos contains an implicit soundness proof, it should be possible to give a Dialectica or Copenhagen interpretation of separation logic, thus counter examples from Heyting arithmetic would (by the soundness theorem) yield counter examples for separation logic.

References

[Pit02] Andrew M. Pitts. Tripos theory in retrospect. *Math. Structures Comput. Sci.*, 12(3):265–279, 2002. Realizability (Trento, 1999).

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. 2002.

BI-Hyperdoctrines, Higher-Order Separation Logic, and Abstraction

BODIL BIERING, LARS BIRKEDAL, and NOAH TORP-SMITH
IT University of Copenhagen

We present a precise correspondence between separation logic and a simple notion of *predicate* BI, extending the earlier correspondence given between part of separation logic and *propositional* BI. Moreover, we introduce the notion of a BI hyperdoctrine, show that it soundly models classical and intuitionistic first- and higher-order predicate BI, and use it to show that we may easily extend separation logic to *higher-order*. We also demonstrate that this extension is important for program proving, since it provides sound reasoning principles for data abstraction in the presence of aliasing.

Categories and Subject Descriptors: D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Documentation*; F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*Assertions, logics of programs, specification techniques*

General Terms: Reliability, Theory, Verification

Additional Key Words and Phrases: Separation logic, hyperdoctrines, abstraction

ACM Reference Format:

Biering, B., Birkedal, L., and Torp-Smith, N. 2007. BI-Hyperdoctrines, higher-order separation logic, and abstraction. ACM Trans. Program. Lang. Syst. 29, 5, Article 24 (August 2007), 34 pages. DOI = 10.1145/1275497.1275499 http://doi.acm.org/10.1145/1275497.1275499

1. INTRODUCTION

Variants of the recent formalism of *separation logic* [Reynolds 2002; Ishtiaq and O'Hearn 2001] have been used to prove correct many interesting algorithms involving pointers, both in sequential and concurrent settings [O'Hearn 2004; Yang 2001; Birkedal et al. 2004]. Separation logic is a Hoare-style

An extended abstract of the present article appeared in the 2005 *Proceedings of the European Symposium on Programming (ESOP)*. The research of L. Birkedal and N. Torp-Smith was partially supported by Danish Natural Science Research Council Grant 51-00-0315 and Danish Technical Research Council Grant 56-00-0309.
Authors' addresses: B. Biering, L. Birkedal (contact author), N. Torp-Smith, Department of Theoretical Computer Science, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark; email: birkedal@itu.dk.

program logic, and its main advantage over traditional program logics is that it facilitates modular reasoning, that is, *local reasoning*, about programs with shared mutable data. Different extensions of core separation logic [Reynolds 2002] have been used to prove correct various algorithms. For example, Yang [2001] extended the core logic with lists and trees and in Birkedal et al. [2004] the logic was extended with finite sets and relations. Thus, it is natural to ask whether one has to make a new extension of separation logic for every proof one wants to make. This would be unfortunate for formal verification of proofs in separation logic, since it would make the enterprise of formal verification burdensome and dubious. We argue in this article that there is a natural single underlying logic in which it is possible to *define* the various extensions and prove the expected properties thereof; this is then the single logic that should be employed for formal verification.

Part of the pointer model of separation logic, namely, that given by heaps (but not stacks, i.e., local variables), has been related to *propositional* BI, the logic of bunched implications introduced by O'Hearn and Pym [1999]. In this article we show how the correspondence may be extended to a precise correspondence between all of the pointer model (including stacks) and a simple notion of *predicate* BI. We introduce the notion of a *BI hyperdoctrine*, a simple extension of Lawvere's notion of hyperdoctrine [Lawvere 1969], and show that it soundly models predicate BI. The notion of predicate BI we consider is different from the one studied in Pym [2004, 2002], which has a bunched structure on variable contexts. However, we believe that our notion of predicate BI with its class of BI hyperdoctrine models is the right one for separation logic (Pym aimed to model mulitiplicative quantifiers; separation logic only uses additive quantifiers). To make this point, we show that the pointer model of separation logic exactly corresponds to the interpretation of predicate BI in a simple BI hyperdoctrine. This correspondence also allows us to see that it is simple to extend separation logic to *higher-order* separation logic. Now we briefly explain this extension and outline why it is important for program proving.

The force of separation logic comes from both its language of assertions—which is a variant of propositional BI [Pym 2002]—and its language of specifications, or Hoare triples. In the present work, we extend both of these. First, we introduce an assertion language which is a variant of *higher-order predicate* BI. The extension from the traditional assertion language of separation logic simply allows function types, has a type Prop of proposition, and allows quantification over variables of all types. Thus, the assertion language is higher order in the usual sense that it allows quantification over predicates. Next, we present a specification logic for a simple second-order programming language. We provide models for both the new assertion language and the specification logic, and provide inference rules for deriving valid specifications. As it turns out, it is technically straightforward to do so; this emphasizes that our notion of higher-order predicate BI is the correct one for separation logic.

Next we consider the expressiveness of higher-order separation logic and argue, with the use of examples, that it is quite expressive. In particular, we show that higher-order separation logic can be used in a natural way to model data abstraction via existential quantification over predicates corresponding to abstract resource invariants. The main formal rule in this development is

$$\frac{\begin{array}{c}\Delta \vdash \hat{P}{:}\tau \\ \Delta, \vec{x_1}; \Gamma \vdash \{P_1[\hat{P}/x]\}\ c_1\ \{Q_1[\hat{P}/x]\} \\ \vdots \\ \Delta, \vec{x_n}; \Gamma \vdash \{P_n[\hat{P}/x]\}\ c_n\ \{Q_n[\hat{P}/x]\} \\ \Delta; \Gamma, \exists x{:}\tau.(\{P_1\}k_1(\vec{x_1})\{Q_1\} \wedge \cdots \wedge \{P_n\}k_n(\vec{x_n})\{Q_n\}) \vdash \{P\}\ c\ \{Q\}\end{array}}{\Delta; \Gamma \vdash \{P\}\ \mathbf{let}\ k_1(\vec{x}_1) = c_1, \ldots, k_n(\vec{x}_n) = c_n\ \mathbf{in}\ c\ \mathbf{end}\ \{Q\}}\ x \notin \mathrm{FV}(\{P\}\ c\ \{Q\}).$$

Here one may think of x as a predicate describing a resource invariant used by an abstract data type with operations $k_1, \ldots, k_n$. If a client c has then been proved correct under the assumption that such a predicate exists, it is possible to use the client with any concrete resource invariant $\hat{P}$ and implementations $c_1, \ldots, c_n$.

Moreover, we show that, using universal quantification over predicates, we can prove correct polymorphic operations on polymorphic data types, for example, reversing a list of elements described by an arbitrary predicate. For this to be useful, however, it is clear that a higher-order programming language would be preferable (such that one could program many more useful polymorphic operations, e.g., the **map** function for lists); we have chosen to stick with the simpler second-order language here to communicate more easily the ideas of higher-order separation logic.

Before proceeding with the technical development, we give an intuitive justification of the use of BI hyperdoctrines to model higher-order predicate BI. A powerful way of obtaining models of BI is by means of functor categories (presheaves), using Day's construction to obtain a doubly closed structure on the functor category [Pym et al. 2004]. Such functor categories can be used to model propositional BI in two different senses: In the first sense, one models *provability*, that is, entailment between propositions, and it works because the lattice of subobjects of the terminal object in such functor categories forms a BI algebra (a doubly Cartesian closed preorder). In the second sense, one models *proofs*, and this works because the whole functor category is doubly Cartesian closed. Here we seek models of provability of predicate BI. Since the considered functor categories are toposes and hence model higher-order predicate logic, one might think that a straightforward extension is possible. But, alas, it is not the case. In general, for this to work, *every* lattice of subobjects (for any object, not only the terminal object) should be a BI algebra and, moreover, to model substitution correctly, the BI algebra structure should be preserved by pulling back along any morphism. We show this can only be the case if the BI algebra structure is trivial, that is, coincides with the Cartesian structure (see Theorem 2.7). Our theorem holds for any topos, not just for the functor categories just mentioned. Hence, we need to consider a wider class of models for predicate BI than just toposes and this justifies the notion of a BI hyperdoctrine. The intuitive reason at BI hyperdoctrines work is that predicates are not required to be modeled by subobjects, but they can be something more general.

Another important point of BI hyperdoctrines is that they are easy to come by: Given any complete BI algebra B, there is a canonical BI hyperdoctrine in which predicates are modeled as B-valued functions; this is explained in detail in Example 2.6.

The rest of the article is organized as follows. In Section 2, we first recall Lawvere's notion of a *hyperdoctrine* [Lawvere 1969] and briefly recall how it can be used to model intuitionistic and classical first- and higher-order predicate logic. More details about this can be found in the handbook chapter [Pitts 2001] and the book [Jacobs 1999]. We then introduce the concept of a BI hyperdoctrine and show that it models BI. In Section 3, we show that the standard pointer model of BI is an instance of our class of models. The new class of models provides a straightforward way to give semantics to a higher-order extension of BI, and we discuss ramifications of this extension for separation logic in Section 4. In Section 5, we introduce the programming language considered in this work. It is a simple extension of the standard programming language of separation logic with simple procedures and calls to these. We use the higher-order logic just introduced to give a specification logic for the programming language. In Section 6, we present examples which illustrate how this specification logic can be used to reason about data abstraction, using existential quantification over predicates. In Section 7 we present some simple applications of universal quantification over predicates in program proving. In the last section we discuss related and future work.

This article is the full version of an extended abstract presented at the ESOP 2005 conference. Compared to the conference version, this work includes more detailed proofs and a much more extensive discussion of applications of higher-order separation logic in program proving, in particular for data abstraction.

2. BI HYPERDOCTRINES

We first introduce Lawvere's notion of a hyperdoctrine [Lawvere 1969] and briefly recall how it can be used to model intuitionistic and classical first- and higher-order predicate logic (see, e.g., the handbook chapter [Pitts 2001] and book [Jacobs 1999] for more explanations). We then define the notion of a BI hyperdoctrine, which is a straightforward extension of the standard notion of hyperdoctrine, and explain how it can be used to model predicate BI logic.

2.1 Hyperdoctrines

A first-order hyperdoctrine is a categorical structure tailored to model first-order predicate logic with equality. The structure has a base category $\mathcal{C}$ for modeling the types and terms, and a $\mathcal{C}$-indexed category $\mathcal{P}$ for modeling formulas. Recall that a Heyting algebra is a bi-Cartesian closed partial order, that is, a partial order which, when considered as a category, is Cartesian closed ($\top$, $\wedge$, $\to$) and has finite coproducts ($\bot$, $\vee$).

Definition 2.1. Let $\mathcal{C}$ be a category with finite products. A *first-order hyperdoctrine* $\mathcal{P}$ over $\mathcal{C}$ is a contravariant functor $\mathcal{P} : \mathcal{C}^{op} \to \mathbf{Poset}$ from $\mathcal{C}$ into the category of partially ordered sets and monotone functions, and has the following properties.

(1) For each object X, the partially ordered set $\mathcal{P}(X)$ is a Heyting algebra.
(2) For each morphism $f : X \to Y$ in $\mathcal{C}$, the monotone function $\mathcal{P}(f) : \mathcal{P}(Y) \to \mathcal{P}(X)$ is a Heyting algebra homomorphism.
(3) For each diagonal morphism $\Delta_X : X \to X \times X$ in $\mathcal{C}$, the left adjoint to $\mathcal{P}(\Delta_X)$ at the top element $\top \in \mathcal{P}(X)$ exists. In other words, there is an element $=_X$ of $\mathcal{P}(X \times X)$ satisfying that for all $A \in \mathcal{P}(X \times X)$,

$$\top \leq \mathcal{P}(\Delta_X)(A) \quad \text{iff} \quad =_X \leq A.$$

(4) For each product projection $\pi : \Gamma \times X \to \Gamma$ in $\mathcal{C}$, the monotone function $\mathcal{P}(\pi) : \mathcal{P}(\Gamma) \to \mathcal{P}(\Gamma \times X)$ has both a left adjoint $(\exists X)_\Gamma$ and a right adjoint $(\forall X)_\Gamma$.

$$A \leq \mathcal{P}(\pi)(A') \quad \text{if and only if} \quad (\exists X)_\Gamma(A) \leq A'$$
$$\mathcal{P}(\pi)(A') \leq A \quad \text{if and only if} \quad A' \leq (\forall X)_\Gamma(A)$$

Moreover, these adjoints are natural in Γ, that is, given $s : \Gamma \to \Gamma'$ in $\mathcal{C}$,

$$\begin{tikzcd} \mathcal{P}(\Gamma' \times X) \arrow[r, "\mathcal{P}(s\times id_X)"] \arrow[d, "(\exists X)_{\Gamma'}"'] & \mathcal{P}(\Gamma \times X) \arrow[d, "(\exists X)_\Gamma"] \\ \mathcal{P}(\Gamma') \arrow[r, "\mathcal{P}(s)"'] & \mathcal{P}(\Gamma) \end{tikzcd} \qquad \begin{tikzcd} \mathcal{P}(\Gamma' \times X) \arrow[r, "\mathcal{P}(s\times id_X)"] \arrow[d, "(\forall X)_{\Gamma'}"'] & \mathcal{P}(\Gamma \times X) \arrow[d, "(\forall X)_\Gamma"] \\ \mathcal{P}(\Gamma') \arrow[r, "\mathcal{P}(s)"'] & \mathcal{P}(\Gamma). \end{tikzcd}$$

The elements of $\mathcal{P}(X)$, where X ranges over objects of $\mathcal{C}$, are referred to as *$\mathcal{P}$-predicates*.

Interpretation of first-order logic in a first-order hyperdoctrine. Given a (first-order) signature with types X, function symbols $f : X_1, \dots, X_n \to X$, and relation symbols $R \subset X_1, \dots, X_n$, a *structure* for the signature in a first-order hyperdoctrine $\mathcal{P}$ over $\mathcal{C}$ assigns an object $[\![X]\!]$ in $\mathcal{C}$ to each type, a morphism $[\![f]\!] : [\![X_1]\!] \times \cdots \times [\![X_n]\!] \to [\![X]\!]$ to each function symbol, and a $\mathcal{P}$-predicate $[\![R]\!] \in \mathcal{P}([\![X_1]\!] \times \cdots \times [\![X_n]\!])$ to each relation symbol. Any term t over the signature, with free variables in $\Gamma = \{x_1{:}X_1, \dots, x_n{:}X_n\}$ and of type X, say, is interpreted as a morphism $[\![t]\!] : [\![\Gamma]\!] \to [\![X]\!]$, where $[\![\Gamma]\!] = [\![X_1]\!] \times \cdots \times [\![X_n]\!]$, by induction on the structure of t (in the standard manner in which terms are interpreted in categories).

Each formula φ with free variables in Γ is interpreted as a $\mathcal{P}$-predicate $[\![\varphi]\!] \in \mathcal{P}([\![\Gamma]\!])$ by induction on the structure of φ using the properties given in Definition 2.1. For atomic formulas $R(t_1, \dots, t_n)$, the interpretation is given by

$$\mathcal{P}(\langle [\![t_1]\!], \dots, [\![t_n]\!] \rangle)([\![R]\!]).$$

In particular, the atomic formula $t =_X t'$ is interpreted by the $\mathcal{P}$-predicate

$$\mathcal{P}(\langle [\![t]\!], [\![t']\!] \rangle)(=_{[X]}).$$

The interpretation of other formulas is defined by structural induction. Assume φ, φ' are formulas with free variables in Γ and that ψ is a formula with free variables in $\Gamma \cup \{x{:}X\}$. Then,

$$\begin{aligned} [\![\top]\!] &= \top_H & [\![\varphi \wedge \varphi']\!] &= [\![\varphi]\!] \wedge_H [\![\varphi']\!] \\ [\![\bot]\!] &= \bot_H & [\![\varphi \vee \varphi']\!] &= [\![\varphi]\!] \vee_H [\![\varphi']\!] \\ & & [\![\varphi \to \varphi']\!] &= [\![\varphi]\!] \to_H [\![\varphi']\!] \end{aligned}$$

$$\begin{aligned} [\![\forall x{:}X.\psi]\!] &= (\forall [\![X]\!])_{[\![\Gamma]\!]}([\![\psi]\!]) \in \mathcal{P}([\![\Gamma]\!]) \\ [\![\exists x{:}X.\psi]\!] &= (\exists [\![X]\!])_{[\![\Gamma]\!]}([\![\psi]\!]) \in \mathcal{P}([\![\Gamma]\!]), \end{aligned}$$

where $\wedge_H, \vee_H$, etc., is the Heyting algebra structure on $\mathcal{P}([\![\Gamma]\!])$. Finally, one may show that $[\![\varphi[f(x)/y]]\!]$ is interpreted by $\mathcal{P}([\![f]\!])([\![\varphi]\!])$, so one should think of $\mathcal{P}(g)$ as the interpretation of substitution.

A formula φ with free variables in Γ is said to be *satisfied* if $[\![\varphi]\!]$ is the top element of $\mathcal{P}([\![\Gamma]\!])$. This notion of satisfaction is *sound* for intuitionistic predicate logic in the sense that all provable formulas are satisfied. Moreover, it is *complete* in the sense that a formula is provable if it is satisfied in all structures in first-order hyperdoctrines. A first-order hyperdoctrine $\mathcal{P}$ is sound for *classical* predicate logic in case all the fibers $\mathcal{P}(X)$ are Boolean algebras.

Definition 2.2 (*Hyperdoctrine*). A (general) *hyperdoctrine* is a first-order hyperdoctrine with the following additional properties: $\mathcal{C}$ is Cartesian closed, and there is an internal Heyting algebra H (for the definition of internal Heyting algebra, see e.g., MacLane and Moerdijk [1994]) and a natural bijection $\Theta_X : Obj(\mathcal{P}(X)) \simeq \mathcal{C}(X, H)$.

Higher-order intuitionistic predicate logic is first-order intuitionistic predicate logic extended with a type Prop of propositions and with higher types. See, for instance Jacobs [1999] for a formal presentation. A hyperdoctrine is sound for higher-order intuitionistic predicate logic: The Heyting algebra H is used to interpret the type Prop of propositions and higher types (e.g., Prop^X, the type for predicates over X), are interpreted by exponentials in $\mathcal{C}$. The natural bijection Θ_X is used to interpret substitution of formulas in formulas: Suppose φ is a formula with a free variable q of type Prop and with remaining free variables in Γ, and that ψ is a formula with free variables in Γ. Then $[\![\psi]\!] \in \mathcal{P}([\![\Gamma]\!])$, $[\![\varphi]\!] \in \mathcal{P}([\![\Gamma]\!] \times H)$, and $\varphi[\psi/q]$ (φ with ψ substituted in for q) is interpreted by $\mathcal{P}(\langle \mathrm{id}, \Theta([\![\psi]\!])\rangle)([\![\varphi]\!])$. For more details see, for instance the handbook chapter [Pitts 2001].

Again it is the case that a hyperdoctrine $\mathcal{P}$ is sound for classical higher-order predicate logic in case all the fibers $\mathcal{P}(X)$ are Boolean algebras.

Example 2.3 (*Canonical Hyperdoctrine Over a Topos*). Let $\mathcal{E}$ be a topos. It is well-known that $\mathcal{E}$ models higher-order, intuitionistic predicate logic. In addition, a topos also models full subset types and extensionality (see, e.g., Jacobs [1999]). The interpretation is given by interpreting types as objects in $\mathcal{E}$, terms as morphisms in $\mathcal{E}$, and predicates as subobjects in $\mathcal{E}$. The topos $\mathcal{E}$ induces a canonical $\mathcal{E}$-indexed hyperdoctrine $\mathrm{Sub}_{\mathcal{E}} : \mathcal{E}^{op} \to \mathbf{Poset}$, which maps an object X in $\mathcal{E}$ to the poset of subobjects of X in $\mathcal{E}$ and a morphism $f : X \to Y$ to the pullback functor $f^* : \mathrm{Sub}(Y) \to \mathrm{Sub}(X)$. Then the standard interpretation of predicate logic in $\mathcal{E}$ coincides with the interpretation of predicate logic in the hyperdoctrine $\mathrm{Sub}_{\mathcal{E}}$. Compared to the standard interpretation in toposes, however, hyperdoctrines do not require that predicates are always modeled by subobjects, but can come from some other universe. This means that hyperdoctrines describe a wider class of models than do toposes.

2.2 BI Hyperdoctrines

We now present a straightforward extension of first-order hyperdoctrines which models first and higher-order predicate BI. Recall that a *BI algebra* is a Heyting algebra which has an additional symmetric monoidal closed structure $(\mathrm{I}, *, -\!*)$ [Pym 2002].

Definition 2.4 (*Bi Hyperdoctrine*).

—A first-order hyperdoctrine $\mathcal{P}$ over $\mathcal{C}$ is a *first-order BI hyperdoctrine* in the case where all the fibers $\mathcal{P}(X)$ are BI algerbras and the reindexing functions $\mathcal{P}(f)$ are BI algebra homomorphisms.

—A *BI hyperdoctrine* is a first-order BI hyperdoctrine with the additional properties that $\mathcal{C}$ is Cartesian closed, and there is a BI algebra B and a bijection $\Theta_X : Obj(\mathcal{P}(X)) \simeq \mathcal{C}(X, B)$, natural in X. In other words, $Obj(\mathcal{P}(-))$ and $\mathcal{C}(-, B)$ are isomorphic as objects in the functor category $\mathsf{Set}^{\mathcal{C}^{op}}$.

First-order predicate BI is first-order, intuitionistic predicate logic with equality, extended with formulas I, $\varphi * \psi$, $\varphi -\!* \psi$ satisfying the following rules (in any context Γ including the free variables of the formulas).

$$(\varphi * \psi) * \theta \vdash_\Gamma \varphi * (\psi * \theta) \qquad \varphi * (\psi * \theta) \vdash_\Gamma (\varphi * \psi) * \theta \qquad \vdash_\Gamma \varphi \leftrightarrow \varphi * \mathrm{I}$$

$$\varphi * \psi \vdash_\Gamma \psi * \varphi \qquad \frac{\varphi \vdash_\Gamma \psi \quad \theta \vdash_\Gamma \omega}{\varphi * \theta \vdash_\Gamma \psi * \omega} \qquad \frac{\varphi * \psi \vdash_\Gamma \theta}{\varphi \vdash_\Gamma \psi -\!* \theta}$$

Our notion of predicate BI should not be confused with the one presented in Pym [2002]; the latter seeks to include a BI structure on contexts but we do not attempt to do that here, since this is not what is used in separation logic. In particular, weakening at the level of variables is always allowed.

$$\frac{\varphi \vdash_\Gamma \psi}{\varphi \vdash_{\Gamma \cup \{x:X\}} \psi}$$

We interpret first-order predicate BI in a first-order BI hyperdoctrine simply by extending the interpretation of first-order logic in the first-order hyperdoctrine defined before by

$$\begin{aligned} [\![\mathrm{I}]\!] &= \mathrm{I}_B \\ [\![\varphi * \psi]\!] &= [\![\varphi]\!] *_B [\![\psi]\!] \\ [\![\varphi -\!* \psi]\!] &= [\![\varphi]\!] -\!*_B [\![\psi]\!], \end{aligned}$$

where I_B, $*_B$, and $-\!*_B$ comprise the monoidal closed structure in the BI algebra $\mathcal{P}([\![\Gamma]\!])$. We then have

THEOREM 2.5.

(1) *The interpretation of first-order predicate BI given previously is sound and complete.*

(2) *The interpretation of higher-order predicate BI given previously is sound and complete.*

Proof. Soundness is proved by straightforward induction and completeness is proved by forming the Lindenbaum-Tarski algebra over each context Γ of variables, and showing that this gives a first-order BI hyperdoctrine in the first case, and a BI hyperdoctrine in the second. The proof is a simple extension of the proof of the corresponding result for intuitionistic predicate logic given in Jacobs [1999]. □

Of course, a first-order BI hyperdoctrine is sound for classical BI in the case where all the fibers $\mathcal{P}(X)$ are Boolean BI algebras and all the reindexing functions $\mathcal{P}(f)$ are Boolean BI algebra homomorphisms. The following is a canonical example of a BI hyperdoctrine, which we will use later in Section 3.2 to show that the pointer model is actually an instance of a BI hyperdoctrine.

Example 2.6 (*Bi Hyperdoctrine Over a Complete Bi Algebra*). Let B be a complete BI algebra, namely, it has all joins and meets. It determines a BI hyperdoctrine over the category **Set** as follows. For each set X, let $\mathcal{P}(X) = B^X$, that is, the set of functions from X to B, be ordered pointwise. Given $f : X \to Y$, $\mathcal{P}(f) : B^Y \to B^X$ is the BI algebra homomorphism given by composition with f. For example if $s, t \in \mathcal{P}(Y)$, that is, $s, t : Y \to B$, then $\mathcal{P}(f)(s) = s \circ f : X \to B$ and $s * t$ is defined pointwise as $(s * t)(y) = s(y) * t(y)$. Equality predicates $=_X$ in $B^{X\times X}$ are defined by

$$=_X (x, x') \overset{\text{def}}{=} \begin{cases} \top & \text{if } x = x' \\ \bot & \text{if } x \neq x' \end{cases},$$

where $\top$ and $\bot$ are the greatest and least elements of B, respectively. The quantifiers use set-indexed joins ($\bigvee$) and meets ($\bigwedge$). Specifically, given $A \in B^{\Gamma\times X}$, one has

$$(\exists X)_\Gamma(A) \overset{\text{def}}{=} \lambda i \in \Gamma.\ \bigvee_{x\in X} A(i, x) \qquad\qquad (\forall X)_\Gamma(A) \overset{\text{def}}{=} \lambda i \in \Gamma.\ \bigwedge_{x\in X} A(i, x)$$

in B^Γ. The conditions in Definition 2.2 are trivially satisfied (Θ is the identity).

This example can be stated more generally by replacing Set with any Cartesian closed category $\mathcal{C}$ and let B be an internal, complete BI algebra, that is, B is a BI algebra object in $\mathcal{C}$ which is complete as an internal Heyting algebra. There are plenty of examples of complete BI algebras: For any Grothendieck topos $\mathcal{E}$ with an additional symmetric monoidal closed structure, $\mathrm{Sub}_{\mathcal{E}}(1)$ is a complete BI algebra, and for any monoidal category $\mathcal{C}$ such that the monoid is cover preserving with respect to the Grothendieck topology J, $\mathrm{Sub}_{\mathrm{Sh}(\mathcal{C},J)}(1)$ is a complete BI algebra [Biering 2004; Pym et al. 2004]. For a different kind of example based on realizability, see Biering et al. [2006].

The following theorem shows that to get interesting models of higher-order predicate BI, it does not suffice to consider BI hyperdoctrines arising as the canonical hyperdoctrine over a topos (as in Example 2.3). Indeed, this is the reason for introducing the more general BI hyperdoctrines.

THEOREM 2.7. *Let $\mathcal{E}$ be a topos and suppose $Sub_{\mathcal{E}} : \mathcal{E}^{op} \to$ **Poset** is a BI hyperdoctrine. Then the BI structure on each lattice $Sub_{\mathcal{E}}(X)$ is trivial, that is, for all $\varphi, \psi \in Sub_{\mathcal{E}}(X)$, $\varphi * \psi \leftrightarrow \varphi \wedge \psi$.*

PROOF. Let $\mathcal{E}$ be a topos and suppose $\mathrm{Sub}_{\mathcal{E}} : \mathcal{E}^{op} \to$ **Poset** is a BI hyperdoctrine. Let X be an object of $\mathcal{E}$ and let $\varphi, \psi, \psi' \in \mathrm{Sub}_{\mathcal{E}}(X)$. Furthermore, let Y be the domain of the mono φ, and notice that the lattice $\mathrm{Sub}_{\mathcal{E}}(Y)$ can be characterized by

$$\mathrm{Sub}_{\mathcal{E}}(Y) = \{\psi \wedge \varphi \mid \psi \in \mathrm{Sub}_{\mathcal{E}}(X)\}. \tag{1}$$

Moreover, notice that the order on $\mathrm{Sub}_{\mathcal{E}}(Y)$ is inherited from $\mathrm{Sub}_{\mathcal{E}}(X)$, that is,

$$\text{for all } \chi, \chi' \in \mathrm{Sub}_{\mathcal{E}}(Y),\ \chi \vdash_Y \chi' \text{ iff } \chi \vdash_X \chi'. \tag{2}$$

Since $\wedge$ is modeled by pullback which by assumption preserves $*$, the following equations hold in $\mathrm{Sub}_{\mathcal{E}}(Y)$ (and therefore also in $\mathrm{Sub}_{\mathcal{E}}(X)$).

$$(\varphi \wedge \psi) *_Y (\varphi \wedge \psi') \leftrightarrow \varphi \wedge (\psi *_X \psi') \tag{3}$$

and

$$(\varphi \wedge \psi) \mathrel{-\!\!*}_Y (\varphi \wedge \psi') \leftrightarrow \varphi \wedge (\psi \mathrel{-\!\!*}_X \psi') \tag{4}$$

By assumption, $\mathrm{Sub}_{\mathcal{E}}(Y)$ forms a BI algebra with connectives $*_Y$, $-\!\!*_Y$, and I_Y, so using the characterization of subobjects of Y given in Eq. (1) yields the following rule for each $\chi \in \mathrm{Sub}_{\mathcal{E}}(X)$:

$$\frac{(\varphi \wedge \psi) *_Y (\varphi \wedge \psi') \vdash_Y \chi \wedge \varphi}{\varphi \wedge \psi \vdash_Y (\varphi \wedge \psi') \mathrel{-\!\!*}_Y (\chi \wedge \varphi)}$$

Using (2), (3), and (4), we deduce that

$$\frac{\varphi \wedge (\psi *_X \psi') \vdash_X \chi \wedge \varphi}{\varphi \wedge \psi \vdash_X \varphi \wedge (\psi' \mathrel{-\!\!*}_X \chi)}$$

for all $\varphi, \psi, \psi', \chi \in \mathrm{Sub}_{\mathcal{E}}(X)$, which implies

$$\cfrac{\varphi \wedge (\psi *_X \psi') \vdash_X \chi \wedge \varphi}{\cfrac{\varphi \wedge \psi \vdash_X \psi' \mathrel{-\!\!*}_X \chi}{(\varphi \wedge \psi) *_X \psi' \vdash_X \chi}}\ . \tag{5}$$

Inserting $\varphi \wedge (\psi *_X \psi')$ for χ into (5) yields

$$\frac{\varphi \wedge (\psi *_X \psi') \vdash_X \varphi \wedge (\psi *_X \psi')}{(\varphi \wedge \psi) *_X \psi' \vdash_X \varphi \wedge (\psi *_X \psi')}\ . \tag{6}$$

Since the entailment preceding the line in Eq. (6) always holds,

$$(\varphi \wedge \psi) *_X \psi' \vdash_X \varphi \wedge (\psi *_X \psi').$$

This gives us projections for $*_X$ by letting ψ be $\top$.

$$(\varphi *_X \psi') \dashv\vdash_X (\varphi \wedge \top) *_X \psi' \vdash_X \varphi \wedge (\top *_X \psi') \vdash_X \varphi$$

Now, let χ be the subobject $(\varphi \wedge \psi) *_X \psi'$, then $\chi \leftrightarrow \chi \wedge \varphi$ due to the projections for $*_X$. Using (5) in bottom-up fashion gives

$$\frac{(\varphi \wedge \psi) *_X \psi' \vdash_X (\varphi \wedge \psi) *_X \psi'}{\varphi \wedge (\psi *_X \psi') \vdash_X (\varphi \wedge \psi) *_X \psi'} \,. \tag{7}$$

By Eqs. (6) and (7) we conclude that for all $\varphi, \psi, \psi' \in \mathrm{Sub}_{\mathcal{E}}(X)$,

$$\varphi \wedge (\psi *_X \psi') \leftrightarrow (\varphi \wedge \psi) *_X \psi'. \tag{8}$$

We already noted the projections for $*_X$, so $\top *_X \mathrm{I}_X \vdash_X \mathrm{I}_X$, which entails $\top \leftrightarrow \mathrm{I}_X$. Let ψ be $\top$ in (8), then $\varphi \wedge (\top *_X \psi') \leftrightarrow (\varphi \wedge \top) *_X \psi'$, and so $\varphi \wedge \psi' \leftrightarrow \varphi *_X \psi'$, as claimed. □

In fact, it is possible to slightly strengthen Theorem 2.7. We say that a logic has *full subset types* [Jacobs 1999] if the following conditions are satisfied:

—For each formula $\varphi(x_1, \ldots, x_n)$, there is a type $\{x_1{:}\tau_1, \ldots, x_n{:}\tau_n \mid \varphi(x_1, \ldots, x_n)\}$.
—For a term N of type $\{x_1{:}\tau_1, \ldots, x_n{:}\tau_n \mid \varphi(x_1, \ldots, x_n)\}$, in a context Γ, there is a term $\mathsf{o}(N)$ of type $\tau_1 \times \cdots \times \tau_n$ in Γ.
—The rule

$$\frac{\Gamma,\, y{:}\{x{:}X \mid \varphi\} \mid \theta[\mathsf{o}(y)/x] \vdash \psi[\mathsf{o}(y)/x]}{\Gamma, x{:}X \mid \theta, \varphi \vdash \psi} \tag{9}$$

is valid. Here $\Gamma \mid \varphi \vdash \psi$ is an alternative notation for $\varphi \vdash_\Gamma \psi$ to make the previous formula more readable.

One can then show

Proposition 2.8. *Adding the aforementioned rules for full subset types to our notion of predicate BI yields a logic where for all formulas φ, ψ in a context Γ,*

$$\varphi \wedge \psi \dashv\vdash_\Gamma \varphi * \psi.$$

The proof may be found in Appendix A. The following is an easy consequence.

Corollary 2.9. *Any BI hyperdoctrine which satisfies the rules for full subset types is trivial.*

The BI hyperdoctrine S, which we define next and which corresponds to the standard pointer model of separation logic, satisfies all of the preceding except the downward direction of (9). When this is the case, we say that the logic has subset types, but not *full* subset types [Jacobs 1999]. In fact, any BI hyperdoctrine over a complete BI algebra, that is, following the recipe of Example 2.6, has subset types but not necessarily full subset types.

3. SEPARATION LOGIC MODELED BY BI-HYPERDOCTRINES

We briefly recall the standard pointer model of separation logic (for a more thorough presentation, see, e.g., Reynolds [2002]) and then show how it can be construed as a BI hyperdoctrine over Set.

The core assertion language of separation logic (which we will henceforth also call separation logic) is often defined as follows. There is a single type Val of values. Terms t are defined by a grammar

$$t ::= x \mid n \mid t + t \mid t - t \mid \cdots,$$

where n : Val are constants for all integers n. Formulas, also called assertions, are defined by

$$\varphi ::= \top \mid \bot \mid t = t \mid t \mapsto t \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \varphi * \varphi \mid \varphi -\!\!* \ \varphi \mid \texttt{emp} \mid \forall x.\varphi \mid \exists x.\varphi.$$

The symbol emp is used in separation logic for the unit of BI.

Note that the aforementioned is just another way of defining a signature (i.e., specification of types, function symbols, and predicate symbols) for first-order predicate BI with a single type Val, function symbols $+, -, \ldots$: Val, Val $\to$ Val, constants n : Val, and relation symbol $\mapsto \subseteq$ Val $\times$ Val.

3.1 The Pointer Model

The standard pointer model of separation logic is usually presented as follows. It consists of a set $[\![\mathsf{Val}]\!]$ interpreting the type Val, a set $[\![\mathsf{Loc}]\!]$ of locations such that $[\![\mathsf{Loc}]\!] \subseteq [\![\mathsf{Val}]\!]$, and binary functions on $[\![\mathsf{Val}]\!]$ interpreting the function symbols $+, -$. The set $H = [\![\mathsf{Loc}]\!] \rightharpoonup_{fin} [\![\mathsf{Val}]\!]$ of finite partial functions from $[\![\mathsf{Loc}]\!]$ to $[\![\mathsf{Val}]\!]$, ordered discretely, is referred to as the set of *heaps*. The set of heaps has a partial binary operation $*$ defined by

$$h_1 * h_2 = \begin{cases} h_1 \cup h_2 & \text{if } h_1 \# h_2 \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where # is the binary relation on heaps defined by $h_1 \# h_2$ iff $\mathrm{dom}(h_1) \cap \mathrm{dom}(h_2) = \emptyset$. The interpretation of the relation $\mapsto$ is the function $[\![\mathsf{Val}]\!] \times [\![\mathsf{Val}]\!] \to P(H)$ given by $h \in [\![v_1 \mapsto v_2]\!]$ iff $\mathrm{dom}(h) = \{v_1\}$ and $h(v_1) = v_2$. To define the standard interpretation of terms and formulas, one assumes a partial function s : Var $\rightharpoonup_{fin}$ $[\![\mathsf{Val}]\!]$, called a *stack* (also called a *store* in the literature). The interpretation of terms depends on the stack and is defined by

$$\begin{aligned} [\![x]\!]s &= s(x) \\ [\![n]\!]s &= [\![n]\!] \\ [\![t_1 \pm t_2]\!]s &= [\![t_1]\!]s \pm [\![t_2]\!]s. \end{aligned}$$

The interpretation of formulas is standardly given by a forcing relation $s, h \models \varphi$, where $\mathrm{FV}(\varphi) \subseteq \mathrm{dom}(s)$, as follows:

$$\begin{array}{ll} s, h \models t_1 = t_2 & \text{iff } [\![t_1]\!]s = [\![t_2]\!]s \\ s, h \models t_1 \mapsto t_2 & \text{iff } \mathrm{dom}(h) = \{[\![t_1]\!]s\} \text{ and } h([\![t_1]\!]s) = [\![t_2]\!]s \\ s, h \models \texttt{emp} & \text{iff } h = \emptyset \\ s, h \models \top & \text{always} \\ s, h \models \bot & \text{never} \end{array}$$

$$\begin{array}{lll}
s, h \models \varphi * \psi & \text{iff} & \text{there exists } h_1, h_2 \in H \text{ such that } h_1 * h_2 = h, \text{ and} \\
& & s, h_1 \models \varphi, \text{ and } s, h_2 \models \psi \\
s, h \models \varphi \mathrel{-\!\!*} \psi & \text{iff} & \text{for all } h', h' \# h, \text{ and } s, h' \models \varphi \text{ implies } s, h * h' \models \psi \\
s, h \models \varphi \vee \psi & \text{iff} & s, h \models \varphi \text{ or } s, h \models \psi \\
s, h \models \varphi \wedge \psi & \text{iff} & s, h \models \varphi \text{ and } s, h \models \psi \\
s, h \models \varphi \rightarrow \psi & \text{iff} & s, h \models \varphi \text{ implies } s, h \models \psi \\
s, h \models \forall x.\varphi & \text{iff} & \text{for all } v \in [\![\mathsf{Val}]\!],\ s[x \mapsto v], h \models \varphi \\
s, h \models \exists x.\varphi & \text{iff} & \text{there exists } v \in [\![\mathsf{Val}]\!], \text{ such that } s[x \mapsto v], h \models \varphi
\end{array}$$

Remark 3.1. The pointer model has a single-sorted signature (the only type is Val), and to get a many-sorted or higher-order version of the pointer model, we add appropriate types to the signature. Variables come with a type $x : X$, and we require that $s(x : X) \in [\![X]\!]$ for all variables $x \in \operatorname{dom} s$. The last two rules of the forcing relation, become typed.

$$s, h \models \forall x : X.\varphi \text{ iff for all } v \in [\![X]\!],\ s[x \mapsto v], h \models \varphi$$

and similar for the exists rule.

We now show how this pointer model is an instance of a BI-hyperdoctrine of a complete Boolean BI algebra (compare with Example 2.6).

3.2 The Pointer Model as a BI Hyperdoctrine

Let $(H_\perp, *)$ be the discretely ordered set of heaps with a bottom element added to represent undefined, and let $* : H_\perp \times H_\perp \to H_\perp$ be the total extension of $* : H \times H \rightharpoonup H$ satisfying $\perp * h = h * \perp = \perp$, for all $h \in H_\perp$, and $h * h' = \perp$ if h and h' are not disjoint. This defines an ordered, commutative monoid with the empty heap $\emptyset$ as the unit for $*$. The powerset of H, $\mathcal{P}(H)$ (without $\perp$) is a complete Boolean BI algebra ordered by inclusion and with monoidal closed structure given by (for $U, V \in \mathcal{P}(H)$):

—I is $\{\emptyset\}$;
—$U * V := \{h * h' \mid h \in U \wedge h' \in V\} \setminus \{\perp\}$; and
—$U \mathrel{-\!\!*} V := \bigcup\{W \subseteq H \mid (W * U) \subseteq V\}$.

It can easily be verified directly that this defines a complete Boolean BI algebra; it also follows from more abstract arguments in Pym et al. [2004] and Biering [2004].

Let S be the BI hyperdoctrine induced by the complete Boolean BI algebra $\mathcal{P}(H)$, as in Example 2.6. To show that the interpretation of separation logic in this BI hyperdoctrine exactly corresponds to the standard pointer model presented earlier, we spell out the interpretation of separation logic in S.

A term t in a context $\Gamma = \{x_1 : \mathsf{Val}, \ldots, x_n : \mathsf{Val}\}$ is interpreted as a morphism between sets:

—$[\![x_i : \mathsf{Val}]\!] = \pi_i$, where $\pi_i : \mathsf{Val}^n \to \mathsf{Val}$ is the ith projection;
—$[\![n]\!]$ is the map $[\![n]\!] : [\![\Gamma]\!] \to 1 \to [\![\mathsf{Val}]\!]$ which sends the unique element of the one-point set 1 to $[\![n]\!]$; and

— $[\![t_1 \pm t_2]\!] = [\![t_1]\!] \pm [\![t_2]\!] : [\![\Gamma]\!] \to [\![\mathsf{Val}]\!] \times [\![\mathsf{Val}]\!] \to [\![\mathsf{Val}]\!]$, where $[\![t_i]\!] : [\![\Gamma]\!] \to [\![\mathsf{Val}]\!]$, for $i = 1, 2$.

The interpretation of a formula φ in a context $\Gamma = \{x_1 : \mathsf{Val}, \ldots, x_n : \mathsf{Val}\}$ is given inductively as follows. Let $[\![\Gamma]\!] = [\![\mathsf{Val}]\!] \times \cdots \times [\![\mathsf{Val}]\!] = [\![\mathsf{Val}]\!]^n$ and write $\overline{\mathrm{v}}$ for elements of $[\![\Gamma]\!]$. Then φ is interpreted as an element of $\mathcal{P}[\![\Gamma]\!]$ as follows:

$$\begin{array}{lcl}
[\![t_1 \mapsto t_2]\!](\overline{\mathrm{v}}) & = & \{h \mid \mathrm{dom}(h) = \{[\![t_1]\!](\overline{\mathrm{v}})\} \text{ and } h([\![t_1]\!](\overline{\mathrm{v}})) = [\![t_2]\!](\overline{\mathrm{v}})\} \\
[\![t_1 = t_2]\!](\overline{\mathrm{v}}) & = & H \text{ if } [\![t_1]\!](\overline{\mathrm{v}}) = [\![t_2]\!](\overline{\mathrm{v}}),\ \emptyset \text{ otherwise} \\
[\![\top]\!](*) & = & H \\
[\![\bot]\!](*) & = & \emptyset \\
[\![\mathtt{emp}]\!](*) & = & \{h \mid \mathrm{dom}(h) = \emptyset\} \\
[\![\varphi \wedge \psi]\!](\overline{\mathrm{v}}) & = & [\![\varphi]\!](\overline{\mathrm{v}}) \cap [\![\psi]\!](\overline{\mathrm{v}}) \\
[\![\varphi \vee \psi]\!](\overline{\mathrm{v}}) & = & [\![\varphi]\!](\overline{\mathrm{v}}) \cup [\![\psi]\!](\overline{\mathrm{v}}) \\
[\![\varphi \to \psi]\!](\overline{\mathrm{v}}) & = & \{h \mid h \in [\![\varphi]\!](\overline{\mathrm{v}}) \text{ implies } h \in [\![\psi]\!](\overline{\mathrm{v}})\} \\
[\![\varphi * \psi]\!](\overline{\mathrm{v}}) & = & [\![\varphi]\!](\overline{\mathrm{v}}) * [\![\psi]\!](\overline{\mathrm{v}}) \\
 & = & \{h_1 * h_2 \mid h_1 \in [\![\varphi]\!](\overline{\mathrm{v}}) \text{ and } h_2 \in [\![\psi]\!](\overline{\mathrm{v}})\} \setminus \{\bot\} \\
[\![\varphi \mathrel{-\!\!*} \psi]\!](\overline{\mathrm{v}}) & = & [\![\varphi]\!](\overline{\mathrm{v}}) \mathrel{-\!\!*} [\![\psi]\!](\overline{\mathrm{v}}) \\
 & = & \{h \mid [\![\varphi]\!](\overline{\mathrm{v}}) * \{h\} \subseteq [\![\psi]\!](\overline{\mathrm{v}})\} \\
[\![\forall x : \mathsf{Val}.\varphi]\!](\overline{\mathrm{v}}) & = & \bigcap_{v_x \in [\![\mathsf{Val}]\!]}([\![\varphi]\!](v_x, \overline{\mathrm{v}})) \\
[\![\exists x : \mathsf{Val}.\varphi]\!](\overline{\mathrm{v}}) & = & \bigcup_{v_x \in [\![\mathsf{Val}]\!]}([\![\varphi]\!](v_x, \overline{\mathrm{v}}))
\end{array}$$

Now it is easy to verify, by structural induction on formulas φ, that the interpretation given in the BI hyperdoctrine S corresponds exactly to the forcing semantics given earlier.

THEOREM 3.2. $h \in [\![\varphi]\!](v_1, \ldots, v_n)$ *iff* $[x_1 \mapsto v_1, \ldots, x_n \mapsto v_n], h \models \varphi$.

As a consequence, we of course obtain the well-known result that separation logic is sound for classical first-order BI. But, more interestingly, the correspondence also shows that we may easily extend separation logic to higher-order, since the BI hyperdoctrine S soundly models higher-order BI. We expand on this in the next section, which also discusses other consequences of the aforementioned correspondence. First, however, we explain that one can also obtain such a correspondence for other versions of separation logic.

3.3 An Intuitionistic Model

Consider again the set of heaps $(H_\bot, *)$ with an added bottom $\bot$, as before. We now define the order by

$$h_1 \sqsupseteq h_2 \quad \text{iff} \quad \mathrm{dom}(h_1) \subseteq \mathrm{dom}(h_2) \text{ and for all } x \in \mathrm{dom}(h_1).\ h_1(x) = h_2(x).$$

Let I be the set of sieves on H, that is, downwards closed subsets of H, ordered by inclusion. This is a complete BI algebra, as can be verified directly or by abstract argument [Biering 2004; Pym et al. 2004].

Now let T be the BI hyperdoctrine induced by the complete BI algebra I, as in Example 2.6. The interpretation of predicate BI in this BI hyperdoctrine corresponds exactly to the intuitionistic pointer model of separation logic, which is presented using a forcing-style semantics in Ishtiaq and O'Hearn [2001].

3.4 The Permissions Model

It is also possible to fit the permissions model of separation logic from Bornat et al. [2005] into the framework presented here. The main point is that the set of heaps (which in that model map locations to values and permissions) has a binary operation $*$, that makes $(H_\perp, *)$ a partially ordered commutative monoid.

Remark 3.3. The correspondences between separation logic and BI hyperdoctrines given previously illustrate that what matters for the interpretation of separation logic is the choice of BI algebra. Indeed, the main relevance of the topos-theoretic constructions in Pym et al. [2004] for models of separation logic is that they can be used to construct suitable BI algebras (as subobject lattices in categories of sheaves).

4. SOME CONSEQUENCES FOR SEPARATION LOGIC

We have shown earlier that it is completely natural and straightforward to interpret first-order predicate BI in first-order BI hyperdoctrines and that the standard pointer model of separation logic corresponds to a particular case of BI hyperdoctrine. Based on this correspondence, in this section we draw some further consequences for separation logic.

4.1 Formalizing Separation Logic

The usefulness of separation logic has been demonstrated in numerous papers before. It has been shown that it can handle simple programs for copying trees, deleting lists, etc. The first proof of a more realistic program appeared in Yang's thesis [Yang 2001], in which he showed correctness of the Schorr-Waite graph marking algorithm. Later, a proof of correctness of Cheney's garbage collection algorithm was published in Birkedal et al. [2004], and other examples of correctness proofs of nontrivial algorithms may be found in Bornat et al. [2004]. In all of these papers, different simple extensions of core separation logic were used. For example, Yang [2001] used lists and binary trees as parts of his term language, and Birkedal et al. [2004] introduced expression forms for finite sets and relations. It would seem a weakness of separation logic that one has to come up with suitable extensions of it every time one has to prove a new program correct. In particular, it would make machine-verifiable formalizations of such proofs more burdensome and dubious if one would have to alter the underlying logic for every new proof.

The right way to look at these "extensions" is that they are really trivial definitional extensions of one and the same logic, namely, the internal logic of the classical BI hyperdoctrine S presented in Section 3. The internal language of a BI hyperdoctrine $\mathcal{P}$ over $\mathcal{C}$ is formed as follows: To each object of $\mathcal{C}$ one associates a type, to each morphism of $\mathcal{C}$ one associates a function symbol, and to each predicate in $\mathcal{P}(X)$ one associates a relation symbol. The terms and formulas over this signature (considered as a higher-order signature [Jacobs

1999]) form the internal language of the BI hyperdoctrine. There is an obvious structure for this language in $\mathcal{P}$.

Let $2 = \{\bot, \top\}$ be a two-element set (the subobject classifier of Set). There is a canonical map $\iota : 2 \to \mathcal{P}(H)$ which maps $\bot$ to $\{\}$ (the bottom element of the BI algebra $\mathcal{P}(H)$) and $\top$ to H (the top element of $\mathcal{P}(H)$).

Definition 4.1. Let φ be an S-predicate over a set X, namely, a function $\varphi : X \to \mathcal{P}(H)$. Call φ *pure* if φ factors through ι.

Thus $\varphi : X \to \mathcal{P}(H)$ is pure if there exists a map $\chi_\varphi : X \to 2$ such that

$$\begin{tikzcd} X \arrow[rr, "\varphi"] \arrow[dr, "\chi_\varphi"'] & & \mathcal{P}(H) \\ & 2 \arrow[ur, "\iota"'] & \end{tikzcd}$$

commutes. This corresponds to the notion of pure predicate traditionally used in separation logic [Reynolds 2002].

The sublogic of pure predicates is simply the standard classical higher-order logic of Set, and thus is sound for classical higher-order logic. Hence one can use classical higher-order logic for defining lists, trees, finite sets, and relations in the standard manner using pure predicates and can also prove the standard properties of these structures, as needed for the proofs presented in the aforementioned papers. In particular, notice that recursive definitions of predicates, which in the papers [Yang 2001; Birkedal et al. 2004; Bornat et al. 2004] are defined at the meta level, can be defined inside the higher-order logic itself, as detailed in Section 4.3. For machine verification one thus need only formalize the same exact logic, namely, a sufficient fragment of the internal logic of the BI hyperdoctrine (with obvious syntactic rules for when a formula is pure). The internal logic itself is “too big” (e.g., it can have class-many types and function symbols); hence the need for a fragment thereof, say, classical higher-order logic with natural numbers.

4.2 Logical Characterizations of Classes of Assertions

Different classes of assertions, precise, monotone, and pure, are introduced by Reynolds [2002], who notices that special axioms for these classes of assertions are valid. Such special axioms are exploited in the proof of Cheney’s garbage collector [Birkedal et al. 2004], where pure assertions are moved in and out of the scope of iterated separating conjunctions, and in the paper O’Hearn et al. [2004], where properties of precise assertions are crucially applied to verify soundness of the hypothetical frame rule. The different classes of assertions are defined semantically and the special axioms are validated using the semantics. We show how the higher-order features of higher-order separation logic allow a logical characterization of the classes of assertions, as well as logical proofs of the properties earlier taken as axioms. This is, of course, important for machine verification, since it means that the special classes of assertions and their properties can be expressed *in the logic*.

To simplify notation we just present the characterizations for *closed* assertions, the extension to open assertions begin straightforward. Recall that closed assertions are interpreted in S as functions from 1 to $\mathcal{P}(H)$, that is, as subsets of H.

In the proofs to follow, we use assertions which describe heaps in a canonical way. Since a heap h has finite domain, there is a unique (up to permutation) way to write an assertion $p_h \equiv l_1 \mapsto n_1 * \ldots * l_k \mapsto n_k$ such that $[\![p_h]\!] = \{h\}$.

Precise assertions. The traditional definition of a precise assertion is semantic inasmuch as an assertion q is precise if and only if for all states (s, h), there is at most one subheap h_0 of h such that $(s, h_0) \models q$. The following proposition logically characterizes closed precise assertions (at the semantic level, this characterization of precise predicates has been mentioned before [O'Hearn et al. 2003]).

PROPOSITION 4.2. *The closed assertion q is precise if and only if the assertion*

$$\forall p_1, p_2 : \mathsf{Prop}.\ (p_1 * q) \wedge (p_2 * q) \leftrightarrow (p_1 \wedge p_2) * q \tag{10}$$

is valid in the BI hyperdoctrine S.

PROOF. The "only-if" direction is trivial, so we focus on the other implication. Thus suppose (10) holds for q, and let h be a heap with two different subheaps h_1, h_2 for which $h_i \in [\![q]\!]$. Let p_1, p_2 be canonical assertions describing the heaps $h \setminus h_1$ and $h \setminus h_2$, respectively. Then $h \in [\![(p_1 * q) \wedge (p_2 * p)]\!]$, so $h \in [\![(p_1 \wedge p_2) * q]\!]$, whence there is a subheap $h' \subseteq h$ with $h' \in [\![p_1 \wedge p_2]\!]$. This is a contradiction. □

One can verify properties for precise assertions *in the logic* without using semantical arguments. For example, one can show that $q_1 * q_2$ is precise if q_1 and q_2 are by the following logical argument: Suppose (10) holds for q_1, q_2. Then,

$$\begin{array}{lll} & (p_1 * (q_1 * q_2)) \wedge (p_2 * (q_1 * q_2)) & \Rightarrow\ ((p_1 * q_1) * q_2) \wedge ((p_2 * q_1) * q_2)) \\ \Rightarrow & ((p_1 * q_1) \wedge (p_2 * q_1)) * q_2 & \Rightarrow\ ((p_1 \wedge p_2) * q_1) * q_2 \\ \Rightarrow & (p_1 \wedge p_2) * (q_1 * q_2), & \end{array}$$

as desired.

Monotone assertions. A closed assertion q is defined to be *monotone* if and only if whenever $h \in [\![q]\!]$, then also $h' \in [\![q]\!]$, for all extensions $h' \supseteq h$.

PROPOSITION 4.3. *The closed assertion q is* monotone *if and only if the assertion* $\forall p{:}\mathsf{Prop}.\ p * q \rightarrow q$ *is valid in the BI hyperdoctrine S.*

This is easily verified, and again, one can show the usual rules for monotone assertions in the logic (without semantical arguments) using this characterization.

Pure assertions. Recall from before that an assertion q is pure iff its interpretation factors through 2. Thus, a closed assertion is pure iff its interpretation is either $\emptyset$ or H.

PROPOSITION 4.4. *The closed assertion q is pure if and only if the assertion*

$$\forall p_1, p_2{:}\mathsf{Prop}.\ (q \wedge p_1) * p_2 \leftrightarrow q \wedge (p_1 * p_2) \qquad (11)$$

is valid in the BI hyperdoctrine S.

PROOF. Again, the interesting direction here is the "if" implication. Suppose (11) holds for the assertion q, and that $h \in [\![q]\!]$. For any heap h_0, we must show that $h_0 \in [\![q]\!]$. This is done via the verification of two claims.

Fact 1: For all $h' \subseteq h$, $h' \in [\![q]\!]$. Proof: Let p_1 be a canonical description of h', and p_2 a canonical description of $h \setminus h'$. Then $h \in [\![q \wedge (p_1 * p_2)]\!]$, so by 11, $h \in [\![(q \wedge p_1) * p_2]\!]$. This means there is a split $h_1 * h_2 = h$ with $h_1 \in [\![q \wedge p_1]\!]$ and $h_2 \in [\![p_2]\!]$. But then, $h_2 = h \setminus h'$, so $h_1 = h'$, and thus, $h' \in [\![q]\!]$.

Fact 2: For all $h' \supseteq h$, $h' \in [\![q]\!]$. Proof: Let p_1 and p_2 be canonical descriptions of h and $h' \setminus h$, respectively. Then, $h' \in [\![(q \wedge p_1) * p_2]\!]$, so by 11, $h' \in [\![q \wedge (p_1 * p_2)]\!]$, and in particular, $h' \in [\![q]\!]$, as desired.

Using Facts 1 and 2, we deduce that $h \in [\![q]\!] \Rightarrow \mathtt{emp} \in [\![q]\!] \Rightarrow h_0 \in [\![q]\!]$. □

4.3 Predicates via Fixed Points

Consider the following predicate clist taken from Parkinson and Bierman [2005]. It is required to satisfy the recursive equation

$$\mathsf{clist} = \lambda(x, s).x = null \vee (\exists j, k.\ x \mapsto j, k * P(j, s) * \mathsf{clist}(k, s)),$$

for some specific P. Solutions to such equations are definable in higher-order separation logic. Indeed, we may define both minimal and maximal fixed points for any monotone operator on predicates, using standard encodings of fixed points (due to Prawitz and Scott, independently). To wit, consider for notational simplicity an arbitrary predicate

$$q : \mathsf{Prop} \vdash \varphi(q) : \mathsf{Prop}$$

satisfying that q only occurs positively in φ. Then

$$\mu q.\varphi(q) = \forall q.(\varphi(q) \to q) \to q$$

is the least fixed point for φ in the obvious sense that $\varphi(\mu q.\varphi(q)) \to \mu q.\varphi(q)$ and $\forall p.(\varphi(p) \to p) \to (\mu q.\varphi(q) \to p)$ holds in the logic. Note that the latter is the corresponding induction principle. Likewise,

$$\nu q.\varphi(q) = \exists q.(q \to \varphi(q)) \wedge q$$

is the maximal fixed point for φ.

5. HIGHER-ORDER SEPARATION LOGIC

We present a programming language and use the higher-order assertion language of the pointer-model BI hyperdoctrine S to give a specification logic for it. The programming language is a simple extension of that of standard separation logic with simple call-by-value procedures, and the program logic includes standard rules for these. The logic is for partial correctness and absence of pointer errors.

Programming language. The programming language uses a restricted set of terms of type Int, referred to as *expressions*, and uses *Booleans*, which consist of a restricted (heap-independent) set of terms of type Prop. E ranges over the set of terms of type Int, and B ranges over the Boolean terms. They are generated by the grammar.

$$\begin{array}{l} E ::= n \mid x \mid E + E \mid E - E \mid E \times E \mid null \\ B ::= E = E \mid E \leq E \mid B \wedge B \mid \cdots \end{array}$$

Formally, Booleans have type Prop in our system, but we sometimes write B : Bool if they can be generated from this grammar (i.e., Boolean expressions are pure assertions). Moreover, officially we always consider expressions and formulas in context and thus write $\Delta \vdash E{:}\mathsf{Int}$, $\Delta \vdash B{:}\mathsf{Bool}$, and $\Delta \vdash P{:}\mathsf{Prop}$ for expressions, Booleans, and general assertions, respectively. A context Δ is a pair $\Delta_l ; \Delta_p$ of contexts for logical and program variables (i.e., finite maps from variables to types).

The syntax of the programming language is given by the following grammar. Here, k ranges over a set of function names and x ranges over a set of program variables.

$$\begin{array}{rcl} c & ::= & \mathbf{skip} \\ & \mid & x := k_i(E_1, \ldots, E_{m_i}) \\ & \mid & \mathbf{newvar}\ x; c \\ & \mid & x := E \\ & \mid & x := [E] \\ & \mid & [E] := E' \\ & \mid & x := \mathbf{cons}(E_1, \ldots, E_m) \\ & \mid & \mathbf{dispose}(E) \\ & \mid & \mathbf{if}\ B\ \mathbf{then}\ c\ \mathbf{else}\ c\ \mathbf{fi} \\ & \mid & \mathbf{while}\ B\ \mathbf{do}\ c\ \mathbf{od} \\ & \mid & c; c \\ & \mid & \mathbf{let}\ k_1(x_1, \ldots, x_{m_1}) = c_1 \\ & & \quad \vdots \\ & & \quad k_n(x_1, \ldots, x_{m_n}) = c_n \\ & & \mathbf{in}\ c\ \mathbf{end} \\ & \mid & \mathbf{return}\ e \end{array}$$

There are some restrictions on the programs, and a program is called *well formed* if it meets them. The restrictions include:

—There is always a **return** at the end of a function body.
—A function name is declared at most once in a **let**.
—There are the right number of parameters in function calls.
—Function bodies modify neither nonlocal variables nor parameters.

The semantics is mostly standard; we specify it formally in the following. Note that the language includes a declaration of new local variables, as well as operations for reading from the heap ($x := [E]$), updating the heap $[E] := E'$, allocating new cells in the heap ($x := \mathbf{cons}(E_1, \ldots, E_m)$), and disposing cells in the heap ($\mathbf{dispose}(E)$). Functions are first order and call-by-value.

Function specifications. There is a judgment

$$\Delta \vdash \gamma\text{:FSpec}$$

stating that γ is a well-formed *function specification* in context Δ.

Function specifications are used to record assumptions about the functions used in programs. The judgment is given by

$$\frac{\Delta \vdash P\text{:Prop} \quad \Delta \vdash Q\text{:Prop}}{\Delta \vdash \{P\}\, k(x_1,\ldots,x_n)\, \{Q\}\text{:FSpec}}$$

$$\frac{\Delta \vdash \gamma\text{:FSpec} \quad \Delta \vdash \gamma'\text{:FSpec}}{\Delta \vdash \gamma \wedge \gamma'\text{:FSpec}}$$

$$\frac{\Delta, x{:}\tau \vdash \gamma\text{:FSpec}}{\Delta \vdash \natural x{:}\tau.\ \gamma\text{:FSpec}} \text{ where } \natural \in \{\exists, \forall\}.$$

The set of free variables for a function specification is defined as the free variables in the assertions occurring in it.

Specifications. We introduce syntax for commands and specifications. There is a judgment $\Delta \vdash c\text{:comm}$ which asserts that the program c is well formed in the context Δ. We omit the formal definition here.

The *specification* of higher-order separation logic is given by a judgment

$$\Delta \vdash \delta\text{:Spec}$$

which asserts that δ is a well-formed specification in the context Δ. This judgment is given by

$$\frac{\Delta \vdash c\text{:comm} \quad \Delta \vdash P\text{:Prop} \quad \Delta \vdash Q\text{:Prop}}{\Delta \vdash \{P\}\ c\ \{Q\}\text{:Spec}}$$

$$\frac{\Delta \vdash \delta\text{:Spec} \quad \Delta \vdash \delta'\text{:Spec}}{\Delta \vdash \delta \wedge \delta'\text{:Spec}}$$

$$\frac{\Delta, x{:}\tau \vdash \delta\text{:Spec}}{\Delta \vdash \natural x{:}\tau.\ \delta\text{:Spec}}\ \natural \in \{\exists, \forall\}.$$

The set $\mathrm{FV}(\delta)$ of free variables of a specification δ is the set of free variables in the assertions and variables in the commands occurring in δ. The set $\mathit{Mod}(\delta)$ of modified variables of δ is the set of modified variables in the commands occurring in δ.

Operational semantics. The operational semantics of the programming language is given by a judgment

$$(\Pi, c, s, h) \Downarrow (s', h'), \tag{12}$$

where Π is a *well-formed semantic function environment*. A semantic function environment maps function names k to pairs $(\vec{x}, c)$, where $\vec{x}$ is a vector of integer variables and c a command from the programming language. Such an environment is well formed if the function bodies only modify local variables (and *ret*, by the **return** command).

$$\Pi\ \mathbf{ok} \text{ iff } \forall (x, c) \in \mathit{cod}(\Pi).\ \mathrm{Mod}(c) = \emptyset$$

We write SemFunEnv for the set of all well-formed semantic function environments.

Intuitively, the judgment (12) says that the state (s, h) is transformed to the state (s', h') by the program c. The judgment is given by the clauses in Figure 1. We occasionally use Δ_p for the domain of s in the definition of the judgment, for example, in the second rule (for assignment). Furthermore, the notation $h - \{n\}$ is used to denote the heap which is like h, but with n taken out of its domain. In the evaluation of a function call $x = k(E)$, a designated variable ret is used to transfer the return value of the function call via the stack to x.

The configuration (Π, c, s, h) is called *safe* if $(\Pi, c, s, h) \not\Downarrow \mathit{wrong}$. A configuration may terminate in a state (s', h'), diverge, or go wrong.

Note that since this semantics is the same as the operational semantics of the language of Parkinson and Bierman [2005], the properties needed to prove the frame rule, namely, safety monotonicity and the frame property [Yang and O'Hearn 2002], are valid for all programs of the language. These properties are:

—*Safety monotonicity.* For all well-formed semantic function environments Π, programs c, stacks s, and heaps h, if (Π, c, s, h) is safe, then for all heaps h' disjoint from h, $(\Pi, c, s, h * h')$ is also safe.

—*The frame property.* For all well-formed semantic function environments Π, programs c, stacks s, and heaps h, if (Π, c, s, h) is safe and h' is disjoint from h, then $(\Pi, c, s, h * h') \Downarrow (s', h'')$ implies that there is h_0 disjoint from h' such that $h'' = h_0 * h'$ and $(\Pi, c, s, h) \Downarrow (s', h_0)$.

5.1 Program Logic Judgments

A list Γ of function specifications is called an *environment*. We shall define the judgment

$$\Delta_l; \Delta_p; \Gamma \mid\models \delta{:}\mathsf{Spec}$$

which states that in the context Δ_l used for logical variables, and context Δ_p used for program variables, given the assumptions about functions recorded in Γ, the specification δ holds. This judgment is defined in several straightforward steps. First, we give the semantics of a triple, relative to a context. The semantics of $[\![\Delta_l; \Delta_p \vdash \delta{:}\mathsf{Spec}]\!]$ is a map from SemFunEnv $\times [\![\Delta_l]\!]$ to the domain $\{\mathbf{true}, \mathbf{false}\}$, and given by (some obvious type annotations are omitted):

$$\begin{array}{rl}
[\![\Delta_l; \Delta_p \vdash \{P\}\ c\ \{Q\}]\!](\Pi, s_l) & \text{iff } \forall s_p \in [\![\Delta_p]\!]. \forall h \in [\![\Delta_l, \Delta_p \vdash P]\!](s_l, s_p). \\
& \quad - (\Pi, c, s_p, h) \text{ is safe, and} \\
& \quad - (\Pi, c, s_p, h) \Downarrow (s'_p, h') \text{ implies} \\
& \qquad h' \in [\![\Delta \vdash Q]\!](s_l, s'_p) \\
[\![\Delta_l; \Delta_p \vdash \delta \wedge \delta']\!](\Pi, s_l) & \text{iff } [\![\Delta_l; \Delta_p \vdash \delta]\!](\Pi, s_l) \text{ and } [\![\Delta_l; \Delta_p \vdash \delta']\!](\Pi, s_l) \\
[\![\Delta_l; \Delta_p \vdash \exists x{:}\tau.\ \delta]\!](\Pi, s_l) & \text{iff } [\![\Delta_l x : t; \Delta_p \vdash \delta]\!](\Pi, (s_l)_{[x \mapsto v]}) \text{ for some } v \in [\![\tau]\!] \\
[\![\Delta_l; \Delta_p \vdash \forall x{:}\tau.\ \delta]\!](\Pi, s_l) & \text{iff } [\![\Delta_l x : t; \Delta_p \vdash \delta]\!](\Pi, (s_l)_{[x \mapsto v]}) \text{ for all } v \in [\![\tau]\!].
\end{array}$$

We call $\Delta_l; \Delta_p \vdash \delta$ *valid* and write $\Delta_l; \Delta_p \mid\models \delta$ iff $[\![\Delta_l; \Delta_p; \vdash \delta]\!](\Pi, s_l) = \mathbf{true}$ for all Π and all $s_l \in [\![\Delta_l]\!]$.

$$(\Pi, \mathbf{skip}, s, h) \Downarrow (s, h) \qquad \frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n}{(\Pi, x := E, s, h) \Downarrow (s_{[x \mapsto n]}, h)}$$

$$\frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n}{(\Pi, \mathbf{return}\ E, s, h) \Downarrow (s_{[ret \mapsto n]}, h)} \qquad \frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad n \in \mathrm{dom}(h) \quad h(n) = n'}{(\Pi, x := [E], s, h) \Downarrow (s_{[x \mapsto n']}, h)}$$

$$\frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad \llbracket \Delta_p \vdash E'{:}\mathsf{Int} \rrbracket s = n' \quad n \in \mathrm{dom}(h)}{(\Pi, [E] := E', s, h) \Downarrow (s, h_{[n \mapsto n']})}$$

$$\frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad n \in \mathrm{dom}(h)}{(\Pi, \mathbf{dispose}(E), s, h) \Downarrow (s, h - \{n\})} \qquad \frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad n \notin \mathrm{dom}(h)}{(\Pi, \mathbf{dispose}(E), s, h) \Downarrow wrong}$$

$$\frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad n \notin \mathrm{dom}(h)}{(\Pi, x := [E], s, h) \Downarrow wrong} \qquad \frac{\llbracket \Delta_p \vdash E{:}\mathsf{Int} \rrbracket s = n \quad n \notin \mathrm{dom}(h)}{(\Pi, [E] := E', s, h) \Downarrow wrong}$$

$$\frac{\{n, n+1, \ldots, n+m\} \notin \mathrm{dom}(h) \quad (\llbracket \Delta_p \vdash E_i{:}\mathsf{Int} \rrbracket s = n_i)_{i=0,\ldots,m}}{(\Pi, x := \mathbf{cons}(E_0, \ldots, E_m), s, h) \Downarrow (s_{[x \mapsto n]}, h_{[n+i \mapsto n_i]_{i=0,\ldots,m}})}$$

$$\frac{(\Pi, c_1, s, h) \Downarrow (s', h') \quad (\Pi, c_2, s', h') \Downarrow (s'', h'')}{(\Pi, c_1; c_2, s, h) \Downarrow (s'', h'')}$$

$$\frac{\llbracket \Delta_p \vdash B{:}\mathsf{Bool} \rrbracket s = \mathbf{false} \quad (\Pi, c_1, s, h) \Downarrow (s', h')}{(\Pi, \mathbf{if}\ B\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1\ \mathbf{fi}) \Downarrow (s', h')}$$

$$\frac{\llbracket \Delta_p \vdash B{:}\mathsf{Bool} \rrbracket s = \mathbf{true} \quad (\Pi, c_0, s, h) \Downarrow (s', h')}{(\Pi, \mathbf{if}\ B\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1\ \mathbf{fi}) \Downarrow (s', h')}$$

$$\frac{\llbracket \Delta_p \vdash B{:}\mathsf{Bool} \rrbracket s = \mathbf{false}}{(\Pi, \mathbf{while}\ B\ \mathbf{do}\ c\ \mathbf{od}, s, h) \Downarrow (s, h)}$$

$$\frac{\llbracket \Delta_p \vdash B{:}\mathsf{Bool} \rrbracket s = \mathbf{true} \quad (\Pi, c; \mathbf{while}\ B\ \mathbf{do}\ c\ \mathbf{od}, s, h) \Downarrow (s', h')}{(\Pi, \mathbf{while}\ B\ \mathbf{do}\ c\ \mathbf{od}, s, h) \Downarrow (s', h')}$$

$$\frac{\begin{array}{c} \Pi(k) = ((x_1, \ldots, x_m), c_k) \\ (\llbracket \Delta_p \vdash E_i{:}\mathsf{Int} \rrbracket s = n_i)_{i=1,\ldots,m} \end{array} \quad (\Pi, c_k, s[x_i \mapsto n_i], h) \Downarrow (s', h')}{(\Pi, x = k(E_1, \ldots, E_m), s, h) \Downarrow (s_{[x \mapsto s'(ret)]}, h')}$$

$$\frac{(\Pi, c, s_{[x \mapsto null]}, h) \Downarrow (s', h') \quad s(x) = v}{(\Pi, \mathbf{newvar}\ x; c, s, h) \Downarrow (s'_{[x \mapsto v]}, h')}$$

$$\frac{(\Pi \cup (k_1 \mapsto ((x_1, \ldots, x_{n_1}), c_1), \ldots, k_n \mapsto ((x_1, \ldots, x_{n_k}), c_n)), c, s, h) \Downarrow (s', h')}{(\Pi, \mathbf{let}\ k_1(x_1, \ldots, x_{n_1}) = c_1, \ldots, k_n(x_1, \ldots, x_{n_n}) = c_n\ \mathbf{in}\ c, s, h) \Downarrow (s', h')}$$

Fig. 1. Operational semantics of the programming language.

LEMMA 5.1. *Let δ be a specification, $x{:}\tau$ a variable, and $\Delta_l \vdash t{:}\tau$ a term such that $(\mathrm{FV}(t) \cup \{x\}) \cap Mod(\delta) = \emptyset$. Further, let $s_l \in \llbracket \Delta_l \rrbracket$, and Π be well formed. Then,*

$$\llbracket \Delta_l ; \Delta_p \vdash \delta[t/x] \rrbracket(\Pi, s_l) \textit{ iff } \llbracket \Delta_l ; \Delta_p, x{:}\tau \vdash \delta \rrbracket(\Pi, (s_l)_{[x \mapsto v]}),$$

where $v = \llbracket \Delta_l \vdash t{:}\tau \rrbracket s_l$.

There is a similar semantics for function specifications. This semantics is a map

$$[\![\Delta_l; \Delta_p \vdash \gamma{:}\mathsf{FSpec}]\!] : \text{SemFunEnv} \times [\![\Delta_l]\!] \mapsto \{\mathbf{true}, \mathbf{false}\}$$

and given in much the same way as the corresponding map for specifications. The only difference is the base case, which is given by

$$\begin{aligned}[\![\Delta_l; \Delta_p \vdash \{P\}\, k_m\, \{Q\}]\!](\Pi, s_l) \ \text{ iff } \ & [\![\Delta_l; \Delta'_p \vdash \{P\}\, c_m\, \{Q\}]\!](\Pi, s_l) \\ & \text{where } \Pi(k_m) = ((x_1, \ldots, x_{n_m}), c_m),\end{aligned}$$

where Δ'_p is Δ_p with the x_i's added (with type Int).

As mentioned, an environment is a list of function specifications. The semantics of an environment is given componentwise.

$$[\![\Delta_l; \Delta_p \vdash \Gamma]\!](\Pi, s_l) \quad \text{iff} \quad [\![\Delta_l; \Delta_p \vdash \gamma]\!](\Pi, s_l) \text{ for all } \gamma \in \Gamma$$

Finally, the semantics of specifications, relative to a context and an environment, is defined by

$$\begin{aligned}\Delta_l; \Delta_p; \Gamma \models \delta \ \text{ iff } \ & \text{for all well-formed } \Pi \text{ and all } s_l \in [\![\Delta_l]\!], \\ & [\![\Delta_l; \Delta_p \vdash \Gamma]\!](\Pi, s_l) \text{ implies } [\![\Delta_l; \Delta_p \vdash \delta]\!](\Pi, s_l).\end{aligned}$$

5.2 Inference Rules

We define a judgment

$$\Delta_l; \Delta_p; \Gamma \vdash \delta$$

for deriving valid specifications. The inference rules are given in Figure 2. For brevity, we have omitted obvious rules for conjunctions of specifications and some structural rules for weakening and strengthening of variable contexts. We first explain some of the rules at an intuitive level, and then show soundness.

5.3 Informal Explanation of Rules

The first two rules are the usual ones for **skip** and assignment from Hoare logic. The rule for **return** is similar to that for assignment, since **return** simply amounts to an assignment to the special variable ret.

The rule

$$\frac{\{P\}\, k(\vec{x})\, \{Q\} \in \Gamma}{\Delta_l; \Delta_p; \Gamma \vdash \{P[\vec{E}/\vec{x}]\}\ y := k(\vec{E})\ \{Q[\vec{E}, y/\vec{x}, ret]\}}, \, \text{Y} \notin \text{FV}(Q) \cup \text{FV}(E)$$

for a function call says that in order to call a function, the precondition for the function must be satisfied. This precondition is recorded in the environment, along with the corresponding postcondition.

The next four rules, which involve the heap-manipulating constructs of the programming language, are the standard rules of separation logic, adapted to our setting. Note that the specifications are "tight" in the sense that they only mention those heap cells that are actually manipulated by the commands. For

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\mathbf{skip}\{P\}} \quad \frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{P[E/x]\}\ x := E\ \{P\}}$$

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{P[E/ret]\}\ \mathbf{return}\ E\ \{P\}} \quad \frac{\{P\}\ k(\vec{x})\ \{Q\} \in \Gamma}{\Delta_l; \Delta_p; \Gamma \vdash \{P[\vec{E}/\vec{x}]\}\ y = k(\vec{E})\ \{Q[\vec{E}, y/\vec{x}, ret]\}},\ \mathrm{Y} \notin \mathrm{FV}(Q) \cup \mathrm{FV}(E)$$

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{\mathsf{emp} \wedge x = m\}x := \mathbf{cons}(E_1, \ldots, E_n)\{x \mapsto E_1[m/x], \ldots, E_n[m/x]\}}$$

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{E \mapsto -\}\mathbf{dispose}(E)\{\mathsf{emp}\}}$$

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{E \mapsto n \wedge x = m\}x := [E]\{E[m/x] \mapsto n \wedge x = n\}}$$

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{E \mapsto -\}[E] := E'\{E \mapsto E'\}}$$

$$\frac{\begin{array}{c}\Delta_l; \Delta_p, \vec{x}_1; \Gamma \vdash \{P_1\}\ c_1\ \{Q_1\} \\ \vdots \\ \Delta_l; \Delta_p, \vec{x}_n; \Gamma \vdash \{P_n\}\ c_n\ \{Q_n\} \\ \Delta_l; \Delta_p; \Gamma, \{P_1\}\ k_1(\vec{x}_1)\ \{Q_1\}, \cdots, \{P_n\}\ k_n(\vec{x}_n)\ \{Q_n\} \vdash \{P\}\ c\ \{Q\}\end{array}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{let}\ k_1(\vec{x}_1) = c_1, \ldots, k_n(\vec{x}_n) = c_n\ \mathbf{in}\ c\ \{Q\}}$$

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ c_1\ \{P'\} \quad \Delta_l; \Delta_p; \Gamma \vdash \{P'\}\ c_2\ \{Q\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ c_1; c_2\ \{Q\}}$$

$$\frac{\Delta_l; \Delta_p, x{:}\mathsf{Int}; \Gamma \vdash \{P \wedge x = null\}\ c\ \{Q\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{newvar}\ x\ \mathbf{in}\ c\ \mathbf{end}\ \{Q\}}\ x \notin \mathrm{FV}(P, Q, \Gamma)$$

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{P \wedge B\}\ c_1\{Q\} \quad \Delta_l; \Delta_p; \Gamma \vdash \{P \wedge \neg B\}\ c_2\{Q\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{if}\ B\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2\ \mathbf{fi}\ \{Q\}}$$

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{P \wedge B\}\ c\ \{P\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{while}\ B\ \mathbf{do}\ c\ \mathbf{od}\ \{P \wedge \neg B\}}$$

$$\frac{\Delta_l; \Delta_p \vdash P \Rightarrow P' \quad \Delta_l; \Delta_p; \Gamma \vdash \{P'\}\ c\ \{Q'\} \quad \Delta_l; \Delta_p \vdash Q' \Rightarrow Q}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ c\ \{Q\}}$$

$$\frac{\Delta_l, x{:}\tau; \Delta_p;\ \Gamma, \gamma \vdash \delta}{\Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.\ \gamma \vdash \delta}\ x \notin \mathrm{FV}(\Gamma, \delta)$$

$$\frac{\Delta_l, x{:}\tau; \Delta_p\ \Gamma \vdash \delta}{\Delta_l; \Delta_p; \Gamma \vdash \forall x{:}\tau.\ \delta}\ x \notin \mathrm{FV}(\Gamma)$$

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ c\ \{Q\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P * P'\}\ c\ \{Q * P'\}}\ \mathrm{Mod}(c) \cap \mathrm{FV}(P') = \emptyset$$

Fig. 2. Program logic.

example, the rule

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{\mathsf{emp} \wedge x = m\}x := \mathbf{cons}(\vec{E})\{x \mapsto \vec{E}[m/x]\}}$$

for **cons** produces a new cell when run in an empty heap. Note that this does *not* mean that **cons** can only be executed in an empty heap. The last rule of the system

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ c\ \{Q\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P * P'\}\ c\ \{Q * P'\}}\ \mathrm{Mod}(c) \cap \mathrm{FV}(P') = \mathtt{emp},$$

called the *frame rule*, implies that one can infer a *global* from a *local* specification, like the one for **cons**. Hence, **cons** can be executed in *any* heap, described by the predicate P (in which x does not occur freely), by the following instance of the frame rule.

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \{\mathsf{emp} \wedge x = m\} x := \mathbf{cons}(\vec{E}) \{x \mapsto \vec{E}[m/x]\}}{\Delta_l; \Delta_p; \Gamma \vdash \{P \wedge x = m\} x := \mathbf{cons}(\vec{E}) \{P * (x \mapsto \vec{E}[m/x])\}}$$

The rule

$$\frac{\begin{array}{c} \Delta_l; \Delta_p, \vec{x}_1; \Gamma \vdash \{P_1\}\ c_1\ \{Q_1\} \\ \vdots \\ \Delta_l; \Delta_p, \vec{x}_n; \Gamma \vdash \{P_n\}\ c_n\ \{Q_n\} \\ \Delta_l; \Delta_p; \Gamma, \{P_1\}\ k_1(\vec{x}_1)\ \{Q_1\}, \cdots, \{P_n\}\ k_n(\vec{x}_n)\ \{Q_n\} \vdash \{P\}\ c\ \{Q\} \end{array}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{let}\ k_1(\vec{x}_1) = c_1, \ldots, k_n(\vec{x}_n) = c_n\ \mathbf{in}\ c\ \{Q\}}$$

for function definitions is the usual one from Hoare logic with procedures [Hoare 1971]. The rules for **while** and **if-then-else** are also standard. The rule of consequence is standard, and the rules

$$\frac{\Delta_l, x{:}\tau; \Delta_p;\ \Gamma, \gamma \vdash \delta}{\Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.\ \gamma \vdash \delta}\ x \notin \mathrm{FV}(\Gamma)$$

$$\frac{\Delta_l, x{:}\tau; \Delta_p;\ \Gamma \vdash \delta}{\Delta_l; \Delta_p; \Gamma \vdash \forall x{:}\tau.\ \delta}\ x \notin \mathrm{FV}(\Gamma)$$

are straightforward adaptations of standard rules of predicate logic. (Note that by the convention that variables in contexts $\Delta_l; \Delta_p$ are all distinct, $x \notin \mathrm{FV}(\delta)$ in the first rule and $x \notin \mathrm{FV}(\Gamma)$ in the second.) They are used later for reasoning about data abstraction. Note here that x may be of any type τ, including higher types for predicates (see the examples in Sections 6 and 7).

5.4 Soundness

THEOREM 5.2. *If a specification*

$$\Delta_l; \Delta_p; \Gamma \vdash \delta$$

can be derived from the rules in Figure 2, *then it is valid.*

PROOF. By induction. For each rule of form

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \delta}{\Delta_l'; \Delta_p'; \Gamma' \vdash \delta'}\ , \tag{13}$$

we check that $\Delta_l'; \Delta_p'; \Gamma' \models \delta'$, under the assumption $\Delta_l; \Delta_p; \Gamma \models \delta$. For axioms of the form

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \delta}\ ,$$

the proof obligation is to show $\Delta_l; \Delta_p; \Gamma \models \delta$.

Consider the rule for **skip**:

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{skip}\ \{P\}}$$

Although trivial, we show soundness of this rule to exercise the definitions. Let Π be a well-formed semantic function environment. It suffices to show that

$$[\![\Delta_l; \Delta_p \vdash \{P\} \textbf{ skip } \{P\}]\!](\Pi, s_l)$$

for all $s_l \in [\![\Delta_l]\!]$. Let $s_p \in [\![\Delta_p]\!]$ and let $h \in [\![P]\!](s_l, s_p)$. Then,

$$(\Pi, \textbf{skip}, s_p, h) \Downarrow (s_p, h)$$

and clearly, $h \in [\![P]\!](s_l, s_p)$, so this rule is sound.

The soundness of the rule for assignment

$$\frac{}{\Delta_l; \Delta_p; \Gamma \vdash \{P[E/x]\}\ x := E\ \{P\}}$$

depends, as usual, on the standard substitution lemma for assertions (not included in the review in Section 3).

Now consider the rule for function calls.

$$\frac{\{P\}\ k_i(x_1, \ldots, x_{n_i})\ \{Q\} \in \Gamma}{\Delta_l; \Delta_p; \Gamma \vdash \{P[E_1/x_1 \cdots E_{n_i}/x_{n_i}]\}\ y = k_i(E_1, \ldots, E_{n_i})\ \{Q[E_1/x_1 \cdots E_{n_i}/x_{n_i}, y/ret]\}}$$

To show soundness, suppose $\{P\}\ k_i(x_1, \ldots, x_{n_i})\ \{Q\} \in \Gamma$. Let $s_l \in [\![\Delta_l]\!]$, and let Π be a well-formed semantic function environment with $[\![\Delta_l; \Delta_p \models \Gamma]\!](\Pi, s_l)$. In particular,

$$[\![\Delta_l; \Delta_p \vdash \{P\}\ k_i(x_1, \ldots, x_{n_i})\ \{Q\}]\!](\Pi, s_l),$$

so if $\Pi(k_i) = ((x_1, \ldots, x_{n_i}), c_i)$, then $[\![\Delta_l; \Delta_p \vdash \{P\}\ c_i\ \{Q\}]\!](\Pi, s_l)$. Now, suppose that $s_p \in [\![\Delta_p]\!]$ and

$$h \in [\![P[E_1/x_1 \cdots E_{n_i}/x_{n_i}]]\!](s_l, s_p) = [\![P]\!](s_l, (s_p)_{[x_1 \mapsto v_1, \ldots, x_{n_i} \mapsto v_{n_i}]}),$$

where $v_j = [\![\Delta_l; \Delta_p \vdash E_j{:}\mathsf{Int}]\!](s_l, s_p)$ and $j \in \{1, \ldots, n_i\}$, by the substitution lemma. This means that if

$$(\Pi, c_i, (s_p)_{[x_1 \mapsto v_1, \ldots, x_{n_i} \mapsto v_{n_i}]}, h) \Downarrow (s'_p, h'),$$

then $h' \in [\![Q]\!](s_l, s'_p)$. Since Π is well formed, c_i does not modify any variables, so s'_p is of the form

$$s'_p = (s_p)_{[x_1 \mapsto v_1, \ldots, x_{n_i} \mapsto v_{n_i}, ret \mapsto s'(ret)]}$$

and by the substitution lemma, $h' \in [\![Q[E_1/x_1 \cdots E_{n_i}/x_{n_i}, s'_p(ret)/ret]]\!](s_l, s_p)$. By the operational semantics for function calls,

$$(\Pi, y = k_i(E_1, \ldots, E_{n_i}), s_p, h) \Downarrow ((s_p)_{[y \mapsto s'_p(ret)]}, h')$$

and thus, the rule holds.

The first rule for existentials is

$$\frac{\Delta, x{:}\tau; \Delta_p; \Gamma, \gamma \vdash \delta}{\Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.\ \gamma \vdash \delta}\ x \notin \mathrm{FV}(\Gamma).$$

Suppose that for all well-formed Π and $s_l \in [\![\Delta_l, x{:}\tau]\!]$,

$$[\![\Delta_l, x{:}\tau; \Delta_p \vdash \Gamma, \gamma]\!](\Pi, s_l) \text{ implies} [\![\Delta_l, x{:}\tau; \Delta_p \vdash \delta]\!](\Pi, s_l)$$

and let $[\![\Delta_l; \Delta_p \vdash \Gamma]\!](\Pi, s_l)$ and $[\![\Delta_l; \Delta_p \vdash \exists x{:}\tau.\ \gamma]\!](\Pi, s)$. This means that $[\![\Delta_l, x{:}\tau; \Delta_p \vdash \gamma]\!](\Pi, (s_l)_{[x \mapsto v]})$ for some $v \in [\![\tau]\!]$. Since $x \notin \mathrm{FV}(\Gamma)$, $[\![\Delta_l, x{:}\tau; \Delta_p \vdash$

$\Gamma]\!](\Pi, (s_l)_{[x\mapsto v]})$. This implies $[\![\Delta_l, x{:}\tau; \Delta_p \vdash \delta]\!](\Pi, (s_l)_{[x\mapsto v]})$, and since $x \notin \mathrm{FV}(\delta)$, we have $[\![\Delta_l; \Delta_p \vdash \delta]\!](\Pi, s)$.

The other rule for existentials is

$$\frac{\Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.\ \gamma \vdash \delta}{\Delta_l, x{:}\tau; \Delta_p; \Gamma, \gamma \vdash \delta}\ .$$

For soundness, first suppose τ is inhabited and that for all well-formed Π and $s_l \in [\![\Delta_l]\!]$,

$$[\![\Delta_l; \Delta_p \vdash \Gamma, \exists x{:}\tau.\ \gamma]\!](\Pi, s_l) \text{ implies } [\![\Delta_l; \Delta_p \vdash \delta]\!](\Pi, s_l)$$

and suppose $[\![\Delta_l, x{:}\tau; \Delta_p \vdash \Gamma, \gamma]\!](\Pi, s_l)$. Since τ is inhabited, this means that

$$[\![\Delta_l, x{:}\tau; \Delta_p \vdash \Gamma, \gamma]\!](\Pi, (s_l)_{[x\mapsto s_l(x)]})$$

and since $x \notin \mathrm{FV}(\Gamma)$, this implies

$$[\![\Delta_l, x{:}\tau; \Delta_p \vdash \Gamma, \exists x{:}\tau.\ \gamma]\!](\Pi, s_l)$$

and thus, $[\![\Delta_l; \Delta_p \vdash \delta]\!](\Pi, s_l)$, as desired. If τ is an empty type, one can make an easy case analysis on whether x occurs in γ.

Soundness of the downwards rule for universals is easy. For soundness of the upwards rule,

$$\frac{\Delta_l; \Delta_p; \Gamma \vdash \forall x{:}\tau.\ \delta}{\Delta_l, x{:}\tau; \Delta_p; \Gamma \vdash \delta}\ ,$$

suppose that for all well-formed Π and $s_l \in [\![\Delta_l]\!]$,

$$[\![\Delta_l; \Delta_p \vdash \Gamma]\!](\Pi, s_l) \text{ implies } [\![\Delta_l; \Delta_p \vdash \forall x{:}\tau.\ \delta]\!](\Pi, s_l)$$

and let $s_l' \in [\![\Delta_l, x{:}\tau]\!]$. Suppose $[\![\Delta_l, x{:}\tau; \Delta_p \vdash \Gamma]\!](\Pi, s_l')$. Since $x \notin \mathrm{FV}(\Gamma)$,

$$[\![\Delta_l; \Delta_p \vdash \Gamma]\!](\Pi, (s_l' - x)),$$

and this implies

$$[\![\Delta_l, x{:}\tau; \Delta_p \vdash \delta]\!](\Pi, (s_l' - x)_{[x\mapsto v]}), \text{ for all } v \in [\![\tau]\!].$$

This means in particular that

$$[\![\Delta_l, x{:}\tau; \Delta_p \vdash \delta]\!](\Pi, (s_l')_{[x\mapsto s_l'(x)]}),$$

which shows the desired result. □

5.5 A Derived Rule

There is an important rule *abstract function definition* that is derivable from the rules in Figure 2. The rule is

$$\frac{\begin{array}{c}\Delta_l \vdash \hat{P}{:}\tau \\ \Delta_l; \Delta_p, \vec{x}_1; \Gamma \vdash \{P_1[\hat{P}/x]\}\ c_1\ \{Q_1[\hat{P}/x]\} \\ \vdots \\ \Delta_l; \Delta_p, \vec{x}_n; \Gamma \vdash \{P_n[\hat{P}/x]\}\ c_n\ \{Q_n[\hat{P}/x]\} \\ \Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.(\{P_1\}k_1(\vec{x}_1)\{Q_1\} \wedge \cdots \wedge \{P_n\}k_n(\vec{x}_n)\{Q_n\}) \vdash \{P\}\ c\ \{Q\}\end{array}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{let}\ k_1(\vec{x}_1) = c_1, \ldots, k_n(\vec{x}_n) = c_n\ \mathbf{in}\ c\ \mathbf{end}\ \{Q\}} \quad (14)$$

$$x \notin \mathrm{FV}(\{P\}\ c\ \{Q\}).$$

Here one may think of x as a predicate describing a resource invariant used by an abstract data type with operations $k_1, \ldots k_n$.

We show how this rule can be derived; for simplicity, we assume $n = 1$ and that there are no parameters. The proof of the more general case is essentially the same. First, by the function definition rule,

$$\frac{\begin{array}{c}\Delta_l; \Delta_p, y; \Gamma \vdash \{P_1[\hat{P}/x]\}\ c_1\ \{Q_1[\hat{P}/x]\} \\ \Delta_l; \Delta_p; \Gamma, \{P_1[\hat{P}/x]\}\ k_1(y)\ \{Q_1[\hat{P}/x]\} \vdash \{P\}\ c\ \{Q\}\end{array}}{\Delta_l; \Delta_p; \Gamma \vdash \{P\}\ \mathbf{let}\ k_1(y) = c_1\ \mathbf{in}\ c\ \{Q\}}\ .$$

The rule for existentials gives us

$$\frac{\Delta_l; \Delta_p; \Gamma, \exists x{:}\tau.\ \{P_1\}\ k_1(y)\ \{Q_1\} \vdash \{P\}\ c\ \{Q\}}{\Delta_l, x{:}\tau; \Delta_p; \Gamma, \{P_1\}\ k_1(y)\ \{Q_1\} \vdash \{P\}\ c\ \{Q\}}\ ,$$

so we need to establish

$$\Delta_l; \Delta_p; \Gamma, \{P_1[\hat{P}/x]\}\ k_1(y)\ \{Q_1[\hat{P}/x]\} \vdash \{P\}\ c\ \{Q\},$$

given

$$\Delta_l; x{:}\tau; \Delta_p; \Gamma, \{P_1\}\ k_1(y)\ \{Q_1\} \vdash \{P\}\ c\ \{Q\}.$$

But this follows from a substitution lemma, since x is not free in $\{P\}\ c\ \{Q\}$.

6. DATA ABSTRACTION VIA EXISTENTIAL QUANTIFICATION

We present an example that demonstrates how one may use the program logic for reasoning using data abstraction. The example involves two implementations of a priority queue, and the intention is, of course, that the client programs which use these implementations should be unaware of and unable to exploit details of the particular implementation used. Data abstraction is modeled via existential quantification over predicates, corresponding to the slogan "abstract types have existential type" [Mitchell and Plotkin 1985].

6.1 Reasoning using Abstract Priority Queues

Priority queues are used frequently in programming, for example, in scheduling algorithms for processes in operating systems [Silberschatz and Galvin 1998]. These consist of pairs (p, v), where v is a stored value and p is the *priority* associated with v. In such a structure, one can then enqueue such pairs and extract an element with the highest priority. Some operations and relations on such queues are needed:

$$\begin{aligned}
&\mathsf{MaxPri}(\varepsilon) = -1\\
&\mathsf{MaxPri}((p, v) \cup Q) = \mathsf{Max}(p, \mathsf{MaxPri}(Q))\\
&\mathsf{MaxPair}(Q, (p, v)) \Leftrightarrow (p, v) \in Q \wedge p = \mathsf{MaxPri}(Q)
\end{aligned}$$

We assume a base type PriQ, whose values are priority queues. These types and operations are only used in the logic, *not* in programs. Observe that the type PriQ is, of course, definable in the higher-order logic.

We now discuss how to reason about client code which uses an abstract priority queue. First, since client programs cannot modify abstract values, we'll use a predicate stating that there is a "handle" to a priority queue. Hence, we introduce the predicate

$$\mathsf{repr}(q, Q)$$

which asserts that the integer denoted by q is a handle to the priority queue Q, but does *not* say anything about how Q is represented. Note that the type of repr is $(\mathsf{Int} \times \mathsf{PriQ}) \Rightarrow \mathsf{Prop}$, a type of predicate.

This will be used as an abstract predicate in our proofs (thus playing the role of x in the abstract function definition rule (14)). Given this predicate, the following are reasonable specifications for the various operations on a priority queue.

Creating a queue. There should be an operation which enables a client program to create a priority queue. Its specification is

$$\{\mathsf{emp}\}\ \mathbf{createqueue}()\ \{\mathsf{repr}(ret, \varepsilon)\},$$

which merely states that upon creation of a queue, a handle to an empty priority queue is returned.

Enqueing. There should be an operation for storing elements in a queue. The specification is

$$\{\mathsf{repr}(q, Q) * v \mapsto \_\}\ \mathbf{enqueue}(q, (p, v))\ \{\mathsf{repr}(q, (p, v) \cup Q)\}.$$

Note that ownership of the cell pointed to by v transfers from the client to the module.

Dequeing. There should be an operation for dequeing. We make sure not to dequeue from an empty queue via the specification

$$\begin{aligned}
&\{\mathsf{repr}(q, Q) \wedge Q \neq \varepsilon\}\\
&\quad \mathbf{dequeue}(q)\\
&\{\exists Q', p, v.(\ \mathsf{repr}(q, Q') \wedge Q = (p, v) \uplus Q' \wedge \mathsf{MaxPair}(Q, (p, v)) \wedge ret = v)\\
&\quad * v \mapsto \_\}.
\end{aligned}$$

Note that the ownership of the dequeued cell is now transferred back to the client.

Disposing a queue. The specification for disposing a queue is

$$\{\mathsf{repr}(q, Q)\}\ \mathbf{disposequeue}(q)\ \{\mathsf{emp}\}.$$

We can now show a specification for a client program c using the abstract specification of the priority queue.

$$\begin{array}{l}\exists \mathsf{repr} : (\mathsf{PriQ} \times \mathsf{Int}) \Rightarrow \mathsf{Prop}.\\ \quad \{\mathsf{emp}\}\ \mathbf{createqueue}()\ \{\mathsf{repr}(ret, \varepsilon)\} \wedge \\ \quad \ldots \\ \quad \{\mathsf{repr}(q, Q)\}\ \mathbf{disposequeue}(q)\ \{\mathsf{emp}\}\\ \vdash \\ \qquad \{P_c\}c\{Q_c\}\end{array}$$

Observe that a client may use multiple instances of priority queues, unlike in O'Hearn et al. [2004], which only considers static modularity.

6.2 Implementations of Priority Queues

One can implement priority queues in many ways. We have verified two implementations: one using sorted linked lists and the other doubly-linked lists. The implementations and proofs make use of some of the properties shown by Reynolds [2002], are fairly standard, and thus omitted. Of course, a client may use either of the two implementations, and we expect that the behavior of a client is independent of which implementation of priority queues is used. The simple model we have devised in this article cannot be used to prove this formally; for that we would need a relationally parametric model.

7. SOME APPLICATIONS OF UNIVERSAL QUANTIfiCATION

In the previous section we saw how to use existential quantification over predicates to reason using data abstraction. In this section we present two examples of how to apply *universal* quantification over predicates (in addition to the examples involving fixed points in Section 4.3).

7.1 Polymorphic Types via Universal Quantification

We show that universally quantified predicates may be used to prove correct polymorphic operations on polymorphic data types.

The queue module example from O'Hearn et al. [2004] is parametric in a predicate P *at the metalevel*. We show that in higher-order separation logic, the parameterization may be expressed *in the logic*. To that end, consider the following version of the parametric list predicate from O'Hearn et al. [2004].

$$\mathsf{list}(P, \beta, i) = \begin{cases} i = null \wedge \mathsf{emp} & \text{if } \beta = \varepsilon \\ \exists j.\, i \mapsto x, j * P(x) * \mathsf{list}(P, \beta', j) & \text{if } \beta = \langle x\rangle \cdot \beta' \end{cases}$$

The predicate P is required to hold for each element of the sequence β involved. Different instantiations of P yield different versions of the list, with different amounts of data stored in the list. If $P \equiv \mathsf{emp}$, then plain values are stored (i.e., no ownership transfer to the queue module in O'Hearn et al. [2004]), and if $P \equiv x \mapsto -, -$, then the addresses of cells are stored in the queue (i.e., ownership of the cells is tranferred in and out of the queue [O'Hearn et al. 2004]).

Returning to our higher-order separation logic, the definition of list may be formalized with

$$i : \mathsf{Int}, \beta : \mathsf{seqInt}, P : \mathsf{Prop}^{\mathsf{Int}} \vdash \mathsf{list}(P, \beta, i) : \mathsf{Prop}.$$

Here we have used a type seqInt of sequences of integers which is easily definable in higher-order separation logic, and the definition of $\mathsf{list}(P, \beta, i)$ can be given by induction on β in the logic.

Suppose **listRev** is the list reversal program given in the Introduction of Reynolds [2002]. Then one can easily show the specification

$$\{\mathsf{list}(P, \beta, i)\}\ \mathbf{listRev}\ \{\mathsf{list}(P, \beta^\dagger, j)\}.$$

By the introduction rule for universal quantification, we obtain the specification

$$\beta : \mathsf{seqInt} \vdash \forall P : \mathsf{Prop}^{\mathsf{Int}}.\ \{\mathsf{list}(P, \beta, i)\}\ \mathbf{listRev}\ \{\mathsf{list}(P, \beta^\dagger, j)\}$$

which expresses that **listRev** is *parametric* in the sense that it, roughly speaking, reverses singly-linked lists uniformly, independently of how much heap storage is used for each element of the list.

Thus we have *one* parametric correctness proof of a specification for **listRev**, which may then be used to prove correct different applications of **listRev** (to lists of different types).

For such parametric operations on polymorphic data types to be really useful, one would of course prefer a higher-order programming language instead of the first-order language considered here. Then one could, for example, program the usual **map** function on lists, and provide a single parametric correctness proof for it. See our joint paper with Yang [Birkedal et al. 2005] for a proposal of separation logic for a higher-order language.

7.2 Invariance

In this subsection we briefly consider an example, suggested to us by John Reynolds, which demontrates that one may use universal quantification to specify that a command does not modify its input state. We disregard stacks here, since they are not important for the argument.

Suppose that our intention is to specify that some command c takes any heap h described by a prediate q, and produces a heap (we assume for simplicity that c terminates) which is an extension of h. We might attempt to use a specification of the form

$$\{q\}\ c\ \{q' * q\}. \tag{15}$$

This does not work, however, unless q is *strictly exact* [Reynolds 2002], that is, uniquely describes the heaps satisfying q (e.g., if q is $\exists\beta{:}\mathsf{seqInt}.\ \mathsf{list}(\mathtt{emp}, \beta, i)$, then c may delete some elements from the list in the input heap h).

Instead, we may use the specification

$$\forall p{:}\mathsf{Prop}.\{q \wedge p\}\ c\ \{q' * p\}, \tag{16}$$

as we see by the following argument. Predicate q describes a set of heaps $[\![q]\!]$. For each $h \in [\![q]\!]$, let $p_h = \{h\}$. Suppose that c terminates in heap h'. Then $h' = h_1 * h$, for some h_1. In other words, the heap h is *invariant* under the execution of c, as intended.

8. RELATED AND FUTURE WORK

We have introduced the notion of a BI hyperdoctrine and showed that it soundly and completely models intuitionistic and classical first- and higher-order BI. We showed that the semantics for BI given by separation logic is an instance of our class of models, and that interesting models for higher-order predicate BI cannot exist in toposes. Several applications of higher-order BI in program proving, and particularly separation logic, were illustrated. Specifically, we introduced higher-order separation logic, and gave sound reasoning principles for data abstraction in the presence of mutable pointer structures, using existential quantification over predicates.

The idea of using data abstraction to reason about complex data structures goes back to Hoare [1972], who introduced the idea of using abstraction functions, namely, functions that map object structures to values of an abstract domain. Modifications of object structures can then be described in terms of their abstract values, which makes implementation-independent specifications possible. Hoare's idea has been extended and applied in a variety of contexts (see, e.g., Leavans [1988], Liskow and Guttag [1986], Leino [1995], Müller [2002], Leino and Müller [2004, 2006], Barnett et al. [2003], Barnett and Naumann [2004], and Naumann and Barnett [2006]). In several of these papers, abstraction functions are captured via so-called *model fields* and the data abstraction technique is combined with ownership-based invariants to deal with mutable pointer structures. The model fields correspond very closely to (some of) the arguments of our existentially quantified propositions, for example, the PriQ argument of the repr predicate in Section 6.1. We believe that our approach to data abstraction using standard higher-order existential quantification gives a particularly clear account of data abstraction by employing standard logical notions, rather than introducing additional new logical concepts. One could argue, however, that our logical approach to data abstraction comes at the price that we move to higher-order logic, which poses difficulties for tool support. More research is needed to evaluate how much of an issue this is in practice. More research is also needed to evaluate how useful our approach is for practical verification; the examples we have considered in this article merely serve to show that the approach is viable. In particular, it would be interesting to extend the presented specification logic to richer programming languages with more of the features found in modern programming languages. We are currently investigating extensions to higher-order programming languages [Nanevski et al. 2006; Krishnaswami et al. 2006] and hope in the future to extend it to object-oriented languages.

In other work, we extended separation logic to a higher-order language [Birkedal et al. 2005], a version of Algol with immutable variables and a first-order heap. The system in *loc. cit.* doesn't distinguish the type system from the specification language: Command types can contain pre- and postconditions written in separation logic in a fashion similar to refinement types. The assertion logic is first order (i.e., no quantification over propositions) but includes a powerful kind of hypothetical frame rule, extending the second-order frame rule of O'Hearn et al. [2004] to higher order. We have worked out a simple translation from hypothetical frame rules to higher-order separation logic, which suggests that all uses of hypothetical frame rules can be represented in higher-order separation logic, but more work is needed to properly analyze this conjecture.

As mentioned in Section 6.2, we expect that one should be able to show that clients cannot detect any differences between different implementations of abstract data types. Such representation independence (i.e., relational parametricity) results have been shown for a Java-like language and for a semantic notion of confinement by Banerjee and Naumann [2005a, b]. It is quite challenging to develop relationally parametric models for separation logic, even for a simple first-order programming language like the one considered in this article. The reason is that standard models of separation logic allow location identities to be observed in the model. This means, in particular, that allocation of new heap cells is not parametric because the location identity of the allocated cell can be observed in the model. In very recent work, the second author and Yang did, however, succeed in defining a relationally parametric model of separation logic [Birkedal and Yang 2006]. However, the model in *loc. cit.* was only developed for a first-order logic with hypothetical frame rules, and thus it is still an open question how to devise a relationally parametric model for higher-order separation logic.

APPENDIX

A. PROOF OF PROPOSITION 2.8

For a term t with $y{:}Y \vdash t(y){:}X$, we add the abbreviation

$$\exists_t.\ \varphi(y) \overset{def}{=} \exists y{:}Y.\ t(y) = x \wedge \varphi(y).$$

The following rule can be deduced:

$$\frac{x{:}X \mid \exists_t.\ \varphi(y) \vdash \psi(x)}{y{:}Y \mid \varphi(y) \vdash \psi[t(y)/x]}$$

In particular, for $y{:}\{x{:}X \mid \varphi\} \vdash o(y){:}X$ we have

$$\frac{x{:}X \mid \exists_o.\ \theta(y) \vdash \psi(x)}{y{:}\{x{:}X \mid \varphi\} \mid \theta(y) \vdash \varphi[o(y)/x]}\ .$$

Let φ, ψ, ψ', χ be formulas in a context $\{x{:}X\}$ (for simplicity we just assume one free variable, the general case is similar). First we show that

$$x{:}X \mid \varphi \wedge \psi \dashv\vdash \exists_o.\ \psi[o(y)/x]. \qquad (17)$$

This is done by

$$\begin{array}{c} x{:}X \mid \exists_o.\ \psi[o(y)/x] \vdash \exists_o.\ \psi[o(y)/x] \\ \hline\hline y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] \vdash (\exists_o.\ \psi[o(y)/x])[o(y)/x] \\ \hline\hline x{:}X \mid \psi \wedge \varphi \vdash \exists_o.\ \psi[o(y)/x] \end{array} ,$$

where the last derivation is the rule for full subset types. For the other direction, consider

$$\begin{array}{c} y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] \vdash \psi[o(y)/x] \\ \hline\hline x{:}X \mid \exists_o.\ \psi[o(y)/x] \vdash \psi \end{array}$$

and

$$\begin{array}{c} x{:}X \mid \varphi \wedge \psi \vdash \varphi \\ \hline\hline y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] \vdash \varphi[o(y)/x] \\ \hline\hline x{:}X \mid \exists_o.\ \psi[o(y)/x] \vdash \varphi \end{array} ,$$

which imply that $x{:}X \mid \exists_o.\ \psi[o(y)/x] \vdash \varphi \wedge \psi$. We also need the following:

$$\begin{array}{c} y{:}\{x{:}X \mid \varphi\} \mid \chi[o(y)/x] \vdash \psi[o(y)/x] \\ \hline x{:}X \mid \exists_o.\ \chi[o(y)/x] \vdash \exists_o.\ \psi[o(y)/x] \end{array} , \qquad (18)$$

which is shown by

$$\begin{array}{c} y{:}\{x{:}X \mid \varphi\} \mid \chi[o(y)/x] \vdash \psi[o(y)/x] \\ \hline\hline x{:}X \mid \chi \wedge \varphi \vdash \psi \\ \hline x{:}X \mid \chi \wedge \varphi \vdash \psi \wedge \varphi \\ \hline\hline x{:}X \mid \exists_o.\ \chi[o(y)/x] \vdash \exists_o.\ \psi[o(y)/x] \end{array} ,$$

where the last derivation follows from Eq. (17). We then have

$$\begin{array}{c} y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] * \psi'[o(y)/x] \vdash \chi[o(y)/x] \\ \hline y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] \vdash \psi'[o(y)/x] \mathrel{-\!\!*} \chi[o(y)/x] \end{array} ,$$

namely,

$$\begin{array}{c} y{:}\{x{:}X \mid \varphi\} \mid (\psi * \psi')[o(y)/x] \vdash \chi[o(y)/x] \\ \hline y{:}\{x{:}X \mid \varphi\} \mid \psi[o(y)/x] \vdash (\psi' \mathrel{-\!\!*} \chi)[o(y)/x] \end{array} .$$

By (18) we then get

$$\begin{array}{c} x{:}X \mid \exists_o.\ (\psi * \psi')[o(y)/x] \vdash \exists_o.\ \chi[o(y)/x] \\ \hline x{:}X \mid \exists_o.\ \psi[o(y)/x] \vdash \exists_o.\ (\psi' \mathrel{-\!\!*} \chi)[o(y)/x] \end{array} ,$$

which by (17) gives us

$$\begin{array}{c} x{:}X \mid \varphi \wedge (\psi * \psi') \vdash \varphi \wedge \chi \\ \hline x{:}X \mid \varphi \wedge \psi \vdash \varphi \wedge (\psi' \mathrel{-\!\!*} \chi) \end{array} .$$

This entails the following:

$$\begin{array}{c}
x{:}X \mid \varphi \wedge (\psi * \psi') \vdash \chi \\ \hline\hline
x{:}X \mid \varphi \wedge (\psi * \psi') \vdash \chi \wedge \varphi \\ \hline\hline
x{:}X \mid \varphi \wedge \psi \vdash \varphi \wedge (\psi' \mathrel{-\!\!*} \chi) \\ \hline\hline
x{:}X \mid \varphi \wedge \psi \vdash \psi' \mathrel{-\!\!*} \chi \\ \hline\hline
x{:}X \mid (\varphi \wedge \psi) * \psi' \vdash \chi
\end{array}$$

Letting χ be $(\varphi \wedge \psi) * \psi'$, respectively $\varphi \wedge (\psi * \psi')$, we read off the equivalence $x{:}X \mid \varphi \wedge (\psi * \psi') \dashv\vdash (\varphi \wedge \psi) * \psi'$. Now, let φ and ψ be I, and ψ' be $\top$; this gives $\mathrm{I} \wedge (\mathrm{I} * \top) \dashv\vdash (\mathrm{I} \wedge \mathrm{I}) * \top$, that is, $\mathrm{I} \dashv\vdash \top$, which in return yields $\varphi \wedge (\top * \psi') \dashv\vdash (\varphi \wedge \top) * \psi'$, namely, $\varphi \wedge \psi' \dashv\vdash \varphi * \psi'$. $\square$

ACKNOWLEDGMENTS

The authors wish to thank Carsten Butz and the anonymous referees of previous versions of the work in this article for helpful comments and insights.

REFERENCES

BANERJEE, A. AND NAUMANN, D. 2005a. Ownership confinement ensures representation independence for object-oriented programs. *J. ACM 52*, 6, 894–960.

BANERJEE, A. AND NAUMANN, D. 2005b. State based ownership, reentrance and encapsulation. In *Proceedings of the European Conference on Object-Oriented Programming*. Lecture Notes in Computer Science, vol. 3586. Springer, 387–411.

BARNETT, M., DELINE, R., FÄHNDRICH, M., LEINO, K., AND SCHULTE, W. 2003. Verification of object-oriented programs with invariants. In *Proceedings of the Conference on Formal Techniques for Java-Like Programs*.

BARNETT, M. AND NAUMANN, D. 2004. Friends need a bit more: Maintaining invariants over shared shate. In *Proceedings of the Conference on Mathematics of Program Construction (MPC)*.

BIERING, B. 2004. On the logic of bunched implications and its relation to separation logic. M.S. thesis, University of Copenhagen.

BIERING, B., BIRKEDAL, L., BUTZ, C., HYLAND, J., VAN OOSTEN, J., AND STREICHER, P. R. T. 2006. Notes on the dialectica topos. To appear.

BIRKEDAL, L., TORP-SMITH, N., AND YANG, H. 2005. Semantics of separation-logic typing and higher-order frame rules. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Press, Chicago, IL, 260–269.

BIRKEDAL, L., TORP-SMITH, N., AND REYNOLDS, J. 2004. Local reasoning about a copying garbage collector. In *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (Venice, Italy). 220–231.

BORNAT, R., CALCAGNO, C., O'HEARN, P., AND PARKINSON, M. 2005. Permission accounting in separation logic. In *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)* (Long Beach, CA). ACM, New York.

BORNAT, R., CALCAGNO, C., AND O'HEARN, P. 2004. Local reasoning, separation and aliasing. In *Proceedings of the SPACE* (Venice, Italy).

BIRKEDAL, L. AND YANG, H. 2006. Relational parametricity and separation logic. To appear.

HOARE, C. A. R. 1972. Proof of correctness of data representations. *Acta Inf. 1*, 271–281.

HOARE, C. A. R. 1971. Procedures and parameters: An axiomatic approach. In *Proceedings of the Symposium on Semantics of Algorithmic Languages*, E. Engler, ed. Springer, 102–116.

ISHTIAQ, S. AND O'HEARN, P. W. 2001. BI as an assertion language for mutable data structures. In *Proceedings of the 28th Annual ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL)* (London).

JACOBS, B. 1999. *Categorical Logic and Type Theory*. Studies in Logic and the Foundations of Mathematics, vol. 141. North-Holland, Amsterdam, The Netherlands.

KRISHNASWAMI, N., BIRKEDAL, L., ALDRICH, J., AND REYNOLDS, J. 2006. Idealized ML and its separation logic. To appear.

LAWVERE, F. 1969. Adjointness in foundations. *Dialectica 23*, 3-4, 281–296.

LEAVANS, G. 1988. Verifying object-oriented programs that use subtypes. Ph.D. thesis, MIT. Published as MIT/LCS/TR-439 in February 1989.

LEINO, K. 1995. Toward reliable modular programs. Ph.D. thesis, California Institute of Technology.

LEINO, K. R. M. AND MÜLLER, P. 2006. A verification methodology for model fields. In *Proceedings of the European Symposium on Programming (ESOP)*, P. Sestoft, ed. Lecture Notes in Computer Science, vol. 3924. Springer, 115–130.

LEINO, K. AND MÜLLER, P. 2004. Object invariants in dynamic contexts. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*.

LISKOW, B. AND GUTTAG, J. 1986. *Abstraction and Specification in Program Development*. MIT Press, Cambridge, MA.

MACLANE, S. AND MOERDIJK, I. 1994. *Sheaves in Geometry and Logic*. Universitext. Springer, New York. A first introduction to topos theory, Corrected reprint of the 1992 edition.

MITCHELL, J. C. AND PLOTKIN, G. D. 1985. Abstract types have existential type. In *Proceedings of the 12th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)* (New Orleans, LA), 37–51.

MÜLLER, P. 2002. *Modular Specification and Verification of Object-Oriented Programs*. Lecture Notes in Computer Science, vol. 2262, Springer.

NANEVSKI, A., AHMED, A., MORRISETT, G., AND BIRKEDAL, L. 2006. Abstract predicates and mutable ADTs in Hoare type theory. Tech. Rep. TR–14-06, Harvard University.

NAUMANN, D. AND BARNETT, M. 2006. Towards imperative modules: Reasoning about invariants and mutable state. *Theor. Comput. Sci. 365*, 143–168.

O'HEARN, P. W. 2004. Resources, concurrency and local reasoning. In *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR)* (London). Lecture Notes in Computer Science, vol. 3170. Springer, 49–67.

O'HEARN, P. W., YANG, H., AND REYNOLDS, J. C. 2004. Separation and information hiding. In *Proceedings of the 31st ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL)* (Venice, Italy). 268–280.

O'HEARN, P. W., YANG, H., AND REYNOLDS, J. C. 2003. Separation and information hiding (work in progress). Extended version of O'Hearn et al. [2004].

O'HEARN, P. AND PYM, D. J. 1999. The logic of bunched implications. *Bull. Symb. Logic 5*, 2 (Jun.).

PARKINSON, M. AND BIERMAN, G. 2005. Separation logic and abstraction. In *Proceedings of the 32nd Annual ACM SIGPLAN - SIGACT Symposium on Principles of Programming Languages (POPL)* (Long Beach, CA). 247–258.

PITTS, A. M. 2001. Categorical logic. In *Handbook of Logic in Computer Science, volume 5: Algebraic and Logical Structures*, S. Abramsky et al., eds. Clarendon Press, Oxford, UK. Chapter 2.

PYM, D. J. 2004. Errata and remarks for the semantics and proof theory of the logic of bunched implications. Addendum to Pym [2002]. http://www.cs.bath.ac.uk/~pym/.

PYM, D. 2002. *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logics Series, vol. 26. Kluwer.

PYM, D. J., O'HEARN, P. W., AND YANG, H. 2004. Possible worlds and resources: The semantics of BI. *Theor. Comput. Sci. 315*, 1, 257–305.

REYNOLDS, J. C. 2002. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS)* (Copenhagen, Denmark). IEEE Press 55–74.

SILBERSCHATZ, A. AND GALVIN, P. 1998. *Operating Systems Concepts*, 5th ed. World Student Series. Addison-Wesley, Reading, MA.

YANG, H. 2001. Local reasoning for stateful programs. Ph.D. thesis, University of Illinois, Urbana-Champaign.

YANG, H. AND O'HEARN, P. 2002. A semantic basis for local reasoning. In *Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)* (Grenoble, France). Springer, 402–416.

Received August 2005; revised November 2006; accepted February 2007