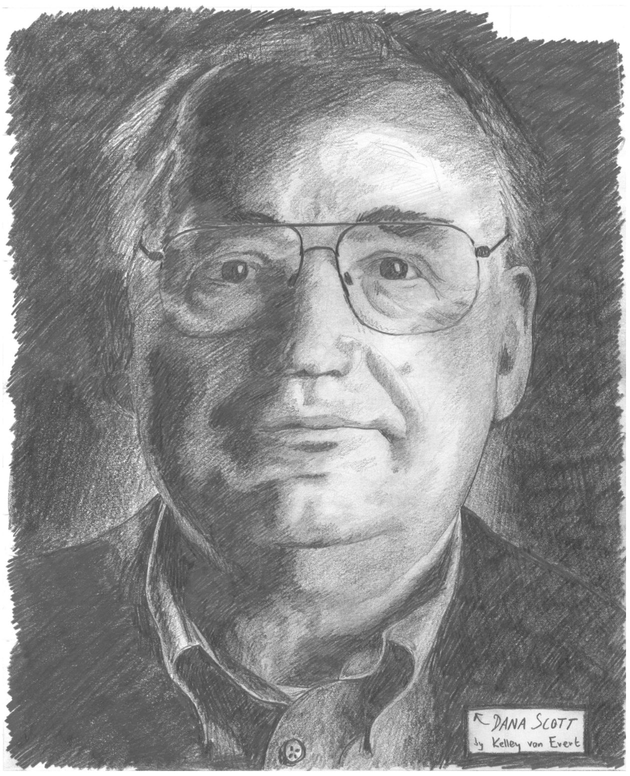
Graph models for the untyped λ-calculus



Kelley Philip van Evert

On the cover: a picture I drew of Dana Scott.

Graph models for the untyped λ-calculus

Kelley Philip van Evert

4046854

Bachelor's thesis in Computer Science

Radboud Universiteit Nijmegen

Supervisor

Herman Geuvers

July 5th, 2013

Acknowledgements

Thanks go to Erik Barendsen for introducing me to the untyped λ-calculus (in his course Semantiek & Correctheid); to Herman Geuvers for letting me explore this particular part of it, generously taking all the time in the world to discuss it with me almost every week; and finally to Niels van der Weide for thoroughly criticizing and questioning the presentation and contents of each new version of my thesis.

Abstract

This thesis discusses a few methods for, and results about, a particular class of models for the untyped λ-calculus called *graph models*. These models, which include Engeler's D_A and Scott-Plotkin's P_ω, are presented as a natural first attempt at models for the untyped λ-calculus building upon Scott's continuous semantics, in which models are derived by domain-theoretic means, as reflexive CPO's.

In the first part, we exhibit Scott's continuous semantics, motivate graph models and present a result taken from [Bucciarelli and Salibra, 2008] demonstrating that that the class of graph models admits a model with a minimal theory, which however still is more specific than the minimal theory $\lambda\beta$. This result is related to the longstanding open question as to whether the continuous semantics admits a model of $\lambda\beta$.

In the second part we focus on modifying the definition of never-extensional graph models slightly in order to obtain extensional models. Two methods were found to be very closely related: it is shown that a method due to Plotkin in a 1972 memorandum (presented in [Plotkin, 1993]), in which terms are interpreted as certain maximal elements, closed under deduction within a certain natural deduction system, is actually an instance of a general scheme that [Hoofman and Schellinx, 1991] extracted from [Bethke, 1986], which collapses graph models over an equivalence relation parametrized by any suitable pre-order satisfying a certain characterization. Finally, these constructions are related to what [Berline, 2000] calls webbed K-models, deriving their name from Krivine who in [Krivine, 1990] proved similar results as Hoofman and Schellinx.

Contents

CHAPTER 1

Introduction

This thesis discusses a few methods for, and results about, a particular class of models for the untyped λ-calculus called *graph models*.

The λ-calculus, a formal system invented in 1936 by Alonzo Church for logical purposes, is an intriguing mathematical object that relates not only to logic, but also to type theory, programming and computation in general. The basic formalism contains a definition of terms and a notion of reduction. After that, the λ-calculus can be approached in several ways. One can for instance investigate extensions of reduction inspired by, say, operational semantics of programming languages. Or one can investigate the relations between logic, type theory and the λ-calculus through the Curry-Howard correspondence.

The λ-calculus comes in two flavors: the typed and the untyped λ-calculus. In this thesis, we focus on the latter. The untyped λ-calculus is very expressive and embodies the notion of effective computability, just as, for instance, Turing-machines do. A prime examples of this expressivity is that it doesn't differentiate between functions and arguments, "all terms being equal". Thus, terms can be applied to themselves. Indeed this is a fundamental property of computing, where programs and data are one and the same thing. Another important example is that every term has a fixed-point.

In this thesis we approach the λ-calculus from the point of view of models. We try to construct models with certain properties, e.g. being extensional. This expressivity described above, allowing us on the one hand to represent all computable functions in a very elegant manner, only makes it that much harder to understand what these terms actually stand for. For instance, if terms stand for functions as well as elements, then the absence of types would seem to imply the nonsensical $X \sim X^X$. Obviously this is not the case, and one must make a selection of *representable* functions, but then there is the requirements that all of them must have fixed-points! Apparently the construction of models is not an easy task, as a model of the λ-calculus is a mathematical interpretation of its terms, and must give mathematical meaning to them.

In fact, even creating models for the λ-calculus had seemed impossible for quite some time after it had been invented. The first model was found, unintentionally, by Dana Scott in the late 1960s. His framework, nowadays referred to as the *continuous semantics*, will also be the framework in which our investigation takes place. In it, we interpret terms in a complete partial orders, and only represent continuous functions over these CPO's. This framework is heavily inspired by computer science, CPO's being systems of information and continuous functions representing programs, which are by nature best described as a process of approximation.

The counterpart of a model is the theory it produces. A model's theory can give insight into the construction, but can otherwise also raise questions about it. A longstanding open question in the literature is whether there even exists a model in Scott's framework that equates only those terms that are convertible (that is, in the basic formalism of the λ-calculus). Similar questions, almost all currently unanswered, arise for different (sub)classes of models and theories. In this thesis we review a partial negative answer to this question offered by [Bucciarelli and Salibra, 2008], stating that the subclass of graph models within this framework does not admit a model with this minimal theory.

In chapter 2 we deal the preliminaries of this thesis. We first briefly reflect upon the λ-calculus, what it is to be a model or a theory for it, and also more formally state the question raised above about the representability of certain classes of models. Then, we lay out the domain theoretic means of Scott's continuous semantics by which we obtain models of the λ-calculus from certain *reflexive CPO's*.

In chapter 3 we motivate and define the concept of graph models, and then present a result taken from [Bucciarelli and Salibra, 2008] demonstrating that that the class of graph models admits a model with a minimal theory, which however still is more specific than the minimal theory $\lambda\beta$. This result is related to the above mentioned longstanding open question as to whether the continuous semantics admits a model of $\lambda\beta$, negatively answering it for the subclass of graph models.

In chapter 4 we focus on modifying the definition of never-extensional graph models slightly in order to obtain extensional models. After discussing why graph models are never extensional, we demonstrate a method, due to [Hoofman and Schellinx, 1991], of extensionalizing graph models, given that an additional pre-order is provided satisfying a certain characterization. This method generalizes a result from [Bethke, 1986]. We also show how a construction due to Plotkin in a 1972 memorandum (presented in [Plotkin, 1993]) is actually an instance of this method. Finally, these constructions are related to what [Berline, 2000] calls webbed K-models, deriving their name from Krivine who in [Krivine, 1990] proved similar results as Hoofman and Schellinx.

The following picture illustrates some of the various structures and construction methods involved in this thesis:

$$\begin{tikzcd}
\begin{array}{c}\text{partial pair}\\ (A, \iota_A)\end{array} \arrow[d, "\begin{array}{l}\textit{canonical}\\ \textit{completion}\end{array}"] & & & \\
\begin{array}{c}\text{total pair}\\ (B, \iota_B)\end{array} \arrow[r, dashed] \arrow[d, "\begin{array}{l}\textit{std. graph model}\\ \textit{construction}\end{array}"] & \begin{array}{c}+\text{ suitable pre-order}\\ (\vdash) \subseteq B^{<\omega} \times B\end{array} \arrow[d, "\begin{array}{l}\textit{Hoofman/}\\ \textit{Schellinx}\end{array}"] & \text{e.g.} & \begin{array}{c}\text{Plotkin's } (\vdash) \text{ or}\\ \text{Bethke's } (\preccurlyeq) \subseteq B^1 \times B\end{array} \\
\begin{array}{c}\text{non-strict refl. CPO,}\\ \text{graph } \lambda\text{-model}\\ (\mathcal{P}_B, \subseteq, \bigcup, F, G)\end{array} & \begin{array}{c}\text{strict. refl. CPO,}\\ \text{non-graph ext. } \lambda\text{-model}\\ (\overline{\mathcal{P}_B}, \subseteq, \bigcup, \overline{F}, G')\end{array} & &
\end{tikzcd}$$

CHAPTER 2

Preliminaries

2.1 λ-calculus, its models and theories

Let us briefly discuss the λ-calculus, and what it is to be a model or a theory of it.

2.1.1 λ-calculus as a syntactical entity

First, we note that the λ-calculus is a syntactic mathematical entity where one has *terms*, a notion of *reduction* $(\rightarrow)$, and a notion of *conversion* $(=)$. The terms are those inductively generated from the following grammar:

$$M ::= x \mid \lambda x.M \mid MM$$

Some well-known terms are (where $(\equiv)$ stands for definitional equality, and common notational conventions apply):

$$\mathsf{I} \equiv \lambda x.x \qquad \mathsf{K} \equiv \lambda xy.x$$

$$\mathsf{1} \equiv \lambda xy.xy \qquad \mathsf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

The reduction relation $(\rightarrow)$ captures the idea of computing, defining how terms can be rewritten, or reduced, to other terms. Sometimes terms cannot be rewritten anymore, and are said to be in *normal form*. In the computing analogy, this would correspond to a *value* resulting from a computation. Reduction is the least relation satisfying (β) which is moreover closed under subterm reduction. If additionally (η) is satisfied, we call it $(\beta\eta)$-reduction.

$$(\lambda x.N)M \to N[x := M] \tag{β}$$

$$\lambda x.Mx \to M \tag{η}$$

The (η) axiom achieves a notion of *extensionality*, and is often considered not necessarily a part of pure λ-calculus. This is why we differentiate between normal and $(\beta\eta)$-reduction.

However, in this thesis we will not be concerned with reduction, but rather focus on conversion $(=)$, which is the reflexive symmetric transitive closure of reduction. From now on, when we talk of this equivalence relation we will denote it either $(=_{\beta\eta})$ or $(=_\beta)$, to differentiate which notion of reduction it extends. Ordinal mathematical equality is $(=)$ and $(\equiv)$ stands for definitional equality (in particular, of λ-terms).

The lambda calculus is very expressive, in that it embodies *effective computability* just like Turing Machines or μ-recursive functions do. It can encode numbers, and thereby all finite data structures. Well-known encodings are the Church-numerals $c_n \equiv \lambda fx.f^n x$, where $f^n x$ is the n-th fold application of f to x, and the Barendregt-numerals $\ulcorner 0 \urcorner \equiv \mathtt{I}$, $\ulcorner n+1 \urcorner \equiv \lambda x.x\mathsf{K}\ulcorner n \urcorner$. Fixing any such encoding $n \mapsto \ulcorner n \urcorner$, we have that any computable function $f : \mathbb{N}^k \to \mathbb{N}$ can be represented by some lambda term M, meaning that for all $n_1, \dots n_k \in \mathbb{N}$:

$$\ulcorner f(n_1, \dots n_k) \urcorner =_\beta M \ulcorner n_1 \urcorner \dots \ulcorner n_k \urcorner$$

Also, all terms have fixed points, which means that for every M there is an X such that $M =_\beta MX$. One can obtain a fixed point of M by taking FM for any *fixed-point combinator* F (of which there are many) such as the well-known $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$. By this property it is easily seen that any recursive function on the naturals can be encoded, because such a recursive function is nothing more than a fixed point of its (non-recursive) specification.

2.1.2 Structure, meaning and applications of models

In this thesis we will be discussing various, quite specific, classes of models of the untyped lambda calculus. A model, here, is a mathematical structure showcasing the properties of the λ-calculus. It entails *interpreting* terms to objects within the model, such that λ equality corresponds to the interpreted equality. It turns out that getting this correspondence exact is not easy, and remains open to investigation in general. In fact even constructing models had seemed impossible for quite a while after the lambda calculus had been invented by Alonzo Church in the 1930s. The first model D^∞ was unintentionally found by Dana Scott in the late 1960s.

As a model exhibits the nature of the λ-calculus, and the λ-calculus has two main constructs (namely, that of application and abstraction), one would expect a model to be a structure of the shape $(X, \cdot, \lambda)$, indeed together with some interpretation function $(\ \ )$ sending λ-terms to X. What the $(\cdot)$ should be is straightforward (a binary function on X), however there are various perspectives on what the λ should be. In this thesis, we consider Scott's *continuous semantics*, where models are derived from *reflexive CPO's* of the shape (X, F, G), and which neatly fits this pattern. It embodies the perspective that λ is an encoding operator that maps the representable functions (in the case of the continuous semantics these are the continuous functions over X) to individual elements of X.

A structure X that is a model of the lambda calculus should account for the syntactical properties observed in the lambda calculus. Maybe the most fundamental syntactical property of the lambda calculus is that functions and arguments coincide. Every term can be applied as a function, or passed as an argument, to any other term. This seems to entail that we must comprehend X's function space in X, that is, $X^X \prec X$, if X is to be a model. However, as we only have to deal with "effectively computable" functions, this is not actually the case, and obvious cardinality issues do not immediately prevent us from finding models. So we must find some subset $\mathscr{F} \subseteq X^X$, of the set of all the functions, that is to be comprehended, $\mathscr{F} \prec X$. A next important syntactical property is that of fixed-points, mentioned above: for every term M there is a term Q such that $MQ =_\beta Q$. This means, specifically, that F cannot contain functions without fixed-points, for every comprehensible function apparently has a fixed-point. This was Scott's original inclination to work in the framework of lattices (or, more generally, CPO's) and continuous functions upon them, as Plotkin remarks in [Plotkin, 1993]. This framework is often referred to as the continuous semantics, and its models continuous models. [As a side note, Plotkin also remarks that this could be seen as a variation of Russell's Paradox. Indeed, in perspective of Church's original aim of logically interpreting λ-terms, one would desire a certain term $(\neg)$ without any fixed-points.]

Models are useful to investigate the lambda calculus in a more semantic way. Often, easier and more logical proofs can be given, model-theoretically, than by syntactic considerations. Scott's original model D^∞ immediately demonstrated that the λ-calculus remains consistent after equating all insolvable terms. In fact our simplest graph model, Engeler's D_A, already demonstrates this by interpreting all unsolvable terms alike to the element

$\varnothing$. And of course, models can be interesting in themselves. Berline [2000] gives a nice overview of various results that can be proven by model-theoretic means using so-called *webbed* models (which are basically just graph models, generalized where necessary), and natural questions that arise when working with such models.

2.1.3 Theories

A *theory* of the λ-calculus is one of equality, thus a subset of $\Lambda \times \Lambda$, which is at least closed under reduction. We have already seen the theories $\lambda\beta$ with equality $(=_\beta)$ and $\lambda\beta\eta$ with equality $(=_{\beta\eta})$. Other theories can arise from other (extended) notions of reduction, or from models by observing which lambda terms are interpreted to the same element. It is this latter source of theories that we concentrate on in this thesis.

Given a model D, we denote its theory $\mathrm{Th}(D)$. Given a class of models $\mathscr{C}$, say for instance the class of graph models, the class of all continuous models (which are built from reflexive CPO's as will be explained later), or the class of extensional continuous models (from strict reflexive CPO's), various question naturally arise about the class of theories that it gives rise to (see [Berline, 2000] for a more thorough consideration):

1. How expressive is this class of models, or exactly how many different theories does this class represent?
2. How representative is its set of theories? Does it contain a model with exactly the $\lambda\beta$ theory? The $\lambda\beta\eta$ theory?
3. Does its set of theories contain a minimal theory, where by T being minimal we mean that it is contained in any other of the theories of the class (setting aside whether or not it is equal to $\lambda\beta$)?

As for (1), it turns out that the class of continuous models, and that of graph models (which is a subclass of the continuous models), both admit $2^{\aleph_0}$ pairwise distinct theories. See for instance [Kerth, 1998].

In this thesis we give a positive answer to (3) for the case of graph models. However, this theory turns out to be still more specific than $\lambda\beta$, so that the graph models do not admit any model with the $\lambda\beta$ theory, negatively answering question (2). This result is taken from [Bucciarelli and Salibra, 2008].

It has also been shown, in [Di Gianantonio et al., 1995], that a minimal extensional continuous model exists, however this proof does not carry over to non-extensional models, and it is still unknown whether the theory of this model is $\lambda\beta\eta$. In general, question (2) remains unsettled for most (variations) of the described classes.

2.2 Basic domain theory

Domain theory a study of *computability*, initiated by Dana Scott in the late 1960s when he was looking for a denotational semantics of the λ-calculus and found the first model D^∞. Complete partial orders (domains of computation) and continuous functions (computable functions) are the main protagonists of the field.

Definition 2.1 (CPO). A complete partial order (CPO), is a structure $(D, \sqsubseteq, \bot, \bigsqcup)$ with:

- a non-empty set D;
- a partial order $(\sqsubseteq) \subseteq D \times D$;
- a least element $\bot$, for which $\bot \sqsubseteq d$ for all $d \in D$;
- for every directed set $X \subseteq D$, a supremum $\bigsqcup X$ [where X is called directed iff every two elements in X have a common greater-or-equal element in X, and the supremum is greater or equal to all elements in X].

Note that the existence of $(\bot)$ is already guaranteed by being the supremum of $\varnothing$, so we will often leave it out of the definition. We denote CPO's D, E, and usually annotate their components only when necessary to disambiguate (e.g. $\sqsubseteq_D$).

Often these structures are referred to, more accurately, as *directed-complete* partial orders. However, in our case the distinction is less important because we'll only be working with directed completeness, and thus we prefer the shorter name.

A very related notion to the partial order is the pre-order, often denoted $(\preccurlyeq)$ or $(\vdash)$, which discards the antisymmetry constraint. They owe their name to "almost being" a partial order: if one takes equivalence classes over the symmetric closure of the pre-order, a partial order is obtained.

Definition 2.2 (Monotonicity & Continuity). Let $f : D \to E$ be a function.

- f is called monotone if it preserves order monotonically, that is, $f(x) \sqsubseteq f(y)$ whenever $x \sqsubseteq y$.

- f is called (Scott-)continuous if furthermore it preserves suprema of directed sets. That is, if X is directed, then $f(\bigsqcup X) = \bigsqcup f[X]$.

Note that a continuous function automatically also preserves least elements, because $\bot = \bigsqcup \varnothing$ and $\varnothing$ is vacuously directed. Also, monotonicity follows from preserving suprema of directed sets: if $x \sqsubseteq y$, then $f(y) = f(x \sqcup y) = f(x) \sqcup f(y)$ by continuity because $\{x, y\}$ is directed, and thus $f(x) \sqsubseteq f(y)$.

2.2.1 Some common CPO's

Lemma 2.3. *The set $[D \to E]$ of continuous functions from D to E also forms a CPO, where we define:*

- *$f \sqsubseteq g$ iff for all $x \in D$, $f(x) \sqsubseteq_E g(x)$;*
- *$\bot = D \times \{\bot_E\}$, the function that sends all elements $x \in D$ to $\bot_E$;*
- *$\bigsqcup \mathscr{F} = x \mapsto \sqcup_E\{f(x) \mid f \in \mathscr{F}\}$.*

Proof.

- $(\sqsubseteq)$ carries over reflexivity, transitivity and antisymmetry from $(\sqsubseteq_E)$.
- For arbitrary f, we have $\bot \sqsubseteq f$ because $\bot_E \sqsubseteq_E f(x)$. So $\bot$ is indeed a least element w.r.t. $(\sqsubseteq)$.
- First, note that $(\bigsqcup)$ is well-defined because $\{f(x) \mid f \in F\}$ is directed: if we have two of it's elements $f(x)$ and $g(x)$, then by directedness of F we have a function h with $f, g \sqsubseteq h$, and thus $f(x), g(x) \sqsubseteq_E h(x)$, and hence the set is directed. Now we can easily see that $\bigsqcup F$ is a supremum of F, because clearly if $f \in F$, for all $x \in D$, $f(x) \sqsubseteq_E (\bigsqcup F)(x)$. 證畢

Lemma 2.4. *The Cartesian product $D \times E$ of two CPO's also forms a CPO, where we define:*

- *$(d, e) \sqsubseteq (d', e')$ iff $d \sqsubseteq_D d'$ and $e \sqsubseteq_E e'$;*
- *$\bot = (\bot_D, \bot_E)$;*
- *$\bigsqcup X = (\bigsqcup_D \pi_1[X], \bigsqcup_E \pi_2[X])$, where π_i is the i-projection function.*

Proof.

- $(\sqsubseteq)$ carries over reflexivity, transitivity and antisymmetry from $(\sqsubseteq_D)$ and $(\sqsubseteq_E)$.
- Obvious.

- Note that $\pi_1[X]$ is directed, too (and similarly $\pi_2[X]$): for if $d, d' \in \pi_1[X]$, then there are e, e' such that $(d, e), (d', e') \in X$, and then by directedness of X we have some (d'', e'') such that, in particular, $d, d' \sqsubseteq_D d''$ and $d'' \in \pi_1[X]$. 證畢

2.2.2 Useful results

Let us briefly prove some useful lemma's that we will be needing later on.

Lemma 2.5. *A function $f : D_1 \times D_2 \to E$ is continuous if and only if the functions $d_1 \mapsto f(d_1, d_2)$ and $d_2 \mapsto f(d_1, d_2)$ are continuous, for any given d_2 and d_1, respectively.*

Proof. ($\Longrightarrow$) Suppose f is continuous, and let $X \subseteq D_1$ be directed. Name the first argument function g, and then compute:

$$\begin{aligned}
g(\bigsqcup_{D_1} X) &= f(\bigsqcup_{D_1} X, d_2) \\
&= f(\bigsqcup_{D_1 \times D_2} \{(d_1, d_2) \mid d_1 \in X\}) && \text{(definition of } (\bigsqcup_{D_1 \times D_2})) \\
&= \bigsqcup_E \{f(d_1, d_2) \mid d_1 \in X\} && (f \text{ continuous, directedness of the set}) \\
&= \bigsqcup_E g[X]
\end{aligned}$$

($\Longleftarrow$) Suppose the two argument functions are continuous, and let $X \subseteq D_1 \times D_2$ be directed. Compute:

$$\begin{aligned}
f(\bigsqcup_{D_1 \times D_2} X) &= f(\bigsqcup_{D_1} \pi_1[X], \bigsqcup_{D_2} \pi_2[X]) && \text{(definition of } (\bigsqcup_{D_1 \times D_2})) \\
&= \bigsqcup_{E \atop d_1 \in \pi_1[X]} \bigsqcup_{E \atop d_2 \in \pi_2[X]} \{f(d_1, d_2)\} && \text{(assumption, twice)} \\
&= \bigsqcup_{E \atop (d_1, d_2) \in X} && (X \text{ directed}) \\
&= \bigsqcup_E f[X] && 證畢
\end{aligned}$$

Lemma 2.6. *For any continuous $f : [D_1 \times D_2 \to E]$, the function $\widehat{f}$ which sends $d_1 \in D_1$ to $d_2 \mapsto f(d_1, d_2)$ is continuous $[D_1 \to [D_2 \to E]]$. And additionally, this mapping $f \mapsto \widehat{f}$ is continuous.*

Proof. First, let us prove that $\widehat{f}$ is continuous. Let $X \subseteq D_1$ be directed. Compute:

$$\begin{aligned}
\widehat{f}(\textstyle\bigsqcup_{D_1} X) &= d_2 \mapsto f(\textstyle\bigsqcup_{D_1} X, d_2) \\
&= d_2 \mapsto \bigsqcup_{d_1 \in X}{}_E\, f(d_1, d_2) && \text{(lemma 2.5)} \\
&= \bigsqcup_{d_1 \in X}{}_{[D_2 \to E]} \{d_2 \mapsto f(d_1, d_2)\} && \text{(definition } (\textstyle\bigsqcup_{D_2 \to E})) \\
&= \textstyle\bigsqcup_{D_2 \to E} \widehat{f}[X]
\end{aligned}$$

Second, that the mapping $f \mapsto \widehat{f}$ is continuous. Let $\mathscr{F}$ be a directed set of continuous functions from $D_1 \times D_2$ to E. Compute:

$$\begin{aligned}
\widehat{\textstyle\bigsqcup \mathscr{F}} &= d_1 \mapsto \Big(d_2 \mapsto (\textstyle\bigsqcup \mathscr{F})(d_1, d_2)\Big) \\
&= d_1 \mapsto \Big(d_2 \mapsto \bigsqcup_{f \in \mathscr{F}}{}_E\, f(d_1, d_2)\Big) && \text{(definition of } (\textstyle\bigsqcup_{[D_1 \times D_2 \to E]})) \\
&= d_1 \mapsto \Big(\bigsqcup_{f \in \mathscr{F}}{}_{[D_2 \to E]}\, d_2 \mapsto f(d_1, d_2)\Big) && \text{(definition of } (\textstyle\bigsqcup_{[D_2 \to E]})) \\
&= \bigsqcup_{f \in \mathscr{F}}{}_{[D_1 \to [D_2 \to E]]}\, d_1 \mapsto \Big(d_2 \mapsto f(d_1, d_2)\Big) && \text{(definition of } (\textstyle\bigsqcup_{[D_1 \to [D_2 \to E]]})) \\
&= \bigsqcup_{f \in \mathscr{F}}{}_{[D_1 \to [D_2 \to E]]}\, \widehat{f} && \text{證畢}
\end{aligned}$$

Note that this last lemma, which in [Barendregt, 1984] is appropriately named *Continuity of abstraction*, relates to Schönfinkel's early observation that unary functions are enough to simulate functions of several variables. Suppose you have a function $f(x, y, \dots)$ of more than one variable, then you can "abstract away" one of the variables (say, y) by defining a new function $\widehat{f}$ sending any y to the expression $f(x, y, \dots)$, and of course $f(x, y, \dots) = \widehat{f}(y)(x, \dots)$. This is an important principle in the λ-calculus, as one only has unary "function" terms.

Suppose now we were to relate the representable functions to continuous ones over some CPO, as is the case when building models in Scott's continuous semantics (which we will be doing throughout this thesis), then it is good to know that this technique preserves continuity of the functions. And this is exactly what the lemma states.

2.2.3 (Strict) reflexive CPO's

CPO's are often used to solve so-called *domain equations*, which find their origin in investigating particular kinds of data types, in the field of theoretical computer science. In our case, we would like to equate the (definable) function space over our domain to the domain itself, because lambda terms (which comprise the domain) denote functions from lambda terms to lambda terms (the function space). Acknowledging the fact that we'll only be working with continuous functions, our domain equation is:

$$D \cong [D \to D] \tag{2.1}$$

Definition 2.7 (Strict reflexivity). We call a CPO D strict reflexive iff the domain equation 2.1 holds. That is, if functions $F : D \to [D \to D]$ and $G : [D \to D] \to D$ exist such that:

$$F \circ G = \mathrm{id}_{[D\to D]}$$
$$G \circ F = \mathrm{id}_{D}$$

Furthermore, we require F and G to be continuous.

However, as we will see, we do not always need a full isomorphism between the two sides of the domain equation: an embedding of the continuous function space into D will then suffice. Thus we define:

Definition 2.8 (Reflexivity). We call a CPO D reflexive iff functions $F : D \to [D \to D]$ and $G : [D \to D] \to D$ exist such that:

$$F \circ G = \mathrm{id}_{[D\to D]}$$

2.3 Continuous semantics: models from reflexive CPO's

Continuous semantics refers to the class of lambda models that arise in the category of reflexive CPO's and continuous functions. It was originally Dana Scott's idea to look for models of the untyped lambda calculus in the category of complete lattices and continuous functions, after it had been discovered that all comprehensible functions must have fixed-points. This setting can however easily be slightly generalized to reflexive CPO's (instead of complete lattices).

Let $(D, \sqsubseteq)$ be a reflexive CPO, so there exist continuous functions

$$\begin{array}{l} G : \big[[D \to D] \to D\big] \\ F : \big[D \to [D \to D]\big] \end{array} \qquad \text{with} \qquad F \circ G = \mathrm{id}_{[D \to D]}$$

This is simply a convenient way of saying that the continuous function space $[D \to D]$ is embedded into D (though does not necessarily reach all of D). For $d, e \in D$, abbreviate $d \cdot e = de = F(d)(e)$.

Definition 2.9 (Interpretation of lambda terms)**.** Define the interpretation of a lambda term, under an environment $\rho : \mathrm{Var} \to D$, as follows:

$$\begin{aligned} (x)^D_\rho &:= \rho(x) \\ (MN)^D_\rho &:= (M)^D_\rho \cdot (N)^D_\rho \\ (\lambda x.N)^D_\rho &:= G\big(d \mapsto (N)^D_{\rho[x:=d]}\big) \end{aligned}$$

For this definition to be OK, we need the function $d \mapsto (N)^D_{\rho[x:=d]}$ to be continuous.

Lemma 2.10. *Given any N and ρ, the function $d \mapsto (N)^D_{\rho[x:=d]}$ is continuous.*

Proof. We prove this by induction on the structure of N.

- If N is a variable y it is either the constant function for $\rho(y)$ or the identity function on D, and both are trivially continuous.
- If $N \equiv PQ$ we have monotonicity and continuity by the induction hypothesis on P and Q separately, plus the fact that F is continuous (simply shifting $(\sqsubseteq)$ and $(\bigsqcup)$ through the definitions).
- The case $N \equiv \lambda y.P$ is the interesting case. By induction we know that the expression $(P)^D_{\rho[x:=d][y:=e]}$ is continuous both as a function of d, as of e (separately), and thus by lemma 2.5 the map h that sends $(d, e) \mapsto (P)^D_{\rho[x:=d][y:=e]}$ is continuous, too.

Then we compute:

$$\begin{aligned} d \mapsto (\lambda y.P)^D_{\rho[x:=d]} &= d \mapsto G\big(e \mapsto (P)^D_{\rho[x:=d][y:=e]}\big) \\ &= d \mapsto G\big(e \mapsto h(d,e)\big) \\ &= d \mapsto G\big(\widehat{h}(d)\big) \\ &= G \circ \widehat{h} \end{aligned}$$

Noting that $\widehat{h}$ is continuous by lemma 2.6, we conclude $G \circ \widehat{h}$ is also continuous by G, too, being continuous. 證畢

Definition 2.11 (Equality). Define $D \vDash M = N$ iff for all ρ, $(M)^D_\rho = (N)^D_\rho$. For closed sentences M we simply write $(M)^D$, as one can easily show this interpretation does not depend on ρ, and thus for closed sentences M and N we have $D \vDash M = N$ iff $(M)^D = (N)^D$. We often omit the "$D \vDash$" part if the model D is clear from the context, writing simply $M = N$; but it is important to remember that this equality on λ-terms is different than either $(\equiv)$ or $(=_{\beta(\eta)})$.

For D together with this interpretation to be a lambda model, we must show that β-equality is preserved. That is, for terms M, N, if $M =_\beta N$, then $M = N$. This is the case, but first let us prove a substitution lemma:

Lemma 2.12 (Substitution Lemma). $(M[x := N])^D_\rho = (M)^D_{\rho[x:=(N)^D_\rho]}$.

Proof. By induction of the structure of M. First, let $M \equiv y$. If $x \equiv y$, then $M[x := N] \equiv N$, and both interpretations are equal to $(N)^D_\rho$. If $x \not\equiv y$, then both interpretations are equal to $\rho(y)$. Now let $M \equiv PQ$. Then

$$\begin{aligned} (PQ[x := N])^D_\rho &= (P[x := N]Q[x := N])^D_\rho \\ &= (P[x := N])^D_\rho \cdot (Q[x := N])^D_\rho \\ &= (P)^D_{\rho[x:=(N)^D_\rho]} \cdot (Q)^D_{\rho[x:=(N)^D_\rho]} && \text{(induction hypothesis)} \\ &= (PQ)^D_{\rho[x:=(N)^D_\rho]} \end{aligned}$$

Finally, let $M \equiv \lambda y.P$, in which case

$$\begin{aligned}
\big((\lambda y.P)[x := N]\big)^D_\rho &= \big(\lambda y.(P[x := N])\big)^D_\rho \\
&= G\big(d \mapsto (P[x := N])^D_\rho\big) \\
&= G\big(d \mapsto (P)^D_{\rho[x:=(N)^D_\rho]}\big) && \text{(induction hypothesis)} \\
&= (\lambda y.P)^D_{\rho[x:=(N)^D_\rho]} && 證畢
\end{aligned}$$

Lemma 2.13 (Defined equality preserves $\rightarrow_\beta$). $(\lambda x.M)N = M[x := N]$.

Proof. Let ρ be any environment, then we compute

$$\begin{aligned}
\big((\lambda x.M)N\big)^D_\rho &= (\lambda x.M)^D_\rho \cdot (N)^D_\rho \\
&= G\big(d \mapsto (M)^D_{\rho[x:=d]}\big) \cdot (N)^D_\rho \\
&= \Big((F \circ G)\big(d \mapsto (M)^D_{\rho[x:=d]}\big)\Big)(N)^D_\rho \\
&= \big(d \mapsto (M)^D_{\rho[x:=d]}\big)(N)^D_\rho && \text{(reflexivity)} \\
&= (M)^D_{\rho[x:=(N)^D_\rho]} \\
&= (M[x := N])^D_\rho && \text{(substitution lemma)}
\end{aligned}$$

Reduction $(\rightarrow_\beta)$ is defined as the least relation that, after satisfying $(\lambda x.M)N \rightarrow_\beta M[x := N]$, is furthermore closed under the following rules:

$$\begin{array}{lll}
M \rightarrow_\beta N & \text{implies} & \lambda x.M \rightarrow_\beta \lambda x.N \\
M \rightarrow_\beta N & \text{implies} & PM \rightarrow_\beta PN \\
M \rightarrow_\beta N & \text{implies} & PM \rightarrow_\beta PN
\end{array}$$

But obviously these rules are also true of $(=)$, and so we are done. 證畢

Now we can easily see that β-equality is preserved under our defined equality, since it is simply the reflexive symmetric transitive closure of β-reduction.

2.3.1 A note on extensionality

In general terms, mathematical structures are called *extensional* when in it, objects that "behave the same actually are the same". In this view, it might be considered a bit of an obvious, or obviously desired, property. It makes more sense to talk of optional extensionality, when one takes the perspective that objects are but names, or descriptions, of certain behaviors or properties. A setting in which mathematical objects are just identified with their name, and behavior is secondary, is then called *intensional*.

In set theory, extensionality is expressed by means of identifying sets with "the totality of their elements" so that sets are equal only if they have the same elements, and is taken as an axiom:

$$\forall x, y, \quad x = y \quad \text{iff} \quad (\forall z,\ z \in x \Leftrightarrow z \in y)$$

This is natural, seen as sets are meant to represent exactly what they contain. In the λ-calculus, it is natural to identify terms with their functional behavior, so that we might say that two terms "behave the same" if they represent the same function. Therefore, the axiom of extensionality states:

$$\forall M, N, \quad M = N \quad \text{iff} \quad (\forall P,\ MP = NP) \tag{Ext}$$

Lemma 2.14. *Other, equivalent, formulations of extensionality for lambda calculus are*

$$\forall x, M, \quad \lambda x.Mx = M \tag{η}$$

$$\mathtt{I} = \mathbf{1} \tag{$\mathtt{I} = \mathbf{1}$}$$

Proof.

- $(\mathtt{I} = \mathbf{1}) \implies (\eta)$. By $\lambda x.Mx = \mathbf{1}M = \mathtt{I}M = M$.
- $(\eta) \iff$ (Ext). First, let M, N be terms and suppose that always $MP = NP$. In particular, for variables x we have $Mx = Nx$. Thus $M = \lambda x.Mx = \lambda x.Nx = N$. The other way around is easier: because always $(\lambda x.Mx)P = MP$, $\lambda x.Mx = M$.
- $(\eta) \implies (\mathtt{I} = \mathbf{1})$. Applying (η) we get $\lambda y.My = M$ for arbitrary M. Then applying (Ext) we get $\lambda xy.xy = \lambda x.x$. 證畢

However, this axiom is very "optional" in that it is not always considered as an axiom. $\lambda\beta$ and $\lambda\beta\eta$ are simply two different theories. When considering models of the lambda calculus, extensionality is of course not an axiom, but rather a property.

If a reflexive CPO is strict, the resuling model is extensional. Note that we have extensionality iff for all terms M, the equality $\lambda x.Mx = M$ holds. Considering a term M and environment ρ, we see that this equality indeed holds in the case that $G \circ F = \mathrm{id}_D$:

$$(\lambda x.Mx)^D_\rho = G\Big(d \mapsto F\big((M)^D_\rho\big)\big(d\big)\Big) = G\Big(F\big((M)^D_\rho\big)\Big) = (G \circ F)\big((M)^D_\rho\big)$$

CHAPTER 3

Graph models and the minimal graph theory

In this chapter we present *graph models* as a straightforward way of defining simple reflexive CPO's which can be used as λ-models. These structures, originally due to [Engeler, 1981], were amongst the first models for the untyped λ-calculus, invented after Scott's original D^∞ and Scott/Plotkin's $\mathcal{P}_\omega$. A slight variation of graph models were also found by Plotkin in 1972 (see [Plotkin, 1993]) and in fact we will be presenting an extensionalization construction from this paper in the last chapter, demonstrating that the difference, at least as far as it concerns technical construction, is non-essential. We also present a general construction method for these models called *canonical completion*, which can be used to more easily construct graph models with certain desired properties.

Using canonical completion, and a second construction method called the *weak product*, we then demonstrate that the class of graph models admit a certain minimal graph model, who's theory is included in the theory of any other graph model. This theory, however, is still more specific than $\lambda\beta$, which means that to graph model has $\lambda\beta$ as it's theory. This result, together with the weak product, is due to [Bucciarelli and Salibra, 2008].

3.1 Comprehension and application

As we have seen in the previous chapter, a lambda model can be built upon any reflexive CPO. So to construct a lambda model, we turn to finding such a suitable CPO $(D, \sqsubseteq, \bigsqcup)$ which comprehends its own continuous function space.

And in fact, we don't have to look far at all to find such a CPO. The powerset CPO $(\mathcal{P}_B, \subseteq, \bigcup)$ will suffice, given that the set B has a certain property. We want to be able to comprehend any continuous function f. Now the continuity of f allows us to represent it as the set of all finite fragments of its graph, which we call its *trace*:

$$\mathrm{Tr}(f) = \{\alpha \mapsto b \mid b \in f(\alpha),\ \text{and } \alpha \text{ is a finite subset of } B\}$$

Continuity of f gives us that if for any b, x we have $b \in f(x)$, then there must be some finite $\alpha \subseteq x$ such that $b \in f(\alpha)$. So the trace contains at least all information that f contains. Monotonicity of f ensures us that if $\alpha \mapsto b \in \mathrm{Tr}(f)$, then for all $x \supseteq \alpha$ we have $b \in f(x)$ so that the trace does not contain more information than f does. This shows that a trace $\mathrm{Tr}(f)$ completely, and thus uniquely, represents its function f. And indeed continuity of f is necessary, as well as sufficient, as easy counterexamples can demonstrate.

Additionally, it is easy to see that this encoding $f \mapsto \mathrm{Tr}(f)$ is continuous. If a function f has larger or equal results, for all inputs, than g, then indeed it will have a larger or equal trace, so we have monotonicity. And if we have a directed set of functions, then its directedness ensures us that its functions are compatible, and thus the traces simply "add up".

Now that we can (continuously) represent the continuous function space $[\mathcal{P}_B \to \mathcal{P}_B]$ within $\mathcal{P}_{B^{<\omega} \times B}$, it is clear how obtaining reflexivity is easier than before: we simply require there to be an embedding $\iota : B^{<\omega} \times B \prec B$, and interpret $\alpha \mapsto b$ as $\iota(\alpha, b) \in B$. Thus we go ahead and define G and F as:

$$G(f) = \mathrm{Tr}(f) = \{\alpha \to b \mid b \in f(\alpha)\}$$
$$F(x) = y \mapsto \{b \mid \exists \alpha \subseteq y,\ \alpha \to b \in x\}$$

Note that we write $\alpha \to b$ instead of $\alpha \mapsto b$ or $\iota(\alpha, b)$, following the literature (e.g. [Berline, 2000]). As seen, the variables a, b, c denote elements of B, and α, β, γ denote finite subsets of B. Associating $(\to)$ to the right, we also write, e.g., $\alpha \to \beta \to c$ for $\alpha \to (\beta \to c)$. Lastly, be abbreviate $\alpha \rightsquigarrow b$ if $\alpha \to b$ or $\alpha \to \alpha_0 \to \cdots \to \alpha_\ell \to b$ for certain $\alpha_0, \ldots \alpha_\ell$.

Let us briefly prove everything we just handwaivingly stated.

Lemma 3.1. *The mappings F and G are continuous:*

$$\begin{array}{l} F : \big[\mathcal{P}_B \to [\mathcal{P}_B \to \mathcal{P}_B]\big] \\ G : \big[[\mathcal{P}_B \to \mathcal{P}_B] \to \mathcal{P}_B\big] \end{array} \quad \textit{with} \quad F \circ G = id_{[\mathcal{P}_B \to \mathcal{P}_B]}$$

Proof. First let us prove that $F \circ G = \mathrm{id}_{[\mathcal{P}_B \to \mathcal{P}_B]}$. Consider any $y \in \mathcal{P}_B$, so that we must prove $F(G(f))(y) = f(y)$.

($\subseteq$) Let $b \in F(G(f))(y)$, say by $\alpha \subseteq y$ such that $\alpha \to b \in G(f)$, which is because $b \in f(\alpha)$. By monotonicity of f we then have $b \in f(\alpha) \subseteq f(y)$.

($\supseteq$) Let $b \in f(y)$. By continuity of f we have some $\alpha \subseteq y$ such that $b \in f(\alpha)$, and thus $\alpha \to b \in G(f)$. Then by definition of F we have $b \in F(G(f))(y)$.

Continuity of F. Let T be a directed set. Compute:

$$\begin{aligned} F(\bigcup T)(y) &= \{b \mid \exists \alpha \subseteq y,\ \alpha \to b \in \bigcup T\} \\ &= \bigcup_{t \in T} \{b \mid \exists \alpha \subseteq y,\ \alpha \to b \in t\} \\ &= \bigcup F[T](y) \end{aligned}$$

Continuity of $G = \mathrm{Tr}$. Let $\mathscr{F}$ be a directed set of continuous functions. Compute:

$$\begin{aligned} \mathrm{Tr}\big(\bigsqcup \mathscr{F}\big) &= \mathrm{Tr}\big(x \mapsto \bigcup_{f \in \mathscr{F}} f(x)\big) \\ &= \{\alpha \to b \mid b \in \bigcup_{f \in \mathscr{F}} f(\alpha)\} \\ &= \bigcup_{f \in \mathscr{F}} \{\alpha \to b \mid b \in f(\alpha)\} \\ &= \bigcup \mathrm{Tr}[\mathscr{F}] \end{aligned}$$

證畢

And thus we can proceed to define a graph model as precisely such a structure:

Definition 3.2 (Graph model). A graph model, or *total pair*, is a structure (B, ι) where B is a non-empty set, and ι is an injection from $B^{<\omega} \times B$ into B. The domain B is sometimes called the *web* of the model, and ι the *web injection*.

As we have seen, graph models exploit the characteristic possibility of a continuous function on a powerset domain to be represented by finite fragments of its graph. Following [Bethke, 1986], we call these fragments *instructions*, a metaphor which in the last chapter will turn out to easily motivate an extensionalization method. The requirement that $B^{<\omega} \times B \prec B$

which allows us to then represent continuous functions as elements $x \in \mathcal{P}_B$ is more of a technical consideration than one of meaning.

We made a quite particular choice regarding the shape of "finite fragments". The somewhat artificial construction would have worked equally well if we chose to put

$$\mathrm{Tr}(f) = \{\alpha \to \beta \mid \beta \subseteq f(\alpha)\}$$

...in which case we would have defined the slightly different

$$F(x) = y \mapsto \bigcup\{\beta \mid \exists\alpha \subseteq y,\ \alpha \to \beta \in x\}$$

And in fact, this is exactly what Plotkin [1993] does. We will refer to this class of models as Plotkin graph models. Somewhat surprisingly, although the formal construction is quite similar, the resulting theories are not necessarily. In fact, Plotkin [1993] shows that the resulting models are isomorphic if and only if either the atom sets that one starts with have the same cardinality, or if the models are trivial.

Definition 3.3 (Plotkin graph model). A Plotkin graph model is a structure (B, ι) where B is a non-empty, and ι is an injection from $B^{<\omega} \times B^{<\omega}$ into B.

Disregarding the fact that the resulting theories and models are often different, in the following chapter we'll show that the constructions that Plotkin offered to extensionalize his graph models work equally well for ordinary graph models. Furthermore, we'll show that (again, only in the setting of ordinary graph models, although using "ported" constructions from the Plotkin graph model setting) these extensionalizations are actually quite similar.

3.2 Canonical completion

A useful and obvious construction method for graph models is to extend a partially specified graph model (A, ι) simply by inductively adding missing parts of the web injection until a complete graph model, $\overline{(A, \iota)}$, is obtained. In fact, this is what a graph model was conceived to be by Engeler in [Engeler, 1981], where the ι was not yet present (but simply the identity). The addition of a possibly already given web injection ι is just a slight generalization, which allows for construction of more interesting models. As said, we apply the completion construction to *partial graph models* which to this end are defined as follows:

Definition 3.4 (Partial graph model). A *partial graph model*, or *partial pair*, is a structure (A, ι) where A is a non-empty set, and ι is a partial injection from $A^{<\omega} \times A$ into A.

We define the set B, to be the graph model web, inductively as follows:

$$\begin{aligned} B_0 &= A \\ B_{n+1} &= B_n \cup \left({B_n}^{<\omega} \times B_n - \mathrm{dom}(\iota)\right) \\ B &= \bigcup_{n \in \mathbb{N}} B_n \end{aligned}$$

We define the completed web injection ι_B by:

$$\iota_B(\alpha, b) = \begin{cases} \iota(\alpha, b) & \text{if } \iota(\alpha, b) \downarrow \\ (\alpha, b) & \text{else} \end{cases}$$

Definition 3.5 (Canonical completion). The graph model (B, ι_B) is called the *canonical completion* of (A, ι), and is denoted as $\overline{(A, \iota)}$.

Now that we have two web injections ι and ι_B, we need to be a little careful writing down $\beta \to c$, as it can lead to ambiguity. When necessary, we explicitly annotate $(\to)$ with the (partial) graph model it belongs to, as in $\beta \to_B c$ for $\iota_B(\beta, c)$.

Definition 3.6 (Rank). For any element b of B, we define the *rank of b*, denoted $\mathrm{rank}(b)$, to be the least $n \in \mathbb{N}$ such that $b \in B_n$.

We can get into trouble if certain pairs are already present in A, but this is a non-interesting constructional detail, so we'll assume that this is not the case.

The canonical construction unifies the graph model concept, as we see that for instance Engeler's model D_A is the completion of the partial pair $A = (A, \varnothing)$ and the model $\mathcal{P}_\omega$, due to Plotkin and Scott (independently) is the completion of $(\{0\}, \{(\varnothing, 0) \mapsto 0\})$.

3.3 Minimal graph theory by means of a weak product

The weak product of a family of graph models is the main construction method introduced in [Bucciarelli and Salibra, 2008] to show that the minimal graph theory exists. It takes a family of graph models, of at most countable size, and generates a new graph model that in a way "combines" exactly the structure of all given models, thereby making its theory more specific than that of any of its factor models. The idea is quite elegant, and the proof revolves around an induction with a smart closure property on environments.

Definition 3.7 (Weak product). Let $\mathscr{B} = \{(B_i, \iota_i) \mid i \in I\}$ be a family of graph models of at most countable size. Assume for simplicity that the webs of these models are pairwise disjoint. Then the weak product of $\mathscr{B}$, written $\Diamond\mathscr{B}$, is simply defined as the canonical completion of the (disjoint) union of the individual graph models:

$$\Diamond\mathscr{B} = \overline{\Big(\bigcup_{i \in I} B_i, \bigcup_{i \in I} \iota_i \Big)}$$

Now let us consider more precisely what we meant by saying that a weak product "combines" the structure of all its factors. It can be easily seen that no information is "thrown away" when constructing the weak product, so let us try to reconstruct this information.

From now on, we let E be a weak product of graph models, and let B_i be one of its factors. To disambiguate $(\rightarrow)$, we sometimes denote it either $(\rightarrow_E)$ or $(\rightarrow_{B_i})$, to be precise on which graph model it lives in. Also, note that if $\beta \rightarrow_E c \in y \cap B_i$ we necessarily have $\beta \rightarrow_E c = \beta \rightarrow_{B_i} c$, in which case we intentionally leave away the annotation, writing simply $\beta \rightarrow c \in y \cap B_i$.

Definition 3.8 (Anchoredness). We say that b is anchored in B_i iff either $b \in B_i$, or $b = \beta \rightsquigarrow c$ for certain β and $c \in B_i$.

Definition 3.9 (Original part & Flattening). We define the original part $\llcorner b \lrcorner \in B_i$ of any b that is anchored in B_i, and the flattening function $\downarrow : E \rightarrow E$ that takes B_i-anchored elements to their original part but leaves the rest be, mutually by recursion on the rank of b:

$$\llcorner b \lrcorner = \begin{cases} b & \text{if } b \text{ has rank } 0 \\ (\downarrow[\beta] \cap B_i) \rightarrow_{B_i} \llcorner c \lrcorner & \text{else, and let } b = \beta \rightarrow_E c \end{cases} \qquad \text{(Original part)}$$

$$\downarrow b = \begin{cases} \llcorner b \lrcorner & \text{if } b \text{ is anchored in } B_i \\ b & \text{else} \end{cases} \qquad \text{(Flattening)}$$

Now that we have some tools to extract certain B_i-specific parts of elements $x \in \mathcal{P}_E$, let us turn to the main technical objective of the proof.

Lemma 3.10. *The B_i content of an interpretation in E is exactly the term's B_i-interpretation. That is, for any closed λ-term M and B_i-environment τ,*

$$(M)^{B_i}_{\tau} = (M)^{E}_{\tau} \cap B_i$$

Note that the $(\subseteq)$ direction of this equation is easy, simply because B_i is smaller than E and the way the contents of interpretations are propagated through the definitions of F and G. So what we are actually proving is that the B_i content of the E-interpretation is no more than would have interpreted in B_i in the first place. What we try to gain by this result is showing that E-equality of terms implies B_i-equality. The following picture illustrates how we obviously need equality instead of only inclusion to do this.

$$\begin{array}{ccc} (M)^E \cap B_i & = & (N)^E \cap B_i \\ \cup\!\!\mid & & \cup\!\!\mid \\ (M)^{B_i} & =^{??} & (N)^{B_i} \end{array}$$

This lemma is a statement about the interpretation of terms, which is naturally proven by an induction on the structure of terms. So we need to account for the case of open terms, too, which often entails extracting the gist of the statement and, along the way of the induction, assuming it holds for the environment (which is OK, as it will hold for all terms in the end). Bucciarelli and Salibra cleverly defined this closure property as follows:

Definition 3.11 (Closure property). We say that an element $x \in \mathcal{P}_E$ is closed iff $\downarrow[x] \subseteq x$. Moreover we say that an environment ρ is closed iff for all variables y, the element $\rho(y)$ is closed. Note that if x is closed and $b \in x$ is anchored in B_i, then $\llcorner b \lrcorner \in x$, too. Regarding environments, we abuse notation writing simply $\rho \cap B_i$ for $y \mapsto \rho(y) \cap B_i$.

The motivation for this definition is that all B_i content of an E-interpretation, given the definitions of interpretation in terms of F and G, must already be present in the E-interpretations of its term's subterms. However, this information may be "hidden within" other, mixed-up, E content, later to be extracted by means of application. Saying that an interpretation is closed is saying that it has no additional hidden content.

Consider the following example. Let $\beta \cup \{b\} \subseteq B_i$ and $\gamma \cap B_i = \varnothing$. Suppose we find terms M and N whose E-interpretations are OK (contain only B_i content that is already contained in their B_i-interpretation) such that additionally $\beta \cup \gamma \to b \in (M)^E$ and $\beta \cup \gamma \subseteq (N)^E$, though there is no $\beta' \subseteq B_i$ such that $\beta' \subseteq (N)^E$ and $\beta' \to b \in (M)^E$. Then the E-interpretation $(MN)^E$ of their application is not OK, as it contains the instruction b which will not be present in $(MN)^{B_i}$.

Thus we prove lemma 3.10 by an induction on the structure of M, while along the way assuming that ρ is closed. First, we prove the stronger statement that every intermediary interpretation is closed, as well as that the important direction of the equality of lemma 3.10 holds.

Lemma 3.12. *For any lambda term M and closed E-environment ρ:*

1. $(M)^E_\rho$ *is closed*
2. $(M)^E_\rho \cap B_i \subseteq (M)^E_{\rho \cap B_i}$

Proof. Simultaneously, by induction on the structure of M.

- If M is a variable, then both statements are trivially true.

- If $M \equiv NP$, then (1) holds because application, in general, preserves closedness. To see this, let $x, y \in \mathcal{P}_E$ and suppose that $b \in xy$, say by $\alpha \subseteq y$ and $\alpha \to b \in x$. Assume that b is anchored in B_i, for else we are done. Because x is closed, we also have $\alpha \cap B_i \to \llcorner b \lrcorner \in x$, and obviously $\alpha \cap B_i$ is also a subset of y, as it's a subset of $\alpha \subseteq y$. Thus we conclude that $\llcorner b \lrcorner \in xy$, and we are done.

  Also (2) follows easily: suppose that $c \in (NP)^E_\rho \cap B_i$, say by $\beta \to c \in (N)^E_\rho \cap B_i$ and $\beta \subseteq (P)^E_\rho \cap B_i$. Then because $(N)^E_\rho$ and $(P)^E_\rho$ are closed, also $\llcorner \beta \to c \lrcorner = {\downarrow}[\beta] \cap B_i \to c \in (N)^E_\rho \cap B_i$ and ${\downarrow}[\beta] \cap B_i \subseteq {\downarrow}[\beta] \subseteq (P)^E_\rho$. And thus $c \in (NP)^E_\rho$.

- So let us consider the case that $M \equiv \lambda x.N$. Let use first prove (1). Suppose that $b \in (\lambda x.N)^E_\rho$. We must prove that ${\downarrow} b \in (\lambda x.N)^E_\rho$. Note that if b is not anchored in B_i, then ${\downarrow} b = b \in (\lambda x.N)^E_\rho$ and we are immediately done, so let $b = \beta \to_E c$, anchored in B_i.

  By definition of interpretation, $c \in (N)^E_{\rho[x:=\beta]}$. Because interpretation is monotone w.r.t environments, $c \in (N)^E_{\rho[x:=\overline{\beta}]}$. Then, because $\rho[x := \overline{\beta}]$ is closed and c is anchored in B_i, part (1) of the lemma gives us that $\llcorner c \lrcorner \in (N)^E_{\rho[x:=\overline{\beta}]}$. Now that $\llcorner c \lrcorner \in B_i$ making it B_i specific content of the interpretation, by (2) we know that

$$\llcorner c \lrcorner \in (N)^E_{\rho[x:=\overline{\beta}] \cap B_i} = (N)^E_{(\rho \cap B_i)[x:=\overline{\beta} \cap B_i]} = (N)^E_{(\rho \cap B_i)[x:={\downarrow}[\beta] \cap B_i]}$$

Thus by definition of interpretation,

$$\begin{aligned}\llcorner\beta \rightarrow_E c\lrcorner &= \downarrow[\beta] \cap B_i \rightarrow_{B_i} \llcorner c\lrcorner \\ &= \downarrow[\beta] \cap B_i \rightarrow_E \llcorner c\lrcorner \in (\lambda x.N)^E_{\rho\cap B_i}\end{aligned}$$

And finally, because monotonicity gives us that $(\lambda x.N)^E_{\rho\cap B_i} \subseteq (\lambda x.N)^E_\rho$ we have

$$\downarrow(\beta \rightarrow_E c) = \llcorner\beta \rightarrow_E c\lrcorner \in (\lambda x.N)^E_\rho$$

...completing the proof of (1). Now let us prove (2), which is a bit easier. Suppose we find a $b \in (\lambda x.N)^E_\rho$ with additionally $b \in B_i$. We must prove $b \in (\lambda x.N)^E_{\rho\cap B_i}$ which is to say that it is already in the interpretation of $\lambda x.N$ within B_i. By definition of interpretation we already have $\beta \subseteq B_i$ and $c \in B_i$ such that $b = \beta \rightarrow_{B_i} c$ and $c \in (N)^E_{\rho[x:=\beta]} \cap B_i$. Then by (2)

$$\begin{aligned}c &\in (N)^E_{\rho[x:=\beta]\cap B_i} \\ &= (N)^E_{(\rho\cap B_i)[x:=\beta]} && \text{(because } \beta \subseteq B_i\text{)}\end{aligned}$$

And thus by definition of interpretation we have $b \in (\lambda x.N)^E_{\rho\cap B_i}$, concluding the proof of (2). 證畢

Proof of lemma 3.10. Let τ be a B_i-environment, and also let us concentrate on the important inclusion $(M)^E_\tau \cap B_i \subseteq (M)^{B_i}_\tau$ (we already noted that the other direction is easy). We prove this by induction on the structure of M.

- If M is a variable, then it holds trivially.
- Let $M \equiv NP$, and suppose $b \in (NP)^E_\tau \cap B_i = (N)^E_\tau(P)^E_\tau \cap B_i$, say by $\beta \subseteq (P)^E_\tau$ and $\beta \rightarrow_E b \in (N)^E_\tau$. By lemma 3.12 (1), we know that both $(N)^E_\tau$ and $(P)^E_\tau$ are already closed (as τ is indeed a specific instance of a closed E-environment), and thus $\downarrow[\beta]\cap B_i \subseteq \downarrow[\beta] \subseteq (P)^E_\tau$ and $\downarrow(\beta \rightarrow_E b) = \llcorner\beta \rightarrow_E b\lrcorner = \downarrow[\beta]\cap B_i \rightarrow_{B_i} b \in (N)^E_\tau$. By induction, $(N)^E_\tau = (N)^{B_i}_\tau$ and $(P)^E_\tau = (P)^{B_i}_\tau$, and thus $b \in (NP)^{B_i}_\tau$.
- Easier. Let $M \equiv \lambda x.N$, and suppose $b \in (\lambda x.N)^E_\tau \cap B_i$, which means that $b = \beta \rightarrow_{B_i} c$ for certain $\beta \subseteq B_i$ and $c \in B_i$ such that $c \in (N)^E_{\tau[x:=\beta]}$. By induction $c \in (N)^{B_i}_{\tau[x:=\beta]}$, so that $b \in (\lambda x.N)^{B_i}_\tau$. 證畢

Lemma 3.13 (Theory of a weak product). *The theory of a weak product is included in the theory of each of its factors. That is,* $\mathrm{Th}(E) \subseteq \mathrm{Th}(B_i)$.

Proof. Let M, N be terms such that $E \vDash M = N$, and τ a B_i-environment. We simply compute:

$$(M)^{B_i}_\tau = (M)^E_\tau \cap B_i = (N)^E_\tau \cap B_i = (N)^{B_i}_\tau$$ 證畢

Now we show how the construction techniques previously introduced allow us to prove the existence of the minimal graph theory.

Recall that the minimal graph theory T_{min}, if existent, is the theory:

$$T_{\text{min}} = \{M = N \mid \forall \text{ graph models } B,\ B \vDash M = N\}$$

For all M, N, if in any model B, the equation $M = N$ fails to hold, then it is not in T_{min}. But we can simply enumerate all equations $M = N$, and all of these either do or do not fail in some graph model. With countable choice we can pick, for any equation $M = N$ which should not be in T_{min}, some model $B_{M \neq N}$. We then construct the weak product of these counter-exemplary models:

$$E = \Diamond\{B_{M \neq N} \mid M = N \text{ fails to hold in some graph model}\}$$

Theorem 3.14 (Minimal graph theory). *The minimal graph theory exists, and it is* $\text{Th}(E)$.

Proof. By theorem 3.13 we know that $\text{Th}(E) \subseteq \text{Th}(B_{M \neq N})$ for all M, N for which $M = N$ fails to hold in some graph model. But then we cannot have that $E \vDash M = N$, as $B_{M \neq N} \vDash M \neq N$. Thus the theory of E excludes all equations that fail in some graph model, and thus it is the minimal theory T_{min}. 證畢

This theory, however, is still more specific than $\lambda\beta$. We do not prove this, but rather give an indication why it is not the case (lemma 41 of [Bucciarelli and Salibra, 2008]).

In [Selinger, 2003] it is proven that, in a model of $\lambda\beta$, non-convertible terms must necessarily be incomparable. That is, if E is to be a model of $\lambda\beta$, then for every $M \neq_\beta N$ we have $(M)^E \cap (N)^E = \varnothing$. However, in any graph model B we have $(\Omega_3)^B \subseteq (\lambda x.\Omega_3 x)^B$ (lemma 40 of [Bucciarelli and Salibra, 2008]), where $\Omega_3 \equiv (\lambda x.xxx)(\lambda x.xxx)$.

CHAPTER 4

Constructing extensional graph-like models

This chapter investigates the possibility of modifying the graph model definition slightly in order to obtain extensional models. As is the definition of a graph structure, we pursue a simple solution for this. Extensionality is a requirement on the identification of certain objects, so collapsing a given graph model over some suitable equivalence relation seems to be the most appropriate solution. In fact, although it doesn't work directly, this idea idea will turn out to be nearly sufficient.

After discussing exactly why graph structures cannot be extensional and a few basic details on extensionality, we first prove (a slight variation of) some results due to [Hoofman and Schellinx, 1991] which show that any graph model can be collapsed over an equivalence relation into an extensional model, given a pre-order over it's web that satisfies a certain characterization. This characterization was inspired by a particular pre-order ($\preccurlyeq$) that Bethke introduced in [Bethke, 1986] to modify D_A into an extensional model.

Then we review how a seemingly different approach due to Plotkin in 1970 (presented in [Plotkin, 1993]) actually already implemented this method, in an elegant and quite general way, by defining a pre-order ($\vdash$) by means of a natural deduction system with inference rules corresponding nearly 1-to-1 to the elements of the characterization given by Hoofman and Schellinx.

Finally, we very briefly relate these constructions to what Berline calls K-models in [Berline, 2000]. These K-models are one of many possible variations of the standard definition of a graph model which Berline attempts to unify in a framework of what she calls *webbed* *models*, and derive from results from [Krivine, 1990] very similar to those of Hoofman and Schellinx.

4.1 Graph models and extensionality

Graph models and the graph algebras underlying them are never extensional. This is an inherent property of graph models (algebras) in the way we constructed them. The elements of a graph algebra are sets of finite fragments, or instructions, of the functions they represent, but some fragments are simply "bigger" than others without saying anything "more" and are redundant when included in an element which already contains the others. Thus we can easily find elements with the same behavior that are not the same exact sets.

An obvious example is an instruction $\beta \cup \beta' \to b$ as opposed to a "smaller" instruction $\beta \to b$. If an element x contains the latter, it might as well contain the former, too. We say that the former instruction is *redundant* (in the presence of the latter).

That a graph algebra is never extensional doesn't necessarily imply graph models aren't either. Not all elements in the graph algebra underlying a model are hit by lambda terms, and it might just be be that the interpretation of any particular lambda term already contains all redundant instructions. However, this is not the case, as can be observed by comparing the interpretations of $\mathtt{I}$ and $\mathsf{1}$. Analogous to how $\beta \cup \beta' \to b$ is redundant in the presence of $\beta \to b$, so is $\{\beta \to b\} \to \beta \cup \beta' \to b$ in the presence of $\{\beta \to b\} \to \beta \to b$. The interpretation of $\mathsf{1}$ will always contain this latter redundant instruction, but $(\!|\mathtt{I}|\!)$ will not, so that $(\!|\mathsf{1}|\!) \setminus (\!|\mathtt{I}|\!) \neq \varnothing$. Recall that

$$(\!|\mathtt{I}|\!) = \{\alpha \to b \mid b \in \alpha\}$$
$$(\!|\mathsf{1}|\!) = \{\alpha \to \beta \to c \mid \exists \beta' \subseteq \beta,\ \beta' \to c \in \alpha\}$$

Indeed this argument quickly generalizes to show that always

$$(\lambda \vec{x}.\vec{x}) \neq (\lambda \vec{x}y.\vec{x}y).$$

What does this redundancy amount to? It turns out that the instruction metaphor previously mentioned gives rise to a very simple and intuitive characterization of redundancy: an instruction $\beta \to b$ is less informative than an instruction $\alpha \to a$ when it needs "more" input and generates "less" output, where "more" and "less" are defined again in terms of redundancy. Suppose that $(\preccurlyeq) \subseteq B \times B$ is a (pre-order) redundancy relation, and

$(\sqsubseteq) \subseteq B^{<\omega} \times B^{<\omega}$ is derived from it in an obvious way ($x \sqsubseteq y$ iff $\forall b \in x, \exists c \in y, b \preccurlyeq c$), then this characterization reads:

$$\beta \rightarrow b \preccurlyeq \alpha \rightarrow a \quad \text{iff} \quad \alpha \sqsubseteq \beta \wedge b \preccurlyeq a \tag{i$^-$}$$

Let us also immediately state a slight generalization, which we will be needing soon, where instead of a redundancy relation ($\preccurlyeq$) we have the "more informed" $(\vdash) \subseteq B^{<\omega} \times B$:

$$\beta \rightarrow b \dashv \{\alpha_i \rightarrow a_i \mid i < n\} \quad \text{iff} \quad \bigcup_i \alpha \sqsubseteq \beta \wedge b \dashv \{a_i \mid i < n\} \tag{i}$$

In fact, this is one of two conditions on ($\preccurlyeq$) that Hoofman and Schellinx, in [Hoofman and Schellinx, 1991], prove to be sufficient (and necessary) to construct an extensional model from a graph model. They were inspired by a specific redundancy relation defined by Bethke in [Bethke, 1986], where the "instruction" metaphor also finds its origin. The other condition regards *atoms* (instructions that are not "real" instructions of the form $\alpha \rightarrow b$ for certain α and b). Even though a graph model B may never be extensional, it doesn't even get close if it doesn't have $B^{<\omega} \times B \sim B$. The following proposition shows how, in order to even get the interpretations of $\mathbf{1}$ and $\mathtt{I}$ comparable, all instructions must be meaningful (of the shape $\alpha \rightarrow b$), so we cannot have any atoms.

Proposition 4.1. *$B^{<\omega} \times B \sim B$ if and only if $(\mathtt{I}) \subseteq (\mathbf{1})$.*

Proof. ($\Longrightarrow$) Easy. For $\alpha \rightarrow b \in (\mathtt{I})$ we have $b = \beta \rightarrow c$ by assumption and note that $\alpha \rightarrow \beta \rightarrow c \in (\mathbf{1})$.

($\Longleftarrow$) Easy. Take an arbitrary $b \in B$, $\alpha = \{b\}$, so that $\alpha \rightarrow b \in (\mathtt{I})$. By assumption, $\alpha \rightarrow b \in (\mathbf{1})$, and thus $b = \beta \rightarrow c$ for certain β, c. 證畢

This is why the second condition of Hoofman and Schellinx states that, although not every instruction must necessarily be a real instruction, it must at least identify with some finite set of real instructions, where identification is defined as mutual redundancy. We delightfully call this condition that of *squashing atoms.*

$$\forall a,\ \exists \beta \rightarrow b, \quad \beta \rightarrow b \preccurlyeq a \preccurlyeq \beta \rightarrow b \tag{ii$^-$}$$

And again, we have a slight generalization:

$$\forall a,\ \exists E_a \in B^{<\omega}, \quad E_a \sqsubseteq \{a\} \sqsubseteq E_a \tag{ii}$$

4.2 Extensionalizing via a redundancy pre-order

We now prove a slight variation of the result of [Hoofman and Schellinx, 1991] that, given a graph model (B, ι) with a pre-order $(\vdash)$ that satisfies above stated conditions (i) and (ii), we can construct an extensional model (strict reflexive CPO). The differences are:

- Instead of working with complete lattices, we work only with CPO's, simplifying the construction a bit by not having to account for non-directed sets;
- instead of taking equivalence classes, we take maximal elements, essentially $\bar{x} = \bigcup[x]_\equiv$);
- instead of being given a pre-order $(\preccurlyeq) \subseteq B \times B$, we use a "pre-order" $(\vdash) \subseteq B^{<\omega} \times B$.

Our choice to work with maximal elements instead of equivalence classes is not an important one. It simply corresponds more clearly to the extensionalization construction in [Plotkin, 1993] where Plotkin takes elements that are closed under a particular natural deduction system. For the same reason, of compatibility with Plotkin's construction, we generalized the usage of a pre-order $(\preccurlyeq)$ to a relation $(\vdash)$.

First, fix a "pre-order" $(\vdash)$ on $B^{<\omega} \times B$. Abuse notation and write $\beta \vdash \gamma$ for $\forall c \in \gamma,\ \beta \vdash c$ and $x \vdash c$ for $\exists \beta \subseteq x,\ \beta \vdash c$. Then it is clear what "pre-order" means, namely that the overloaded $(\vdash) \subseteq B^{<\omega} \times B^{<\omega}$ is a pre-order. Then we define the pre-order $(\sqsubseteq)$ and equivalence relation $(\equiv)$ over $\mathcal{P}_B$ by:

$$x \sqsupseteq y \quad \text{iff} \quad \forall c \in y,\ x \vdash c$$

$$x \equiv y \quad \text{iff} \quad x \sqsupseteq y \sqsupseteq x$$

As said, we will not work with equivalence classes, but rather with maximal elements $\bar{x}$ that are closed under redundancy:

$$\bar{x} = \{c \mid x \vdash c\}$$

This closure definition is a good one, because $\bar{x} = \bar{\bar{x}}$ follows from transitivity of $(\vdash)$, and we also easily see that $\bar{x} = \bigcup[x]_\equiv$.

Write, a bit inelegantly, $\overline{\mathcal{P}_\mathcal{B}}$ for the set $\{\bar{x} \mid x \in \mathcal{P}_B\}$. Now $(\overline{\mathcal{P}_\mathcal{B}}, \subseteq, \bigcup)$ is a CPO, because $(\bigcup)$ preserves suprema of directed sets within $\overline{\mathcal{P}_\mathcal{B}}$:

Lemma 4.2. *Taking unions of directed sets of maximal elements preserves maximality.*

Proof. Let X be a directed set of maximal elements. We must demonstrate that $\bigcup X$ is maximal, too. Suppose $b \dashv \beta \subseteq \bigcup X$, where, say, $\beta = \{b_i \in x_i \in X \mid i \ldots\}$. By directedness of X we have an $x \in X$ such that $\bigcup_i x_i \subseteq x$, and thus $b \dashv \beta \subseteq x$. By x being maximal we have $b \in x$, and thus finally $b \in \bigcup X$, making $\bigcup X$ maximal. 證畢

Let $\overline{F}$ such that $\overline{F}(x) = \overline{F(x)}$, and redefine G as:

$$\begin{aligned} G' : &\left[\left[\overline{\mathcal{P}_{\mathcal{B}}} \to \overline{\mathcal{P}_{\mathcal{B}}}\right] \to \overline{\mathcal{P}_{\mathcal{B}}}\right] \\ &f \mapsto \overline{\{\beta \to b \mid b \in f(\overline{\beta})\}} \end{aligned}$$

That G' is even really continuous we present in the following theorem, of which (1) and (2) are the main result of (section 3 of) [Hoofman and Schellinx, 1991]:

Theorem 4.3. *The CPO $(\overline{\mathcal{P}_{\mathcal{B}}}, \subseteq, \bigcup)$ together with $\overline{F}$ and G' is:*

1. *reflexive iff $(\vdash)$ satisfies $(i{\Rightarrow})$;*
2. *strict reflexive iff $(\vdash)$ satisfies both (i) and (ii).*

Additionally, $F = \overline{F}$ follows from $({\Leftarrow}i)$.

Proof. By the following lemma's, to be proven in the remainder of this section:

1. 4.7: G' is continuous, 4.8: $F \circ G' = \mathrm{id}_{[\overline{\mathcal{P}_{\mathcal{B}}} \to \overline{\mathcal{P}_{\mathcal{B}}}]}$
2. 4.9: $G' \circ F = \mathrm{id}_{\overline{\mathcal{P}_{\mathcal{B}}}}$ 證畢

First, we prove some basic results.

Lemma 4.4. *For any index set I and family of elements (x_i) we have $\overline{\bigcup_i x_i} = \overline{\bigcup_i \overline{x_i}}$.*

Proof. Compute:

$$\begin{aligned}
\overline{\bigcup_i \overline{x_i}} &= \{c \mid \bigcup_i \{d \mid x_i \vdash d\} \vdash c\} \\
&\subseteq \{c \mid \{d \mid \bigcup_i x_i \vdash d\} \vdash c\} = \overline{\{d \mid \bigcup_i x_i \vdash d\}} \\
&= \overline{\overline{\bigcup_i x_i}} = \overline{\bigcup_i x_i} \\
&\subseteq \overline{\bigcup_i \overline{x_i}}
\end{aligned}$$

證畢

Lemma 4.5. *For any x and y, $x \sqsubseteq y \iff \bar{x} \subseteq \bar{y}$. So for maximal elements, $(\subseteq)$ and $(\sqsubseteq)$ coincide.*

Proof. $(\implies)$ Let $b \dashv \alpha_1 \subseteq x$, we must prove that $b \dashv \alpha_1 \subseteq y$ for certain α_2. By $x \sqsubseteq y$ we know that there is $\alpha_1 \dashv \alpha_2 \subseteq y$. By transitivity of $(\vdash)$ we have $b \dashv \alpha_2 \subseteq y$, and we are done.

$(\impliedby)$ Let (without loss of generality) $\beta = \{b_1, b_2\} \subseteq x$, we must show that some $\gamma \subseteq y$ exists such that $\beta \dashv \gamma$. By $\bar{x} \subseteq \bar{y}$ we have (for $i = 1, 2$) $b_i \dashv \gamma_i \subseteq y$. Thus, taking $\gamma = \bigcup_i \gamma_i$, we have $\beta \dashv \gamma \subseteq y$. 證畢

Lemma 4.6. *If f is a continuous function over $\overline{\mathcal{P}_\mathcal{B}}$, and x some maximal element, then $b \in f(x) \iff \exists \beta \subseteq x,\ b \in f(\bar{\beta})$.*

That is, f is still defined by it's "finite" fragments, although we're working in the world of $\overline{\mathcal{P}_\mathcal{B}}$, in which such "finite" pieces such as $\bar{\beta}$ are actually always infinite sets.

Proof. $(\impliedby)$ Follows from monotonicity of f.

$(\implies)$ Observe that whenever $\beta \subseteq x$, $\bar{\beta} \subseteq x$ because of x being maximal, so that $x = \bigcup_{\beta \subseteq x} \beta = \bigcup_{\beta \subseteq x} \bar{\beta}$. And of course $\{\bar{\beta} \mid \beta \subseteq x\}$ is a directed set within $\overline{\mathcal{P}_\mathcal{B}}$. Thus $f(x) = f\big(\bigcup_{\beta \subseteq x} \bar{\beta}\big) = \bigcup_{\beta \subseteq x} f(\bar{\beta})$, and thus we must have that for certain $\beta \subseteq x$, $b \in f(\bar{\beta})$. 證畢

Lemma 4.7. *G' is a continuous mapping.*

Proof. First of all, monotonicity is obvious from it's definition, so let us check whether it preserves unions of directed sets.

Let $\mathscr{F}$ be a directed set of continuous functions within $\overline{\mathcal{P}_\mathcal{B}}$. We must demonstrate that $G'(\bigsqcup \mathscr{F}) = \bigcup_{f\in\mathscr{F}} G'f$. First, note that $G'[\mathscr{F}]$ is directed. Also note that we already proved $\overline{\bigcup_i x_i} = \overline{\bigcup_i \overline{x_i}}$ in a lemma 4.4. Now compute:

$$\begin{aligned}
G'\left(\bigsqcup \mathscr{F}\right) &= \overline{\{\beta \to b \mid b \in \left(\bigsqcup \mathscr{F}\right)(\overline{\beta})\}} && \text{(definition of } G')\\
&= \overline{\{\beta \to b \mid b \in \bigcup_{f\in\mathscr{F}} f(\overline{\beta})\}} && \text{(definition of } \textstyle\bigsqcup)\\
&= \overline{\bigcup_{f\in\mathscr{F}} \{\beta \to b \mid b \in f(\overline{\beta})\}} && \text{(shifting } \textstyle\bigcup)\\
&= \overline{\bigcup_{f\in\mathscr{F}} \overline{\{\beta \to b \mid b \in f(\overline{\beta})\}}} && \text{(lemma 4.4)}\\
&= \bigcup_{f\in\mathscr{F}} \overline{\{\beta \to b \mid b \in f(\overline{\beta})\}} && (\textstyle\bigcup \text{ preserves maximality})\\
&= \bigcup_{f\in\mathscr{F}} G'f && \text{(definition of } G')
\end{aligned}$$

證畢

Lemma 4.8. *If we have (i⇒), then* $\overline{F} \circ G' = id_{[\overline{\mathcal{P}_\mathcal{B}} \to \overline{\mathcal{P}_\mathcal{B}}]}$ *making the CPO* $(\overline{\mathcal{P}_\mathcal{B}}, \subseteq, \bigcup)$ *reflexive.*

Proof. Let f be a continuous function over $\overline{\mathcal{P}_\mathcal{B}}$, and x a maximal element. We must show that $\overline{F}(G'(f))(x) = f(x)$. First, compute:

$$\begin{aligned}
\overline{F}(G'(f))(x) &= \overline{F}\left(\{\alpha \to a \mid a \in f(\overline{\alpha})\}\right)(x)\\
&= \overline{\{b \mid \exists\beta \subseteq x,\ \beta \to b \dashv \{\alpha_i \to a_i \mid a_i \in f(\overline{\alpha_i})\}\}}
\end{aligned}$$

(⊇) This direction is easy. By lemma 4.6 we have some $\beta \subseteq x$ with $b \in f(\overline{\beta})$. Reflexivity of $(\vdash)$ gives us $\beta \to b \dashv \{\beta \to b\}$, and then $b \in \overline{F}(G'(f))(x)$ as you can see from the computation above (taking $\{\alpha_i \to a_i \mid i \dots\} = \{\beta \to b\}$).

(⊆) Let $b \in \overline{F}(G'(f))(x)$, say because of $\beta \subseteq x$ with $\beta \to b \dashv \{\alpha_i \to a_i \mid a_i \in f(\overline{\alpha_i})\}$. Condition (i⇒) gives us that $\bigcup_i \alpha_i \sqsubseteq \beta$ and $b \dashv \{a_1, \dots a_n\}$. Then by lemma 4.5

we have $\bigcup_i \alpha_i \subseteq \overline{\beta}$ and by monotonicity of f:

$$b \dashv \{a_1, \ldots a_n\} \subseteq f(\overline{\bigcup_i \alpha_i}) \subseteq f(\overline{\beta}) \subseteq f(x)$$

By maximality of $f(x)$ we conclude that $b \in f(x)$. 證畢

Lemma 4.9. *If we have* (*i*) *and* (*ii*)*, then* $G' \circ \overline{F} = id_{\overline{\mathcal{P}_\mathcal{B}}}$ *making the CPO* $(\overline{\mathcal{P}_\mathcal{B}}, \subseteq, \bigcup)$ *strict reflexive.*

Proof. Let x be an arbitrary maximal element. We must show that $G'(\overline{F}(x)) = x$. First, compute:

$$G'(\overline{F}(x)) = \overline{\{\beta \to b \mid b \in \overline{F}(x)(\overline{\beta})\}} = \overline{\{\beta \to b \mid \exists\gamma \subseteq \overline{\beta},\ \gamma \to b \in x\}}$$

($\supseteq$) Let $a \in x$. Suppose that a is a real instruction, say $a = \beta \to b$. Then we just take $\gamma = \beta \subseteq \overline{\beta}$ to see that indeed $a \in G'(\overline{F}(x))$. Now suppose a is atom, so by (ii)$^+$ and x being maximal we have real instructions $E_a \subseteq x$. From the just proved result about real instructions we have $E_a \subseteq G'(\overline{F}(x))$, and then by maximality of $G'(\overline{F}(x))$ and (ii) we have $a \in G'(\overline{F}(x))$.

($\subseteq$) First, consider an arbitrary $\beta \to b$ for which we have a $\gamma \subseteq \overline{\beta}$ with $\gamma \to b \in x$. By lemma 4.5, $\gamma \subseteq \overline{\beta}$ implies $\gamma \sqsubseteq \beta$, and then by ($\Leftarrow$i), we have $\beta \to b \in x$, too. Because, in general, $y \subseteq x$ implies $\overline{y} \subseteq x$ for maximal x, we now have $G'(\overline{F}(x)) \subseteq x$, completing the proof. 證畢

Because of the following lemma we have $\overline{F} = F$, given that ($\vdash$) satisfies ($\Leftarrow$i):

Lemma 4.10. *F preserves maximality if* ($\vdash$) *satisfies* ($\Leftarrow$*i*).

Proof. Let x be maximal. Suppose $b \dashv \alpha \subseteq Fxy$. Because (say) $\alpha = \{a_1, \ldots a_n\}$, and for each i: $\alpha_i \subseteq y$ and $\alpha_i \to a_i \in x$. Letting $\beta = \bigcup_i \alpha_i$, it follows from ($\Leftarrow$i) that $\beta \to b \in x$, so that $b \in F(x)(y)$. Thus $F(x)(y)$ is maximal, too. 證畢

4.3 An instance: Plotkin's natural deduction method

In [Plotkin, 1993], Plotkin essentially implements this idea, and defines the redundancy relation ($\vdash$) is terms of a natural deduction system. In the paper, he uses it on his variation of graph models, and the atom set is the singleton set $\{\varepsilon\}$. "Porting" his construction to ordinary graph models, and using an arbitrarily large atom set, the construction is as follows.

The redundancy relation $(\vdash)$ is defined as derivation within the natural deduction system given by the following three inference rules, in which to every atom a, a finite set of real instruction E_a is assigned:

$$\text{for each atom } a\text{:} \quad \frac{E_a}{a} \quad \text{and} \quad \frac{a}{E_a} \qquad \text{(Squashing atoms)}$$

$$\frac{\alpha_1 \to a_1 \qquad \cdots \qquad \alpha_n \to a_n \qquad \begin{matrix}[\beta] \\ \alpha_1 \dots \alpha_n\end{matrix} \qquad \begin{matrix}[a_1 \dots a_n] \\ b\end{matrix}}{\beta \to b} \qquad \text{(Redundancy)}$$

Here, we abuse the natural deduction notation a bit by placing a whole set of instructions under a tree instead of just one (in the case of $\alpha_1 \dots \alpha_n$ and E_a). Naturally, placing a set β under a tree we mean that for every $b \in \beta$, the same tree exists with only b at the bottom. Note that the example that we started with follows from (Redundancy) and the way natural deduction systems work:

$$\frac{\beta \to b \qquad \begin{matrix}[\beta \cup \beta'] \\ \beta\end{matrix} \qquad [b]}{\beta \cup \beta' \to b} \ \text{(Redundancy)}$$

As we are dealing with a natural deduction system, $(\vdash)$ is automatically a pre-order. Applying our earlier construction, we now get $(\sqsubseteq)$, $(\equiv)$, $\bar{x}$, $\overline{\mathcal{P}_{\mathcal{B}}}$ and G'.

It should be obvious that $(\vdash)$ satisfies both conditions.

Lemma 4.11. *Plotkin's $(\vdash)$ satisfies condition (i).*

Proof. Recall that (i) states:

$$\beta \to b \dashv \{\alpha_i \to a_i \mid i < n\} \quad \text{iff} \quad \bigcup_i \alpha \sqsubseteq \beta \wedge b \dashv \{a_i \mid i < n\} \qquad \text{(i)}^+$$

$(\Longleftarrow)$ This follows directly from (Redundancy).

$(\Longrightarrow)$ By induction on the structure of the proof of the antecedent. The rules of (Squashing atoms) do not apply, because $c \neq \beta \to b$ and $c \neq \alpha_i \to a_i$ for any atom c or index $i < n$. Consider the case of (Redundancy). Let us say the last proof rule was applied as follows:

$$\dfrac{\gamma_1 \to c_1 \qquad \cdots \qquad \gamma_k \to c_k \qquad \overset{[\beta]}{\gamma_1, \dots \gamma_k} \qquad \overset{[c_1, \dots c_k]}{b}}{\beta \to b}\ \text{(Redundancy)}$$

The induction hypothesis gives us (in particular) that for $j = 1 \dots k$, $\bigcup_i \alpha_i \sqsubseteq \gamma_j$ and $c_j \dashv \{a_1, \dots a_n\}$. Combining this, by transitivity of $(\vdash)$, with $\{\gamma_1, \dots \gamma_k\} \sqsubseteq \beta$ and $b \dashv \{c_i, \dots c_k\}$ we get $\bigcup_i \alpha_i \sqsubseteq \beta$ and $b \dashv \{a_1, \dots a_n\}$. 證畢

Note that this lemma is the counterpart of lemma 2.2 in Plotkin's paper. Though in his paper, he had more inference rules so that this proof-theoretic result was slightly less obvious.

Lemma 4.12. *Plotkin's* $(\vdash)$ *satisfies condition* (*ii*).

Proof. This follows even more obviously from the inference rules. 證畢

Comparing our inference rules to Plotkin's original ones, we see that Plotkin had way more than we do (note that he identified the single atom ε with a finite set of real instructions E):

$$\dfrac{}{\varnothing \to \varnothing} \qquad \text{(Axiom 1)}$$

$$\dfrac{E}{\varepsilon} \ \text{and}\ \dfrac{\varepsilon}{E} \qquad \text{(Rules 1 and 2)}$$

$$\dfrac{\mu \to (\nu \cup \nu_2)}{(\mu \cup \mu_2) \to \nu} \ \text{and}\ \dfrac{\mu \to \nu \qquad \mu_2 \to \nu_2}{(\mu \cup \mu_2) \to (\nu \cup \nu_2)} \qquad \text{(Rules 3 and 4)}$$

$$\dfrac{\mu \to \nu \qquad \overset{[\mu_2]}{\mu} \qquad \overset{[\nu]}{\nu_2}}{\mu_2 \to \nu_2} \qquad \text{(Rule 5)}$$

The reason for this is twofold. First, the normal variation of graph models, in which the right-hand side of an instruction contains only a single instruction instead of a finite set just like the left-hand side, causes Axiom 1 and Rule 4 to lose meaning. Second, Rule 3 was never necessary in the first place, as it follows from Rule 5 (as we demonstrated in our case of normal graph models, earlier).

4.4 K-models

Independent of the results treated so far by Bethke, Hoofman, Schellinx and Plotkin, similar results about extensional graph-like models were obtained by Krivine [1990]. Berline [2000] systematically presents a whole slew of variations of the standard graph model, one of which is based upon Krivine's work. In the way she presents these K-models, they are exactly the same with the difference that she works with a binary pre-order ($\preccurlyeq$) instead of a relation ($\vdash$) like we did. Recall that this difference is one we introduced, generalizing Hoofman/Schellinx' construction, in order for it to be more directly compatible with Plotkin's natural deduction system, and thus is of no (formal) significance.

Bibliography

Henk Barendregt. *The Lambda Calculus, its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. 1984.

Chantal Berline. From computation to foundations via functions and application: The λ-calculus and its webbed models. *Theoretical Computer Science*, 249(1):81–161, 2000.

Ingemarie Bethke. How to construct extensional combinatory algebras. *Indagationes Mathematicae (Proceedings)*, 89(3):243–257, 1986.

Antonio Bucciarelli and Antonino Salibra. Graph lambda theories. *Mathematical Structures in Computer Science*, 18(5):975–1004, 2008.

Pietro Di Gianantonio, Furio Honsell, and Gordon Plotkin. Uncountable limits and the lambda calculus. 1995.

Erwin Engeler. Algebras and combinators. *Algebra Universalis*, 13(1):389–392, 1981.

Raymond Hoofman and Harold Schellinx. Collapsing graph models by preorders. In D.H. Pitt, P.-L. Curien, S. Abramsky, A.M. Pitts, A.l Poigné, and D.E. Rydeheard, editors, *Category Theory and Computer Science*, number 530 in Lecture Notes in Computer Science, pages 53–73. Springer-Verlag Berlin, 1991.

Rainer Kerth. Isomorphism and equational equivalence of continuous λ-models. *Studia Logica*, 61(3):403–415, 1998.

Jean Louis Krivine. *Lambda-Calcul, Types et modèles*. Masson, Paris, 1990. English translation: Lambda-calculus, types and models, René Cori, 2002.

Gordon D. Plotkin. Set-theoretical and other elementary models of the λ-calculus. *Theoretical Computer Science*, 121(1–2):351–409, 1993.

Peter Selinger. Order-incompleteness and finite lambda reduction models. *Theoretical Computer Science*, 309(1–3):43–63, 2003.