On constructions of partial combinatory algebras and their relation to topology

Niels Voorneveld

Master thesis
Department of Mathematics

Supervisor: Prof. dr. Jaap van Oosten
Second reader: Prof. dr. Albert Visser

Mathematical Sciences
University of Utrecht
The Netherlands
August 2015

Abstract

In this thesis we study models of partial combinatory algebras and we aim to construct new examples. Mainly, the sets of representable functions of these models and their connections to topology are explored. One famous example is Scott's Graph model, which can simulate the untyped lambda calculus. Using a theorem by Jaap van Oosten, we study an extension of this Graph Model in which the complement function is representable. This new model is decidable. Relations with other models yield two more decidable extensions, with interesting categorical properties. Additionally, a model is constructed that has a connection to the Cantor topology, and a model is made using the power set of another model.

Contents

Chapter 1

Introduction

Realizability was first conceptualised by Stephen Cole Kleene in 1945 [10]. He constructed a method to assign numbers to mathematical statements in such a way that tells something about the truth of that statement. It gave a classical interpretation of Bouer's Intuitionism, yet was not quite like it. Other mathematicians started looking for similar systems, and the field of realizability was born.

One of the various constructions of realizability is that of the partial combinatory algebra (pca), which was first formally defined by Feferman [5]. It uses combinatory logic to define a structure on a variety of different sets. Models by Kleene could be seen as examples of this type of construction, using Gödel numbering or a system of maps between natural numbers. The recent rise of computer science highlights another interesting property of pca's, the fact that they can be used as register machines. They can simulate recursive functions using a representation of the natural numbers within the system. But what is central to the importance of the pca in the field of realizability might be the discovery of a certain topos. It started with Martin Hyland, who together with others detailed the construction of the effective topos. This structure was akin to Kleene's original system of realizability, and by its topos-theoretic aspects generated concepts of truth in higher function spaces. Later studies by John Longley yielded a more general construction of toposes, using as its base a partial combinatory algebra. A pca could be used to construct a realizability topos in two steps, which retained the interesting properties of the underlying model.

In chapter 2, an overview of preliminary information about partial combinatory algebras is given. This thesis treats with three fundamental examples of pca's. Mainly Scott's Graph model, defined on the power set of the natural numbers. But also Kleene's second model on the set of functions on the natural numbers, and the universal domain model. All three of them have a tight connection with topology, meaning the set of representable functions is the same as the set of continuous functions of a specific topology. These fundamental pca's are also strongly linked. Each one can be modelled in the other two using what is known as applicative morphisms. In Bauer's thesis [1], specific applicative morphisms are defined that have additional properties. One pair forms a retraction and one an inclusion. Pca's can never realize all functions between its elements. In Van Oosten's book [16], a method is detailed to construct a new pca from an old one that adds a function to the set of representable functions. This new *extension* can be seen as the least pca to do this.

In chapter 3, we will consider the construction of toposes from pca's. Applicative morphisms give rise to functors, and sometimes even geometric morphisms between their realizability toposes. A brief overview of this method of constructing toposes and functors will be given.

In chapter 4, we will study various constructions of new models. By adding the complement function to Scott's Graph model, I find an extension that unlike its predecessor is decidable. This means that one can check whether two elements are equal. I also further study the relation between Scott's Graph model and various different topologies. Two other extensions, of Kleene's second model and of the universal domain model, are found. The applicative morphisms between the original pca's can be lifted to function as applicative morphisms between the extensions. This means they are all decidable. The diagrams that are created in this system of extensions form pull-back squares. Neither Scott's Graph model nor its extension with the complement function have a set of realizable functions that is the same as the set of functions continuous in the Cantor topology. I construct a new pca called the Double Graph Model that does have this property. This pca can be modelled within Kleene's second model. From any pca, we can also construct a new one on its power set, which has interesting properties. At the end, I briefly discuss the possible implementation of the system of extensions within a computer. In the same way, extensions can be found on the recursive sub-pca's which have the same properties. We end the thesis with some conclusions in chapter 5.

Chapter 2

Partial Combinatory Algebras

2.1 Basic definitions and concepts

In this section we introduce the main concept of this thesis, Partial Combinatory Algebras, which were first defined by Feferman [5]. These structures use combinatory logic and can be used to model different structures from various fields. We will look at representability and decidability, which are a basic concepts of computability. We will discuss the basic examples of a partial computable algebra and look at the connection between representability and topology.

2.1.1 Basics

One way to look at pca's is by considering them as a computer. It has a set of elements and a set of programs that can act on these elements. But unlike with computers, these two sets are the same. Any element can be seen as both a program and an input. How these programs act on the elements is coded with a specific type of map. We get a basic type of structure.

Definition 2.1.1. A *partial applicative structure (PAS)* is a set A together with a partial map, $A \times A \rightharpoonup A$.

The result of this map on the input (a, b) we write as ab. We see ab as the result of a being applied to b. So a is the program, b the input and ab the output. Since the map need not be total, there is a possibility that ab is undefined. We write that fact as $ab\uparrow$. If ab is defined we write $ab\downarrow$. When using the map multiple times, we will use left association, so for instance $ac(bc)$ means the same thing as $(ac)(bc)$.

With this map, we can construct much more complex operations then just letting one element act on the other. We can do that using terms.
If we take a set V containing an infinite number of variables and a pas A, we can define the set of *terms* $E(A)$ as the smallest set such that:

- $V \subseteq E(A)$
- $A \subseteq E(A)$
- $s, t \in E(A) \Rightarrow (st) \in E(A)$

When writing terms, we again use left association. So a term might look like $x_1a(bx_3a)c$ with $a, b, c \in A$ and $x_1, x_3 \in V$. Now we need a way for the terms to be related to elements of A. When a term does not contain any variables from V, we call it *closed*. In that case, it is just some combination of elements of A, which itself can be seen as an element of A if their application is defined. This way of comparing terms with elements of A we write as $t \downarrow a$, which says that the term t denotes the element $a \in A$. The rules for this are:

- $a \downarrow a$ for all $a \in A$
- $st \downarrow a$ if and only if there are $b, c \in A$ such that $s \downarrow b$, $t \downarrow c$ and $a = bc$.

We can see from this definition, that each term can denote at most one element of A. However, it is possible for two closed terms s, t to denote the same element. If that is the case, we will consider them equal, $s = t$. This makes closed terms indistinguishable from the elements they denote.
With s, t closed terms, we can also define equivalence as $s \simeq t$ meaning, if $s\downarrow$ (there is an $a \in A$ such that $s \downarrow a$) then $s = t$. We can extend that notation to all terms by saying that $s \simeq t$ if for any substitution of variables with elements of A, the resulting closed terms are equivalent. We can write such a substitution as $t[s/x]$ meaning substitute within term t the variable x with term s.

With the construction of terms, we have a set of all possible maps we can cre-

ate by using variables, constants and the application map. For a non-closed term t, we write $t(x_0, ..., x_n)$ as the term/map over the variables $\{x_0, ...x_n\}$. We do need all free variables of the term to be included in the variables used by the map.
We can see those maps as the computable functions within the confines of our definitions. However, these maps are not always easy to write. It would be handy if a term could be represented by a single element, like how in programming a computation can be represented by a single number. We write that as follows:

Definition 2.1.2. A term $t(x_0, ..., x_n)$ is represented by $a \in A$ if for all $a_0, a_1, ..., a_n \in A$ we have:
- $aa_0a_1...a_{n-1} \downarrow$
- if $t[a_0/x_0, a_1/x_1, ..., a_n/x_n] \downarrow$ then $aa_0a_1...a_n = t[a_0/x_0, a_1/x_1, ..., a_n/x_n]$

We can now define what it means to be a partial combinatory algebra.

Definition 2.1.3. A *partial combinatory algebra* (pca) is a pas A which is *combinatory complete*, meaning that every term $t(x_0, ..., x_n)$ is represented by some $a \in A$.

Traditionally, we do not except the fact that representations might be defined on a wider range then the terms that they represent. However, by a result by Faber [3] we know that any pca defined in this sense is isomorphic to a pca in the more traditional sense. Another way to define a pca is by using combinators. There is a definition which asks the pas to have two specific elements. The following theorem describing this definition is by Feferman.

Theorem 2.1.4. *A pas A is a pca if and only if there are elements $k, s \in A$ such that for all $a, b, c \in A$ we have:*
1) $kab = a$
2) $sab \downarrow$
3) If $ac(bc) \downarrow$ then $sabc = ac(bc)$

Proof: If A is a pca, then by taking the term $t(x_1, x_2) = x_1$ we see that by combinatory completeness, there is an element $k \in A$ such that 1) is satisfied. For 2) we can use the term $t(x_1, x_2, x_3) = x_1x_3(x_2x_3)$ which by combinatory completeness yields an element $s \in A$ such that 2) holds.
For the converse, let A be a pas with k, s satisfying point 1 and 2. We will construct every term using this k and s as follows:

Definition 2.1.5. For every term $t \in E(A)$ and variable $x \in V$ we define $\langle x \rangle t$ with induction on $E(A)$:
-$\langle x \rangle x = skk$
-$\langle x \rangle t = kt$ if $t = b \in A$ or $t = y \in V$ with $y \neq x$
-$\langle x \rangle t_1 t_2 = s(\langle x \rangle t_1)(\langle x \rangle t_2)$

Note that for every $a \in A$ we have that $skka = ka(ka) = a$, so it gives the identity map. We denote it as $i := skk$. If we look at the definition, we see that $\langle x \rangle t$ does not have x as a variable. Specifically, if we have a term $t(x_0, ..., x_n)$, then we have that $\langle x_0 \rangle t$ runs over the variables $(x_1, ..., x_n)$ and for all $a_0, a_1, ..., a_n \in A$ we get that if $t(a_0, a_1, ..., a_n) \downarrow$ then $(\langle x_0 \rangle t)(a_1, ..., a_n)a_0 = t(a_0, a_1, ..., a_n)$. Hence, inductively we find that $\langle x_0 \rangle (...(\langle x_{n-1} \rangle (\langle x_n \rangle t))..)$ represents t. So the pas is combinatory complete, hence a pca.

$\square$

2.1.2 Pairing and decidability

We have seen that any pca has elements k and s acting as defined in theorem 2.1.4. These elements are not necessarily unique, so it is important to choose specific elements before we can use them. We can use combinatory completeness to construct more elements, which we can use to compute interesting things from within the structure and show the representability of certain functions. In this subsection we will discuss some of them.

Remember that in the proof of the previous theorem, we use the notation $\langle x \rangle t$ to transform the term into an element which represents it. To simplify the notation, we will write using left association, $\langle x_0 x_1 ... x_n \rangle t := \langle x_0 \rangle (...(\langle x_{n-1} \rangle (\langle x_n \rangle t))..)$

Now, consider any pca $\mathcal{A}$ with some choice of k and s. First we define the elements which can be used for the concepts of truth and false written as T and F and called the *Booleans*. We desire that there is a term t such that for all a, b we have $tTab = a$ and $tFab = b$. A possible choice for this taking $t = i$, $T = k = \langle xy \rangle x$ and $F = \langle xy \rangle$. We also have for all closed terms a, b a term t such that $tT = a$ and $tF = b$. Simply take $t = \langle x \rangle xab$. This is often called the 'if .. then a else b' term. For the following results, we fix the choice of Booleans as given above.

Now we can define a *pairing* on $\mathcal{A}$ by taking $p := \langle xyz \rangle zxy$, $p_0 := \langle w \rangle wT$ and $p_1 := \langle w \rangle wF$. With these we can code two elements $a, b \in \mathcal{A}$ as one and we have a way to reconstruct the original from these elements. The pairing is done using p to create $pab = \langle z \rangle zab$. With this element we have that $p_0(pab) = pabT = Tab = a$ and $p_1(pab) = pabF = Fab = b$. The element p is also called the pairing element and p_0 and p_1 the projection elements.
We extend this concept to code any number of elements. Define inductively for all natural numbers n the pairing maps $j^n : \mathcal{A}^n \to \mathcal{A}$ by taking:

$$j^1(a) = a$$

$$j^n(a_1, ..., a_n) = pa_1 j^{n-1}(a_2, ..., a_n)$$

It is also possible to study the natural numbers within any (non-trivial) pca. We do that by representing the numbers as numerals.

Definition 2.1.6. In a pca $\mathcal{A}$, the *Curry numerals* are inductively defined as:
$\overline{0} = i$ and for all $n \geq 0$: $\overline{(n+1)} = pF\overline{n}$.

One can code any partial recursive function using these numerals. This is for instance discussed in [16]. A consequence of this is that we can encode the set of finite sequences. Let $a_0, a_1, ..., a_{n-1}$ be a finite sequence, we define $[a_0, a_1, ..., a_{n-1}] \in \mathcal{A}$ inductively:

$$[] := p\overline{0}\overline{0}$$

$$[a_0, a_1, ..., a_{n-1}] = p\overline{n} j^n(a_0, ..., a_{n-1}) \text{ if } n > 0$$

It is possible to find elements that can compute from this coding several attributes. For instance the length of the sequence, a specific part of the sequence or the code of the concatenation of two sequence. These things will be useful later.
With Booleans, we can define what it means for certain objects to be decidable.

Definition 2.1.7. We call a subset $B \subseteq A$ *decidable* if there is an element $a \in \mathcal{A}$ such that for all $b \in A$, $ab = T$ if $b \in B$ and $ab = F$ if $b \notin B$.

We can see that if B and C are both decidable, then so are $B \cup C$ and $B \cap C$. Take b and c the representations of the decidability of B and C respectively, then $\langle x \rangle bxT(cx)$ and $\langle x \rangle bx(cx)F$ represent $B \cup C$ and $B \cap C$ respectively.

We can define the decidability of relations in a similar way. The most common relation studied in pca's is the most simple one, equality. In general, it is not possible to effectively check whether two elements are the same.

Definition 2.1.8. A pca A is called *decidable* if there is an element d such that for all $a, b \in A$ we have $dab = T$ if $a = b$ and $dab = F$ if $a \neq b$.

2.1.3 Examples

The simplest example of a pca is one on the singleton set $\{*\}$. All elements are the same and we have the applicative structure of $** = *$. We write this pca as $\mathcal{I}$ and call it the trivial pca. All pca's besides this one are a great deal more interesting. Here are some results:

Proposition 2.1.9. *Let $\mathcal{A}$ be a non-trivial pca, then:*
1) If k and s are as usual, than $k \neq s$
2) $\mathcal{A}$ is not commutative
3) $\mathcal{A}$ is not associative
4) $\mathcal{A}$ is infinite

Proof:
1) If $k = s$, then $i = skk = kkk = k$. Hence for all $a \in \mathcal{A}$ we have $a = iia = kka = k$, so $\mathcal{A}$ is trivial. This leads to a contradiction.
2) If $\mathcal{A}$ were commutative, then $i = skk = kks = k$. So by 1) we have that $\mathcal{A}$ is trivial.
3) If $\mathcal{A}$ were associative, then for all $a \in \mathcal{A}$ we have $a = kak = (kkk)ak = k(kk)ak = (kk)k = k$, so $\mathcal{A}$ is trivial.
4) We have that $s \neq i$, so $s^n \neq s^m$ for all natural numbers $n \neq m$. So there must be infinitely many elements.

□

The best known non-trivial pca is *Kleene's first model*, also written as $\mathcal{K}_1$. It is the structure on the natural numbers $\mathbf{N}$ that uses the coding of partial recursive function. The application is defined by $(a, b) \mapsto \phi_a(b)$.

Lastly, we can look at substructures of pca's.

Definition 2.1.10. Let $\mathcal{B}$ be a pca. A pca $\mathcal{A}$ is a *sub-pca* of $\mathcal{B}$ if it is a subset of $\mathcal{B}$, the application on that subset is the same as in $\mathcal{B}$ and a choice of k and s for $\mathcal{B}$ is contained in $\mathcal{A}$.

2.1.4 Topologies and representability

Take a non-trivial pca $\mathcal{A}$. We have that for any n the closed terms can never describe all partial maps from $\mathcal{A}^n$ to $\mathcal{A}$. This is because the cardinality of the set of all total maps is always higher than the amount of elements within $\mathcal{A}$. So we say that not all maps are representable.

Definition 2.1.11. Let $\mathcal{A}$ be a pas. A partial map $f : \mathcal{A}^n \rightharpoonup \mathcal{A}$ is *representable* if there is an element $a \in \mathcal{A}$ satisfying for all $b_0, b_1, ..., b_{n-1} \in \mathcal{A}$, if $f(b_1, ..., b_{n-1}) \downarrow$ then $ab_1b_2...b_{n-1} = f(b_1, ..., b_{n-1})$.

If we are working within a pca, there will always be a natural way to look at the set of representable functions, since we can look at the recursive construction of the terms. But for some pca's there is another easier way to characterize the set of representable functions. In some cases we can use topologies. A quick reminder of the definition of a topology. A *topology* on a set X is a set of opens $T \subset \mathcal{P}(X)$ which contains $\emptyset$ and X and which is closed under finite intersections and all unions. Examples are the *indiscrete* topology $\{\emptyset, X\}$ and the *discrete* topology $\mathcal{P}(X)$. For $U \in T$ we call U open and $X - U$ closed. A *basis* of a topology T is a subset $B \subseteq T$ such that the closure of that set under finite intersections and arbitrary unions is the topology T itself. A map $f : X \to Y$ is continuous according to the topologies T_X on X and T_Y on Y if for each $U \in T_Y$ (open), $f^{-1}(U) \in T_X$.
Now for its connection with pca's. Let $\mathcal{A}$ be a pas. A topology on a pas is a topology on the underlying set of the pas. Such topologies can have the following properties:

Definition 2.1.12. A topology T on a pas $\mathcal{A}$ is a *repcon topology* if all representable endomaps on $\mathcal{A}$ are continuous on their domain.
A topology T on a pas $\mathcal{A}$ is a *conrep* topology if all continuous endomaps in T are representable in $\mathcal{A}$.
If a topology T on a pas $\mathcal{A}$ is both repcon and conrep, and $\mathcal{A}$ is also a pca, then we call $(\mathcal{A}, T)$ a *pca-topology pair.*

For any pca, there are always two examples of repcon topologies, the discrete and the indiscrete topologies. Of the other repcon topologies, we can distinguish a special type.

Definition 2.1.13. A repcon topology T on $\mathcal{A}$ is said to be *minimal* if it is not indiscrete and for any repcon topology R on $\mathcal{A}$ such that $R \subset T$ either $R = T$ or $R = T_{indiscrete}$.

Note that in the definitions of representability, repcon and conrep topologies do not require the pas to be a pca. This is since these properties may arise before combinatory completeness has been established. These properties may even help in proving it.

The definitions of conrep and repcon only use endomaps, but we can look at maps on higher dimensions. We call a topology *n-conrep* if all continuous n dimensional maps are representable and *n-repcon* if vice versa. We call it *full-conrep* if it is n-conrep for all n and *full-repcon* if n-repcon for all n.

Results:
1) For a pas A, if the application map is continuous by a certain topology, then that topology is full-repcon w.r.t. A. This can be seen by noting that constant maps and identity maps are continuous by any topology. So any closed term is a composition of continuous maps.
2) For a pca $\mathcal{A}$, if a topology is conrep (1-conrep) and 2-repcon w.r.t. $\mathcal{A}$, it is full-conrep. This can be seen by using composition with the pairing maps.

2.2 Three pca's

Apart from the trivial pca and Kleene's first model, there are a lot more interesting yet fairly basic pca's. In this chapter, we will discuss three different main examples that are often discussed in literature and have a fundamental role in this thesis, and all three of them form a pca-topology pair with some topology. The results in this chapter are discussed in [16] and [1].

2.2.1 Scott's Graph model

The first pca we will discuss is by D.S. Scott [14]. It is denoted by **P** and is defined on the set of subsets $\mathcal{P}(\omega)$. The application map is constructed as follows. Firstly, we take a coding of pairs in $\mathbf{N}$. We write $\langle a, b\rangle$ as the number representing the code of two natural numbers a and b. A choice of code can be $\langle a, b\rangle := (a^2 + b^2 + 3a + b)/2$. Secondly, we code the finite subsets of $\mathbf{N}$ as follows: each $p \subset \mathbf{N}$ finite is linked to the natural number $\Sigma_{n\in p}2^n$. Take e_n to be the finite subset belonging to number n. We define the application as follows, for two sets $A, B \in \mathcal{P}(\omega)$:

$$AB = \{m : \exists n(e_n \subset B, \langle n, m\rangle \in A)\}$$

This gives us a pas on $\mathcal{P}(\omega)$. We still need to establish if this is a pca. To do this, we study its relation to the Sierpinski product topology on the set. This topology is defined on $\{0, 1\}^{\mathbf{N}}$ as the product of the Sierpinski topology $\{\emptyset, \{1\}, \{0, 1\}\}$ on $\{0, 1\}$. Using the traditional bijection between $\{0, 1\}^{\mathbf{N}}$ and $\mathcal{P}(\omega)$ we get a topology on the latter set.
This topology has a basis of open sets of the form $U_p := \{B : p \subset B\}$ where $p \subset \mathbf{N}$ is finite. We also have that for finite p, $U_p = \bigcap_{x\in p} U_{\{x\}}$ a finite intersection of other opens. So we can say that $\{U_{\{x\}} : x \in \mathbf{N}\}$ gives a simpler basis of the Sierpinski product topology. Now for its connection with our pas.

Lemma 2.2.1. *The application map $p : \boldsymbol{P} \times \boldsymbol{P} \to \boldsymbol{P}$ is continuous in the Sierpinski product topology.*

Proof: Take x a natural number. $p^{-1}(U_{\{x\}}) = \{(A, B) : x \in A \cdot B\} = \{(A, B) : \exists n, (e_n \subset B, \langle n, x\rangle \in A)\} = \bigcup_n \{U_{e_n} \times U_{\{\langle n,x\rangle\}}\}$ which is a union of opens. So p is continuous.

$\square$

Now we can derive from this that the maps $k : (a, b) \mapsto a$ and $s : (a, b, c) \mapsto a \cdot c \cdot (b \cdot c)$ on $\mathcal{P}(\omega)$ are continuous in the Sierpinski product topology. So in general, what does it mean to be continuous in this topology?

Lemma 2.2.2. *A map $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ is continuous in the Sierpinski product topology if and only if for all $A \subset \boldsymbol{N}$ we have $F(A) := \bigcup\{F(p) : p \subset A, p$ finite$\}$.*

Proof: For two basis elements U_p and U_q of the topology, with p and q finite, we have $U_p \cap U_q = U_{p\cap q}$ which is another element of the basis. So we can write any open V as the union of basic opens.
Assume F is continuous and take $A \subset \mathbf{N}$. Then $A = \bigcup\{p : p \subset A, p \text{ finite}\}$. Denote $V_1^n = \{B : n \in F(B)\}$ which is open, so it can be written as the union of basic opens $V_1^n = \bigcup_{i\in I_i} U_{p_i^n}$. Fix an n.
If $n \in F(A)$, then $A \in V_1^n$, so there is an i such that $A \in U_{p_i^n}$, hence $p_i^n \subset A$ and $n \in F(p_i^n)$. On the other hand, if there is a finite $p \subset A$ such that $n \in F(p)$, then $p \in V_1^n$ so $p \in U_{p_i^n}$ for some i. So $p_i^n \subset p \subset A$, hence $A \in V_1^n$, meaning $n \in F(A)$. We can conclude that $F(A) := \bigcup\{F(p) : p \subset A, p \textit{ finite}\}$.
Now assume the contrary: for F we have that for all A, $F(A) := \bigcup\{F(p) : p \subset A, p \text{ finite}\}$ (which is always possibly since $F(p)$ is in $\mathcal{P}(\omega)$ and hence any union is in $\mathcal{P}(\omega)$). Take a natural number n. We can see that $\{A : n \in F(A)\} = \{A : n \in \bigcup\{F(p) : p \subset A, p \text{ finite}\}\} = \bigcup\{\{A : p \subset A\} : p \text{ finite}, n \in F(p)\} = \{U_p : p \text{ finite}, n \in F(p)\}$ which is a countable union of open sets, hence itself open. So F is continuous.

$\square$

Lemma 2.2.3. *All continuous maps $F : \mathcal{P}(\omega)^k \to \mathcal{P}(\omega)$ are representable.*

Proof: For any continuous map $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ in the Sierpinski product topology, we can define the set $Graph(F) := \{\langle n, m\rangle : m \in F(e_n)\}$. For such a set we have that $Graph(F) \cdot A = \{m : \exists n(e_n \subset A, \langle n, m\rangle \in Graph(F))\} = \{m : \exists n(e_n \subset A, m \in F(e_n))\} = \bigcup\{F(e_n) : e_n \subset A\} = \bigcup\{F(p) : p \subset A, p \text{ finite}\} = F(A)$. Hence $F(A) = Graph(F) \cdot (A)$ for all A.
For $F : \mathcal{P}(\omega)^k \to \mathcal{P}(\omega)$ continuous with arbitrary k, define $Graph(F) := \{\langle n_1, \langle n_2, \langle .... \langle n_k, m\rangle ..\rangle\rangle : m \in F(e_{n_1} \times ... \times e_{n_k})\}$. Then for $X_1, .., X_k \in \mathbf{P}$ you can see with induction that $Graph(F) \cdot X_1 \cdot ... \cdot X_k = \{\langle n_2, \langle .... \langle n_k, m\rangle ...\rangle\rangle : \exists n_1, (e_{n_1} \subset X_1, m \in F(e_{n_1} \times ... \times e_{n_k})\} = \{m : \exists n_1, ..., n_k, (e_{n_1} \subset X_1, ..., e_{n_k} \subset X_k, m \in F(e_{n_1} \times ... \times e_{n_k}))\} = \{F(p_1, ..., p_k) : p_1 \subset X_1, ..., p_k \subset X_n \text{ finite}\} = F(X_1, ..., X_k)$.

$\square$

So we can conclude that we can find elements k and s satisfying theorem 2.1.4, so $\mathbf{P}$ is a pca.

An interesting property of this pca is that it is *total*, meaning the application map is total. With the retraction $\Lambda : \mathbf{P} \to \mathbf{P}^{\mathbf{P}}$ given by $\Lambda(a) = (b \mapsto ab)$ and the section $Graph : \mathbf{P}^{\mathbf{P}} \to \mathbf{P}$ together with the natural pairing, it is possible to model the *untyped lambda calculus* in this pca. This is a construction of effective computable total functions without hierarchy.

2.2.2 Kleene's second model

The second pca we will discuss in this section is Kleene's second model $\mathcal{K}_2$ on the set $\mathbf{N}^{\mathbf{N}}$. It forms a pair with the traditional topology of that space, which can be described as the infinite product of the discrete topology on $\mathbf{N}$. For each finite sequence $\sigma = (k_1, ..., k_n)$ of elements in $\mathbf{N}^{\mathbf{N}}$, it has an open $U_\sigma = \{\alpha \in \mathbf{N}^{\mathbf{N}} : \sigma \sqsubseteq \alpha\}$, the set of sequences starting with σ. These opens form a basis of the topology.

We use a coding of finite sequences into natural numbers denoted by $\langle \alpha_1, ..., \alpha_n \rangle$ or $seq(\alpha)$. Denote for an infinte sequence α, $\bar{\alpha}n = \langle \alpha(1), ..., \alpha(n) \rangle$ and let $\langle n \rangle * \alpha$ be the sequence given by:

$$(\langle n \rangle * \alpha)(x) = \begin{cases} n & \text{if } x = 0 \\ \alpha(x-1) & \text{otherwise} \end{cases}$$

For each $\alpha \in \mathbf{N}^{\mathbf{N}}$ we define a map $F_\alpha : \mathbf{N}^{\mathbf{N}} \to \mathbf{N}$ which sends $\beta \mapsto k$ if there is an n such that $\alpha(\bar{\beta}n) = k + 1$ and for all $m < n$, $\alpha(\bar{\beta}n) = 0$. If no such k exists, the function is undefined. The functions of this form are precisely the functions which are continuous according to the previously defined topology (and the discrete topology on $\mathbf{N}$).
We define the application on $\mathbf{N}^{\mathbf{N}}$ as follows, if for all n, $F_\alpha(\langle n \rangle * \beta) \downarrow$, then $\alpha\beta \downarrow$, else $\alpha\beta \uparrow$. If it is defined, then:

$$\alpha\beta(n) = F_\alpha(\langle n \rangle * \beta)$$

This application map gives us a pas on $\mathbf{N}^{\mathbf{N}}$. It is easy to see that any map by this application map is continuous by the previously defined topology. The converse is a bit more tricky.

Theorem 2.2.4. *Any continuous partial function $F : \boldsymbol{N}^{\boldsymbol{N}} \times \boldsymbol{N}^{\boldsymbol{N}} \rightharpoonup \boldsymbol{N}^{\boldsymbol{N}}$ is representable by this pas.*

Proof: Take F such a function. Define for all natural numbers n and k, $V_n^k := \{(\alpha, \beta) \in \mathbf{N}^\mathbf{N} \times \mathbf{N}^\mathbf{N} : F(\alpha, \beta)(n) = k\}$. We define an element $\gamma \in \mathbf{N}^\mathbf{N}$ inductively as follows, using the coding of finite sequences into natural numbers. Let $\gamma(\langle\rangle) = 0$ and for all finite sequences σ and τ and all natural numbers n:

$$\gamma(\langle\langle\rangle\rangle * \sigma) = 1$$

$$\gamma(\langle\langle n\rangle * \sigma\rangle * \tau) = \begin{cases} 1 & \text{if } \neg\exists k : U_\sigma \times U_\tau \subseteq V_n^k \\ k+2 & \text{if } U_\sigma \times U_\tau \subseteq V_n^k \end{cases}$$

Note that for $k \neq m$, $V_n^k \cap V_n^m = \emptyset$. So for σ and τ there is at most one k such that $U_\sigma \times U_\tau \subseteq V_n^k$. Since F is continuous, and $\{\delta : \delta(n) = k\}$ is open, we have that V_n^k is open. Hence for every $(\alpha, \beta) \in V_n^k$ there are σ and τ such that $(\alpha, \beta) \in U_\sigma \times U_\tau \subseteq V_n^k$. So we find that $\gamma\alpha\beta = (((k+2) - 1) - 1) = k$ by our application.

□

Corollary 2.2.5. *The defined pas is a pca.*

Proof: By the previous lemma we have found that the topology is both 1-conrep (by adding a useless first argument) and 2-conrep. So an element k as in theorem 2.1.4 can be found. Since we can easily define continuous pairing functions, which are then representable, we can conclude that the pas is full-repcon. With the application being continuous, we can say that the function $abc \mapsto ac(bc)$ is continuous. Using pairing we can represent it. So $\mathbf{N}^\mathbf{N}$ with the application is a pca.

□

2.2.3 Universal domain model

The last main example of a pca looks at the theory of domains and is discovered by Scott [13]. Consider a partial order D. A *directed subset* $X \subseteq D$ is a non-empty set with the property that for all $x, y \in X$ there is a $z \in X$ with $x \leq z$ and $y \leq z$. A *bounded subset* $B \subseteq D$ is a set such that there is a $z \in D$ where for all $x \in B$, $x \leq z$. z is also called a *bound* for B. For a subset $X \subseteq D$, a *least upper bound* $\bigvee X$ is the smallest possible bound (which does not always exist). We say that D is *directed complete*, if directed subsets have a least upper bound. D is *bounded complete* if all bounded sets have a least upper bound.

Assume D is directed complete. We call $a \in D$ *compact* if for each directed subset $X \subseteq D$ such that $a \leq \bigvee X$, there is a $b \in X$ such that $a \leq b$. Let $\mathcal{K}(D)$ be the set of compact elements.

Definition 2.2.6. A *domain* D is a partially ordered set which is directed complete and bounded complete with a countable subset $B \subseteq \mathcal{K}(D)$ such that $\forall a \in D : a = \bigvee\{x \in B : x \leq a\}$.

On a domain D, we have a structure called the *Scott topology*, which is the topology with opens $U \subseteq D$ such that:
1) $\forall x \in U, \forall y \geq x : y \in U$
2) For each directed subset $X \subseteq D$, $\bigvee X \in U \Rightarrow X \cap U \neq \emptyset$

This topology has a basis of opens formed by the sets $\uparrow a := \{x \in D : a \leq x\}$ with a compact. For D a domain, we denote $[D \to D]$ for the set of continuous endomaps. An *embedding-projection pair* between two domains D and E is a pair (i, p) of continuous maps with $i : D \to E$ an embedding and $p : E \to D$ a projection such that $p \circ i = 1_D$ and $i \circ p \leq 1_E$.

As discussed in [1] there is a domain $\mathbf{U}$ called the *universal domain* which has the property that for each domain D there is an embedding projection pair (i_D, p_D) from D to $\mathbf{U}$. Combining this with the fact that $[\mathbf{U} \to \mathbf{U}]$ is also a domain, we have an embedding projection pair $(i_{[\mathbf{U}\to\mathbf{U}]}, p_{[\mathbf{U}\to\mathbf{U}]})$ from $[\mathbf{U} \to \mathbf{U}]$ to $\mathbf{U}$. We can hence define the applicative structure $\mathbf{U} \times \mathbf{U} \to \mathbf{U}$ by sending $x, y \to (p_{[\mathbf{U}\to\mathbf{U}]}(x))(y)$. This gives us a structure that is provably a pca.

We can give a concrete example of $\mathbf{U}$. We look at the Cantor topology on $\mathcal{P}(\omega) = \{0,1\}^{\mathbf{N}}$, which is the product of the discrete topology on $\{0,1\}$. So a basis is formed by the open sets: $U_p^q := \{U \subseteq \mathcal{P}(\omega) : p \in U, q \cap U = \emptyset\}$, where p and q are finite.
The set of Cantor open subsets of $\mathcal{P}(\omega)$, excluding the complete set, gives an example of the universal domain model. The order is defined by inclusion and the compact elements of this set are precisely the clopens, the open subsets for which the complement is also open. We can write it as $\mathbf{U} = \mathcal{O}(2^{\mathbf{N}}) - 2^{\mathbf{N}}$.

2.3 Applicative morphisms and assemblies

In this section, we study ways in which we can relate pca's with each other, and structures like assemblies and predicates that can be defined on pca's. The results in this chapter are discussed in [16].

2.3.1 Applicative morphisms

In order to compare different pca's we need a way to relate them. We do this using applicative morphisms, which allow us to establish that one model can be simulated within another model.

Definition 2.3.1. Let $\mathcal{A}, \mathcal{B}$ be two pca's. An applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$ is a map from $\mathcal{A}$ to $\mathcal{P}^*(\mathcal{B}) := (\mathcal{P}(\mathcal{B}) - \{\emptyset\})$ such that there is an $r \in \mathcal{B}$ where for all $a, a' \in \mathcal{A}$, $b \in \gamma(a)$ and $b' \in \gamma(a')$ we have that if $aa' \downarrow$ then $rbb' \downarrow$ and $rbb' \in \gamma(aa')$.

We say that the r in the definition *realizes* γ. If we have two applicative morphisms $\gamma, \delta : \mathcal{A} \to \mathcal{B}$ then we write $\gamma \preceq \delta$ if there is a t such that for all $a \in \mathcal{A}$ with $b \in \gamma(a)$ we have $tb \downarrow$ and $tb \in \delta(a)$. We say that γ and δ are isomorphic if both $\gamma \preceq \delta$ and $\delta \preceq \gamma$, we write that as $\gamma \sim \delta$.
If we have two applicative morphisms $\gamma : \mathcal{A} \to \mathcal{B}$ and $\delta : \mathcal{B} \to \mathcal{C}$ then we can compose them creating the morphism $\delta \circ \gamma$ where $a \mapsto \bigcup_{b \in \gamma(a)} \delta(b)$. If r realizes γ and t realizes δ, then $\langle x \rangle t(rx)$ realizes $\delta \circ \gamma$.
There are some possible properties an applicative morphism can have:

Definition 2.3.2. Let $\gamma : \mathcal{A} \to \mathcal{B}$ be an applicative morphism.
1) γ is *discrete* if for all $a, a' \in \mathcal{A}$ with $a \neq a'$ we have $\gamma(a) \cap \gamma(a') = \emptyset$.
2) γ is *single-valued* if for all $a \in \mathcal{A}$, $\gamma(a)$ only has one element.
3) γ is *projective* if it is isomorphic to a single-valued morphism.
4) γ is *decidable* if there is a $d \in \mathcal{A}$ called the *decider* such that for all $b \in \gamma(T_{\mathcal{A}})$ and $b' \in \gamma(F_{\mathcal{A}})$ we have $db = T_{\mathcal{B}}$ and $db' = F_{\mathcal{B}}$. Here $T_{\mathcal{A}}$ and $F_{\mathcal{A}}$ are the Booleans of $\mathcal{A}$, and $T_{\mathcal{B}}$ and $F_{\mathcal{B}}$ the Booleans of $\mathcal{B}$.
5) γ is *computationally dense* if there is an $m \in \mathcal{B}$ such that: $\forall b \in \mathcal{B}, \exists a \in \mathcal{A}, \forall a' \in \mathcal{A} : b\gamma(a') \downarrow \Rightarrow (aa' \downarrow \wedge m\gamma(aa') = b\gamma(a'))$

Examples:
1) For any pca $\mathcal{A}$, we have an identity pca $id_{\mathcal{A}} : \mathcal{A} \to \mathcal{A}$ given by $a \mapsto \{a\}$.

Its realizer is ι.
2) If we have two pca's $\mathcal{A} \subseteq \mathcal{B}$, the morphism $\iota : \mathcal{A} \to \mathcal{B}$ sending $a \mapsto \{a\}$ is applicative.
3) We have a unique applicative morphism from any pca $\mathcal{A}$ to the trivial pca $\mathcal{I}$. Conversely, for any non-empty subset $A \subseteq \mathcal{A}$, we have an applicative morphism $\mathcal{I} \to \mathcal{A}$ with $* \mapsto A$.
4) For any pca $\mathcal{A}$, there is an applicative morphism $\mathcal{K}_1 \to \mathcal{A}$ sending n to $\{\overline{n}\}$ in $\mathcal{A}$, where $\{\overline{n}\}$ is the n-th Curry numeral.

If an applicative morphism is computationally dense, then it is also decidable. There is an easier method to check whether an applicative morphisms is computationally dense, by P. Johnstone [9]:

Theorem 2.3.3. *An applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$ is computationally dense if and only if there is a function $g : \mathcal{B} \to \mathcal{A}$ and an element $r \in \mathcal{B}$ such that for all $b \in \mathcal{B}$ and $c \in \gamma(g(b))$ we have $rc = b$.*

2.3.2 Extensions of pca's

We have seen that a partial map $f : \mathcal{A} \rightharpoonup \mathcal{A}$ may not always be representable. However, it may be possible to describe it by using an applicative morphism.

Definition 2.3.4. Let $\gamma : \mathcal{A} \to \mathcal{B}$ be an applicative morphism and $f : \mathcal{A} \rightharpoonup \mathcal{A}$ a partial map. f is *representable* w.r.t. γ if there is a $t \in \mathcal{B}$ such that for all $a \in \mathrm{dom}(f)$ and all $b \in \gamma(a)$ we have $tb\downarrow$ and $tb \in \gamma(f(b))$.

So it says that f is representable in the way that γ maps $\mathcal{A}$ into $\mathcal{B}$. With this, we have the following fundamental theorem by [16].

Theorem 2.3.5. *For every pca $\mathcal{A}$ and every partial map $f : \mathcal{A} \rightharpoonup \mathcal{A}$ there exists a pca $\mathcal{A}[f]$ with the same underlying set as $\mathcal{A}$, such that:*
1) f is representable w.r.t. the decidable applicative morphism $\iota_f : \mathcal{A} \to \mathcal{A}[f]$ given by $a \mapsto \{a\}$.
2) For every decidable applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$ such that f is representable by γ, there is a unique decidable applicative morphism $\gamma_f : \mathcal{A}[f] \to \mathcal{B}$ such that $\gamma_f \circ \iota_f = \gamma$. Moreover, if $\delta : \mathcal{A}[f] \to \mathcal{B}$ such that $\delta \circ \iota_f \sim \gamma$ then $\delta \sim \gamma_f$.

This theorem constructs a new pca from the old one, which can be seen as the best pca to both represent f and any function representable by the old pca. We call $\mathcal{A}[f]$ an *extension* of $\mathcal{A}$.
The proof is written out in [16]. Here we will just discuss the construction of the application map on $\mathcal{A}[f]$. If a is the code for $a = [a_1, a_2, ..., a_{n-1}]$ and if $i < n$, denote $a^{<i} := [a_1, a_2, ..., a_{i-1}]$ and ${}^{i\leq}a := [a_i, a_{i+1}, ..., a_{n-1}]$. If $a = [a_1, a_2, ..., a_{n-1}]$ and $b = [b_1, b_2, ..., b_{m-1}]$ let $a * b = [a_1, ..., a_{n-1}, b_1, ..., b_{m-1}]$. Let $N \in \mathcal{A}$ such that with the Booleans T and F in $\mathcal{A}$ we have $NT = F$ and $NF = T$. We define a new application $\cdot^f$ on the underlying set of $\mathcal{A}$. For $a, b \in \mathcal{A}$ we define an *f-dialogue* between a and b as the code of a sequence $u = [u_1, ..., u_{n-1}]$ such that:

$$\forall i < n, \exists v_i \in \mathcal{A} : (a([b] * u^{<i}) = pFv_i) \wedge f(v_i) = u_i$$

We define $a \cdot^f b$ to be c if there is an f-dialogue u between a and b such that $a([b] * u) = pTc$. This c is unique if it exists. If it doesn't exist the application is not defined.

Corollary 2.3.6. *Let $\mathcal{A}$ be a pca.*
1) If f is representable with respect to $\mathcal{A}$ then $\mathcal{A}[f] \simeq \mathcal{A}$.
2) For f, g to partial endomaps on $\mathcal{A}$, we have $\mathcal{A}[f][g] \simeq \mathcal{A}[g][f]$.
3) Every total pca is isomorphic to a non-total one.
4) The applicative morphism $\iota_f : \mathcal{A} \to \mathcal{A}[f]$ as constructed in the previous theorem is computationally dense.

Proof: 1) Let i be the identity morphism of $\mathcal{A}$. Then f is representable w.r.t. i, so there is a decidable applicative morphism $i_f : \mathcal{A}[f] \to \mathcal{A}$ such that $i_f \circ \gamma_f = i$. So indeed, $\mathcal{A}[f] \simeq \mathcal{A}$.
2) Same method as in 1 to prove the existence of compatible decidable applicative morphisms between $\mathcal{A}[f][g]$ and $\mathcal{A}[g][f]$.
3) Use 1 on a total pca $\mathcal{A}$. The result is a non-total isomorphic pca $\mathcal{A}[f]$.
4) Simple consequence of theorem 2.3.3, by considering the map $g : \mathcal{A}[f] \to \mathcal{A}$ with $a \mapsto a$, and the element i.

We can also look at partial functions in more than one variable. Using the coding of sequences, we can observe when such functions are represented by an applicative morphism. It is possible to extend 2.3.5 to also include these multivalued functions. This can be done by using the pairing functions which are representable in the space, to transform f into a function with one variable and apply the theorem on that endofunction. This new space will represent f and be optimal in the same way as in the theorem.

2.3.3 Assemblies

In this section, we look at collections of sets on pca's which can be related to each other in the same way as applicative morphisms relate pca's to each other.

Definition 2.3.7. Let $\mathcal{A}$ be a pca. An *assembly* on $\mathcal{A}$ is a set X together with a map $E : X \to \mathcal{P}^*(\mathcal{A})$.
A *modest set* is an assembly (X, E) such that for all $x, y \in X$ with $x \neq y$ we have $E(x) \cap E(y) = \emptyset$.

Given an applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$, we immediately have an assembly $(\mathcal{A}, \gamma)$ on $\mathcal{B}$. If γ is discrete, then $(\mathcal{A}, \gamma)$ is a modest set.
Assemblies do not necessarily use the structure of a pca, only the underlying sets. The computable effects can be studied when looking at morphisms between them.

Definition 2.3.8. A *morphism* between two assemblies $f : (X, E) \to (Y, F)$ is a map $f : X \to Y$ such that there is an element $a \in \mathcal{A}$ with the property that for all $x \in X$ and $b \in E(x)$ we have $ab\downarrow$ and $ab \in F(f(x))$.

Such an element a is also called a *tracking* of the morphism. For any assembly (X, E) we have the identity morphism tracked by i.
Note that for two applicative morphisms $\gamma, \delta : \mathcal{A} \to \mathcal{B}$ such that $\gamma \preceq \delta$, we have a morphism between $(\mathcal{A}, \gamma)$ and $(\mathcal{A}, \delta)$ given by the identity function on $\mathcal{A}$ and which has as its tracking the element that represents $\gamma \preceq \delta$.

2.3.4 Heyting pre-algebras and Predicates

Predicates are use to look at pca's in a more abstract manner. They are Heyting pre-algebras which give rise to the construction of realizability toposes discussed in chapter 3.

Definition 2.3.9. A *preorder* $(P, \leq)$ is a set P together with a binary relation $\leq$ which is reflexive and transitive.

A preorder may have a *top element* $\top$ and *bottom element* $\perp$ such that for all $a \in P$ we have $a \leq \top$ and $\perp \leq a$. For each preorder $(P, \leq)$ we can define an equivalence relation $a \cong b \Leftrightarrow (a \leq b \,\&\, b \leq a)$. We get a new preorder on the set of equivalence classes $P/\cong$ with the relation $[a] \leq [b] \Leftrightarrow a \leq b$. We get a partial order $(P/\cong, \leq)$ called the *poset reflection* of $(P, \leq)$.
By adding operation we can construct the following structure:

Definition 2.3.10. A *Heyting pre-algebra* is a preorder $(P, \leq)$ with a top and a bottom element together with three binary operations $\wedge$, $\vee$ and $\Rightarrow$ such that for all $a, b, c \in P$:
1) $a \leq (b \wedge c) \Leftrightarrow (a \leq b \,\&\, b \leq c)$
2) $(a \vee b) \leq c \Leftrightarrow (a \leq c \,\&\, b \leq c)$
3) $(a \wedge b) \leq c \Leftrightarrow a \leq b \Rightarrow c$

Note that the poset reflection of a Heyting pre-algebra is again a Heyting pre-algebra.

Definition 2.3.11. Let $\mathcal{A}$ be a pas and X a set. A $\mathcal{P}(\mathcal{A})$*-valued predicate* on X is a function $\phi : X \to \mathcal{P}(\mathcal{A})$

Note here that $\mathcal{A}$ need not be a pca and the resulting sets $\phi(x)$ can be empty. If $a \in \phi(x)$, then we call a a realizer of $\phi(x)$.
Like with applicative morphisms we can relate two $\mathcal{P}(\mathcal{A})$-valued predicates ϕ and ψ on X, to each other by writing $\phi \leq \psi$ if there is an $a \in \mathcal{A}$ such that for all $x \in X$ and all $b \in \phi(x)$ we have $ab\downarrow$ and $ab \in \psi(x)$. In this case we say that $\phi \leq \psi$ is *realized* by a. We get the following result:

Proposition 2.3.12. *If $\mathcal{A}$ is a pca and X a set. The relation $\leq$ from above make the set of $\mathcal{P}(\mathcal{A})$-valued predicates on X into a Heyting pre-algebra.*

Proof: Note that $\phi \leq \phi$ is realized by i. If $\phi \leq \psi$ is realized by a and $\psi \leq \chi$ is realized by b, then $\langle x \rangle b(ax)$ realizes $\phi \leq \chi$. So it forms a preorder.
To prove that it forms a Heyting algebra, define the following predicates and operations. Take $\top$ and $\perp$ to be predicates such that $\top(x) = \mathcal{A}$ and $\perp(x) = \emptyset$ for all $x \in X$. For all predicates ϕ and ψ define:
$(\phi \wedge \psi)(x) = \{pab : a \in \phi(x), b \in \psi(x)\}$
$(\phi \vee \psi)(x) = \{pTa : a \in \phi(x)\} \cup \{pFa : a \in \psi(x)\}$
$(\phi \Rightarrow \psi)(x) = \{a \in \mathcal{A} : \forall b \in \psi(x), (ab\downarrow, ab\psi(x))\}$
To finish the proof, we need to check the following things:

$\chi \leq (\phi \wedge \psi) \Leftrightarrow (\chi \leq \phi, \chi \leq \psi)$
$(\phi \vee \psi) \leq \chi \Leftrightarrow (\phi \leq \chi, \psi \leq \chi)$
$(\phi \wedge \psi) \leq \chi \Leftrightarrow \phi \leq \psi \Rightarrow \chi$
These facts can be proven by manipulating realizers.

□

2.4 Comparing pca's

The pca's discussed in chapter 2.2 have important applicative morphisms that relate them to each other. Some of these morphisms form interesting pairs making an applicative retraction and an applicative inclusion. The results in this chapter are discussed in [1].

2.4.1 Kleene's second and Scott's graph model

We can define an applicative retraction between $\mathcal{K}_2$ and $\mathbf{P}$.
Remember that the code of a finite sequence a is denoted by $seq(a)$. We define a map $\iota : \mathcal{K}_2 \to \mathbf{P}$ as $\alpha \mapsto \{seq(a) : a \in \mathbf{N}^*, a \sqsubseteq \alpha\}$, the set of all finite initial sequences of α encoded as natural numbers. Let $\mathbf{P}' := im(\iota)$ and look at the map $p : \mathbf{P}' \times \mathbf{P}' \to \mathbf{P}$ defined by $(\iota(\alpha), \iota(\beta)) \mapsto \iota(\alpha\beta)$ if $\alpha\beta$ is defined, and $(\iota(\alpha), \iota(\beta)) \mapsto \emptyset$ otherwise. This map is continuous on the Sierpinski product topology, so it can be represented in $\mathbf{P}$ by some element p. Hence, ι is an applicative morphism as a map $a \mapsto \{\iota(a)\}$.
Take $T_{\mathcal{K}_2}$ and $F_{\mathcal{K}_2}$ to be the Booleans of Kleene's second model, and $T_{\mathbf{P}}$ and $F_{\mathbf{P}}$ the Booleans of Scott's Graph model. Note that since $T_{\mathcal{K}_2} \neq F_{\mathcal{K}_2}$, there are $a, b \in \mathbf{N}^*$ such that $a \sqsubseteq T_{\mathcal{K}_2}$, $b \sqsubseteq F_{\mathcal{K}_2}$, $\neg(b \sqsubseteq T_{\mathcal{K}_2})$ and $\neg(a \sqsubseteq F_{\mathcal{K}_2})$. We define a map $d : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$ sending $A \mapsto \{x : seq(a) \in A, x \in T_{\mathbf{P}}\} \cup \{y : seq(b) \in A, x \in F_{\mathbf{P}}\}$. This map sends $\iota(T_{\mathcal{K}_2})$ to $T_{\mathbf{P}}$ and $\iota(F_{\mathcal{K}_2})$ to $F_{\mathbf{P}}$. It is also continuous, so it has a representation which acts as a decider for ι. Hence ι is decidable.

We define another morphism $\delta : \mathbf{P} \to \mathcal{K}_2$ sending A to $\{\alpha : \{n : \exists i, \alpha(i) = n + 1\} = A\}$, the set of all sequences whose non-zero value coincide with the set of successors of A. We use successors since we want the image of the empty set to be non-empty. To see that this is an applicative morphism, we

need a realizer ρ in Kleene's second model such that for all A, B, $\alpha \in \delta(A)$ and $\beta \in \delta(B)$ we have $\rho\alpha\beta \in \delta(AB)$.
Let $\rho : \mathcal{K}_2 \times \mathcal{K}_2 \to \mathcal{K}_2$ be the map such that $\rho(\alpha, \beta)$ sends $\langle n, m\rangle$ (coding of pairs) to $n+1$ if there is an $i < m$ and a j such that $\beta(i) = 1 + \langle n, j\rangle$ and for all $x \in e_j$ there is a $k < m$ such that $\alpha(k) = x$. If not, it sends $\langle n, m\rangle$ to 0. This map is continuous, since the $i < m$ and $k < m$ conditions make sure that only finite initial sequences are used as a reference in its construction. If $\alpha \in \delta(A)$ and $\beta \in \delta(B)$, all non-zero values of $\rho(\alpha, \beta)$ are equal to $x+1$ for some $x \in AB$. So $\rho(\alpha, \beta) \in \delta(AB)$, hence δ is an applicative morphism.
Let $x \in T_{\mathbf{P}} - F_{\mathbf{P}}$ and $y \in F_{\mathbf{P}} - T_{\mathbf{P}}$ (it is easy to see that there is a choice of T and F such that they exist). We define the map $d : \mathcal{K}_2 \times \mathcal{K}_2 \to \mathcal{K}_2$ which sends α to $T_{\mathcal{K}_2}$ if there is a k such that $\alpha(k) = x+1$, or to $F_{\mathcal{K}_2}$ if there is a k such that $\alpha(k) = y+1$. Let it be undefined if either none or both of the above conditions are satisfied. This map sends $\alpha \in \delta(T_{\mathbf{P}})$ to $T_{\mathcal{K}_2}$ and $\alpha \in \delta(F_{\mathbf{P}})$ to $F_{\mathcal{K}_2}$. This map is also continuous on its domain, and hence it has a representation which acts as a decider for δ. So δ is decidable.

A property of these applicative morphisms is that they form an *applicative retraction* $\iota \dashv \delta$. This means that $\delta \circ \iota \sim 1_{\mathcal{K}_2}$ and $\iota \circ \delta \preceq 1_{\mathbf{P}}$. The proof of this fact is briefly sketched in Bauer's thesis [1].

2.4.2 Universal domain model and Scott's graph model

In this section we will look at the relation between the Graph model $\mathbf{P}$ and the Universal domain model $\mathbf{U}$. We have seen that one way to represent $\mathbf{U}$ is by looking at the Cantor open sets. $\mathbf{U}$ can be seen as the set of all opens except the total set $2^{\mathbf{N}}$. We have also seen that it forms a pca-topology pair with the Scott topology, which has a basis of opens of the following form: For every clopen $U \subseteq 2^{\mathbf{N}}$, $\uparrow U := \{V \in \mathbf{U} : U \subset V\}$ is an open set.

First we look at what it means to be Cantor-clopen. We have seen that basic opens in this topology are of the form U_p^q with p and q inite. Note that with a, b, c, d four finite sets, $U_a^b \cap U_c^d = U_{a\cap c}^{b\cap d}$ is another basic open. So any Cantor-open set can be written as $V := \bigcup_{i\in I} U_{p_i}^{q_i}$. Now it is not difficult to see that V is a clopen if and only if it can be written as a finite union of that form. Since we can code the finite sets, we can conclude that the set of clopens is countable. We use a numbering of clopens: denote B_n as the n-th clopen. Define for each natural number n the following open

$C_n := U_{\{n\}}^{\{0,1,\dots,n-1\}}$. Note that $\{C_n\}_{n\in\mathbf{N}}$ gives a partition of $2^{\mathbf{N}}$.

We can now define the following decidable applicative morphisms:

$$\zeta : \mathbf{P} \to \mathbf{U}, A \mapsto \bigcup_{n\in A} C_n$$

$$\eta : \mathbf{U} \to \mathbf{P}, V \mapsto \{n \in \mathbf{N} : B_n \subset V\}$$

By Bauer **??** we know that these indeed are applicative morphisms and that they form an *applicative inclusion* $\eta \dashv \zeta$, which means that $\eta \circ \zeta \sim 1_{\mathbf{P}}$ and $\zeta \circ \eta \preceq 1_{\mathbf{U}}$.

Chapter 3

Realizability toposes

In this chapter, we will discuss how every pca gives rise to a topos called a realizability topos. These toposes function as models of realizability, translating concrete systems of computability to more general objects that represent effective computing in a more pure form. A detailed description of this construction and its properties is given in Van Oosten's book [16]. Hofman gives a short version in his paper [6]. Here we will look at a brief summary of category theory together with the core concepts of the construction of realizability toposes and geometric morphisms. These will hopefully give a general idea of how the structures found in chapter 4 can be lifted into the topos-theoretic space.

3.1 Category theory basics

We start with the general concepts of category theory. Categories are structures of objects with arrows between them. Many mathematical structures can be seen as examples of categories, including the structures that have been previously defined in this thesis. We begin with the definition of a category.

Definition 3.1.1. A *category* C consists of and satisfies the following things:
1) It has a collection of *objects*, denoted $ob(\mathbf{C})$.
2) It has a collection of *morphisms* (or arrows), denoted $hom(\mathbf{C})$. Each morphism goes from an object to another object, it has a *source* object and a *target* object. For $A, B \in ob(\mathbf{C})$, we write $\mathbf{C}(A, B)$ for the collection of

morphisms from A to B. For $f \in \mathbf{C}(A,B)$ we write $f : A \to B$.
3) For all $A, B, C \in ob(\mathbf{C})$ we have a map $\mathbf{C}(A,B) \times \mathbf{C}(B,C) \to \mathbf{C}(A,C)$ which is called the *composition of morphisms*. For $f : A \to B$ and $g : B \to C$, we write the result of the composition as $f \circ g$.
4) It must satisfy *associativity*, meaning for all $f : A \to B$, $g : B \to C$ and $h : C \to D$ we have $(h \circ g) \circ f = h \circ (g \circ f)$.
5) For each $A \in ob(\mathbf{C})$ there is a morphism $1_A \in \mathbf{C}(A,A)$ called the *identity morphism* of A such that for all $B \in ob(\mathbf{C})$, $f \in \mathbf{C}(A,B)$ and $g \in \mathbf{C}(B,A)$ we have $f \circ 1_A = f$ and $1_A \circ g = g$.

Examples:
1) The most general example is the category of sets **SET**. Here the objects are all possible sets, and the morphisms are the maps between sets. Composition is simply composition of maps.
2) We have the category of pca's **PCA**, where the objects are pca's and the morphisms are applicative morphisms.
3) The category of assemblies (**ASS**), and its sub-category of the modest sets (**MOD**) with morphisms as in definition 2.3.8.
4) The category of Heyting algebras, which has as morphisms the maps that preserve the structure of $\leq, \wedge, \vee, \Rightarrow, \top$ and $\perp$.

For each category $\mathbf{C}$ we can define the *dual category* $\mathbf{C}^{op}$ which has the same objects, but the source and the target of the morphisms are switched. So $\mathbf{C}^{op}(A,B) = \mathbf{C}(B,A)$ and the arguments in composition are switched.

Definition 3.1.2. Let $\mathbf{C}, \mathbf{D}$ be two categories. A *functor* F from $\mathbf{C}$ to $\mathbf{D}$ maps each object $A \in ob(\mathbf{C})$ to an object $F(A) \in ob(\mathbf{D})$ and each morphism $f \in \mathbf{C}(A,B)$ to a morphism $F(f) \in \mathbf{D}(F(A), F(B))$ such that the identity morphism and the composition of morphisms is preserved.

Definition 3.1.3. Let $\mathbf{C}$ be a category. A *pull-back square* in $\mathbf{C}$ is a commutative diagram:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g"'] & Y \arrow[d, "h"] \\ Z \arrow[r, "k"] & W \end{tikzcd}$$

with the property that for every object A and every pair of maps $f_A : A \to Y$

and $g_A : A \to Z$ such that $h \circ f_A = k \circ g_A$, there is a unique map $p : A \to X$ satisfying $f \circ p = f_A$ and $g \circ p = g_A$.

One can also see a pull-back square as a diagram that satisfies what is sometimes called the *universal property*. In other words, X together with f and g are optimal in commuting with the rest of the diagram. We find another example of this universal property in finite products:

Definition 3.1.4. Let $\mathbf{C}$ be a category and $X_1, ..., X_n$ a collection of objects. An object X together with morphisms $\pi_i : X \to X_i$ is called the *product* of $X_1, ..., X_n$ (sometimes written as $X_1 \times ... \times X_n$), if for any other object Y with morphisms $f_i : Y \to X_i$, there must be a morphism $g : Y \to X$ such that $\pi_i \circ g = f_i$ for all i.

A category is said to have *finite products*, if for all $X_1, ..., X_n$, $X_1 \times ... \times X_n$ exists. We can compare functors with each other by looking at the following structure.

Definition 3.1.5. Let $\mathbf{C}$ and $\mathbf{D}$ be two categories, with F and G two functors from $\mathbf{C}$ to $\mathbf{D}$. A *natural transformation* η from F to G is a family of morphisms where for each $X \in ob(\mathbf{C})$ there is a morphism from $\mathbf{D}(F(X), G(X))$ denoted as η_X. These morphisms must satisfy the property that for every morphism $f : X \to Y$ in $\mathbf{C}$ we have $G(f) \circ \eta_X = \eta_Y \circ F(f)$.

For the purposes of realizability, we need to look at a special kind of category.

Definition 3.1.6. A *preorder enriched category* is a category C which has for all objects $a, b \in ob(C)$ a preorder on $\mathbf{C}(A, B)$. This preorder must be preserved under composition.

Examples of preorder enriched categories:
1) The category **Preord** of pre-ordered sets with order preserving maps between them.
2) The category **Heytpre** of Heyting pre-algebras, which has as morphisms the maps that after poset reflection (taking the same equivalence) form a Heyting morphisms. For all objects F, G we let $\mathbf{Heytpre}(F, G)$ be ordered pointwise.

Two objects or morphisms might act the same way, while not being the

same. For instance, a morphism $f : A \to B$ is called an *isomorphism* if there is a morphism $g : B \to A$ such that $f \circ g = 1_B$ and $g \circ f = 1_A$. If such a morphism exists, we call A and B *isomorphic*.
Two morphisms $f, g : A \to B$ are said to be *isomorphic* if there are isomorphisms $h : A \to A$ and $k : B \to B$ such that $f = k \circ g \circ h$.
Many structures in Category theory have a pseudo-form, where properties are true up to isomorphisms. We discuss two examples.

Definition 3.1.7. Let $\mathbf{C}$ and $\mathbf{D}$ be two preorder enriched categories. A *pseudo-functor* $F : \mathbf{C} \to \mathbf{D}$ maps each object $A \in ob(\mathbf{C})$ to an object $F(A) \in ob(\mathbf{D})$ and each morphism $f \in \mathbf{C}(A, B)$ to a morphism $F(f) \in \mathbf{D}(F(A), F(B))$ such that the identity morphisms and composition are preserved up to isomorphism.

Definition 3.1.8. Let $\mathbf{C}$ be a category. A *pseudo pull-back square* in $\mathbf{C}$ is a diagram which commutes up to isomorphism:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g"'] & Y \arrow[d, "h"] \\ Z \arrow[r, "k"] & W \end{tikzcd}$$

with the property that for every object A and every pair of maps $f_A : A \to Y$ and $g_A : A \to Z$ such that $h \circ f_A \sim k \circ g_A$, there is map $p : A \to X$ unique up to isomorphism satisfying $f \circ p \sim f_A$ and $g \circ p \sim g_A$.

3.2 Triposes

The construction of realizability toposes is generally done in two steps. The first step is the construction of what is called a tripos. It is short for 'topos-representing indexed pre-ordered set'. It is seen as the step in which one introduces a pre-order to the pca. In general, a tripos is defined as follows:

Definition 3.2.1. Let $\mathbf{C}$ be a category with finite products. A *tripos* $\mathbf{P}$ is a pseudofunctor from $\mathbf{P} : \mathbf{C}^{op} \to \mathbf{Heytpre}$ such that:
1) For every morphism f, we have a left and a right adjoint for $\mathbf{P}(f)$. We write them as $\exists_f$ and $\forall_f$.

2) (*Beck-Chevalley condition*) For each pull-back square with maps $h \circ f = k \circ g$ we have that $\forall_f \circ \mathbf{P}(g)$ is isomorphic to $\mathbf{P}(h) \circ \exists_k$.
3) For each object X there are objects $\pi(X) \in ob(\mathbf{C})$ and $\in_X \in \mathbf{P}(X \times \pi(X))$ (the *membership predicate*) such that: For all $Y \in ob(\mathbf{C})$ and $\phi \in \mathbf{P}(X \times Y)$ such that within $\mathbf{P}(X \times Y)$ (which is a Heyting pre-algebra), we have that $\in_X$ is isomorphic to $\mathbf{P}(1_X \times \{\phi\})$ for some morphism $\{\phi\} : Y \to \pi(X)$.

Now we look at how a pca can be used to construct a tripos. We use the natural structure of $\mathcal{P}(\mathcal{A})$-valued predicates defined before to construct a structure on this powerset.
Given a pca $\mathcal{A}$, we have seen that for any set X the set $\mathcal{P}(\mathcal{A})^X$ of $\mathcal{P}(\mathcal{A})$-valued predicates together with the relation $\leq$ forms a Heyting pre-algebra. From a function $f : X \to Y$ we can construct the map $f^* : \mathcal{P}(\mathcal{A})^Y \to \mathcal{P}(\mathcal{A})^X$ using composition with f. This preserves the Heyting pre-algebra structure on the sets. The map has both adjoints $\exists_f$ and $\forall_f$ which for ϕ a predicate, are given by:
$(\exists_f \phi)(y) = \{a \in \mathcal{A} : \exists x \in X, (f(x) = y \wedge a \in \phi(x))\} = \bigcup_{x, f(x)=y} \phi(x)$
$(\forall_f \phi)(y) = \{a \in \mathcal{A} : \forall b \in \mathcal{A}, \forall x \in X(f(x) = y \Rightarrow ab \downarrow \wedge ab \in \phi(x))\} = \bigcap_{f(x)=y} (\top \Rightarrow \phi)(x)$
We can conclude that this gives a tripos. We call it the realizability tripos of $\mathcal{A}$. We denote it by $\mathbf{T}(\mathcal{A})$.

3.3 Tripos to Topos

From $\mathcal{A}$ we have constructed the tripos $\mathbf{T}(\mathcal{A})$. From this tripos, we can further construct the realizability topos $RT(\mathcal{A})$. This can be seen as a sort f equivalence relation on the tripos.
To make notation a little bit clearer we will write $\leq_y^Y$ for the relation $\leq$ defined on the predicates on the set Y. The subscript describes which variables are used. So for instance for two predicates ϕ, ψ on X, $\phi(x) \leq_x^X \psi(x)$ just means $\forall x \in X, \phi(x) \leq \psi(x)$, or simply $\phi \leq \psi$.
Now we define $RT(\mathcal{A})$.

The *objects* of $RT(\mathcal{A})$ are pairs $(X, \sim_X)$ with X a set and $\sim_X$ a map $X \times X \to \mathcal{P}(\mathcal{A})$, $(x, y) \mapsto (x \sim_X y)$ satisfying:
1) $(x \sim_X y) \leq_{(x,y)}^{X \times X} (y \sim_X x)$

2) $(x \sim_X y \wedge y \sim_X z) \leq^{X\times X\times X}_{(x,y,z)} (x \sim_X z)$.

A *functional relation* between $(X, \sim_X)$ and $(Y, \sim_Y)$ is a predicate $F : X \times Y \to \mathcal{P}(\mathcal{A})$ such that:
1) $(F(x,y)) \leq^{X\times Y}_{(x,y)} (x \sim_X x \wedge y \sim_Y y)$
2) $(F(x,y) \wedge x \sim_X x' \wedge y \sim_Y y') \leq^{X\times X\times Y\times Y}_{(x,x',y,y')} (F(x',y'))$
3) $(F(x,y) \wedge F(x,y')) \leq^{X\times Y\times Y}_{(x,y,y')} (y \sim_Y y')$
4) $(x \sim_X x) \leq^X_x (\exists_y F(x,y))$

A *morphism* in $RT(\mathcal{A})$ between $(X, \sim_X)$ and $(Y, \sim_Y)$ is an equivalence class of functional relations between $(X, \sim_X)$ and $(Y, \sim_Y)$, where F and G are equivalent if $F(x,y) \leq^{X\times Y}_{(x,y)} G(x,y)$ and $G(x,y) \leq^{X\times Y}_{(x,y)} F(x,y)$.

This category is called a topos, since it satisfies the properties of a topos. The standard example of a realizability topos is $RT(\mathcal{K}_1)$, which is also called the *effective topos*. This structure is particularly interesting since it simulates Kleene's original model of realizability.

3.4 Geometric morphisms

Just like pca's can be related to each other using applicative morphisms, so can realizability toposes be related to each other using geometric morphisms. Each applicative morphism gives rise to a geometric morphism between the realizability components.

Lemma 3.4.1. *Every applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$ induces a functor of triposes $\gamma^* : \boldsymbol{T}(\mathcal{A}) \to \boldsymbol{T}(\mathcal{B})$ sending the predicate α to $\bigcup_{a\in\alpha} \gamma(a)$. This functor can be extended to a functor between the realizability toposes $RT(\mathcal{A}) \to RT(\mathcal{B})$*

Specifically, with $\Phi : \mathbf{T}(\mathcal{A}) \to \mathbf{T}(\mathcal{B})$ a functor between triposes constructed by pca's as above, we get an induced functor $\Phi' : RT(\mathcal{B}) \to RT(\mathcal{A})$ which sends $(X, \sim_X) \in ob(RT(\mathcal{B}))$ to $(X, \Phi^+_{X\times X}(\sim_X))$ in $RT(\mathcal{A})$ and $F : (X, \sim_X) \to (Y, \sim_y)$ to $\Phi^+_{X\times Y}(F) : (X, \Phi^+_{X\times X}(\sim_X)) \to (Y, \Phi^+_{Y\times Y}(\sim_Y))$.
There are more powerful functions one can find between realizability toposes. An example is a geometric morphism. They contain functorsboth ways with

additional properties and arise naturally from applicative morphisms in some cases.

Definition 3.4.2. For two toposes $\mathcal{E}$ and $\mathcal{F}$, a *geometric morphism* $f : \mathcal{E} \to \mathcal{F}$ consists of a pair of adjoint functors $f^* \dashv f_*$ with $f_* : \mathcal{E} \to \mathcal{F}$ the *direct image* and $f^* : \mathcal{F} \to \mathcal{E}$ the *inverse image*. Moreover, f^* is required to preserve limits.

Not always can an applicative morphism be used to construct a geometric morphism. In [4], a survey is given of possible cases. But the most general result is given in [7]:

Theorem 3.4.3. *An applicative morphism $\gamma : \mathcal{A} \to \mathcal{B}$ induces a geometric morphism $RT(\mathcal{B}) \to RT(\mathcal{A})$ if and only if it is computationally dense.*

Examples:
1) $\gamma : \mathcal{A} \to \mathcal{A}[f]$ from 2.3.5 induces a geometric morphism.
2) ι, δ, η and ζ from chapter 2.4 induce functors on the realizability toposes.

Chapter 4

Construction of models

To further our understanding of partial combinatory algebras and realizability toposes, we study the three fundamental pca's from before and look at new models in their vicinity. There are several ways to construct new models. One can use theorem 2.3.5 on a familiar pca, one can construct a pca from scratch or one can use the structure of a known pca to construct something else. We will see examples of these three methods. In section 4.1, we will study an extension of Scott's Graph Model. In section 4.2 extensions of the other two fundamental models are constructed using their relation to the Graph Model. Section 4.3 details the construction of a new model which forms a pca-topology pair with the Cantor topology. A general construction of a pca on the power set of another pca is given in 4.4. Lastly, we discuss some of these new models in the light of recursive sub-pca's.

4.1 Further studies on the Graph model

We have seen that the programs coded in Scott's Graph model $\mathbf{P}$ uses checks whether certain finite sets are included. When a set A is applied to a set B, we have that within A there lies the instructions to add elements to the result if certain finite sets are in B. This makes it so it has a natural relation with the Sierpinski product topology $\{\emptyset, \{1\}, \{0,1\}\}^{\mathbf{N}}$.
However, this application has a asymmetric feel to it. One cannot check whether certain finite sets are excluded. By introducing the complement function to the pca using theorem 2.3.5 we do get a pca that can do those

checks. This new pca $\mathbf{P}[C]$ is interesting in various ways. Firstly, unlike its predecessor it is decidable, which will be established in the first section. Secondly, the Cantor topology on $\mathcal{P}(\omega)$ is conrep to this new pca. In the second section, we will discuss this result among other topological properties of $\mathbf{P}$ and $\mathbf{P}[C]$.

4.1.1 The extension $\mathbf{P}[C]$

Let $C : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ be the complement function, sending A to $A^c := (\mathbf{N} - A)$. This map is not representable in $\mathbf{P}$. To see this, we use the fact that $\mathbf{P}$ forms a pca-topology pair with the Sierpinski product topology. Note that in that topology, for any finite non-empty set p, $\emptyset \notin U_p$. These U_p form a basis of the topology, so if $\emptyset \in U$ with U open, then $U = \mathcal{P}(\omega)$. Now take a finite non-empty p. We see that $\mathbf{N} \in U_p$, so $\emptyset \in C^{-1}(U_p)$. But $U_p \neq \emptyset$, so $C^{-1}(U_p) \neq \mathcal{P}(\omega)$. We can conclude that $C^{-1}(U_p)$ is not open, so C is not continuous in the Sierpinski product topology. Since all representable maps are continuous, C is not representable.

We can use theorem 2.3.5 to construct $\mathbf{P}[C]$. This new pca can be seen as the 'best' pca that represents both C and all functions representable in $\mathbf{P}$. Since C is not representable within $\mathbf{P}$, we must conclude that $\mathbf{P}[C]$ is not isomorphic to $\mathbf{P}$.
One of the properties that $\mathbf{P}$ lacked was decidability. Look at the map $D_{\mathbf{P}} : \mathcal{P}(\omega) \times \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ which sends $D(x, y)$ to $T_{\mathbf{P}}$ if $x = y$ and to $F_{\mathbf{P}}$ if not. If $\mathbf{P}$ were decidable, this map would be representable, yet it is not continuous so it cannot be. In the case of the extension however, things have changed.

Theorem 4.1.1. *$\boldsymbol{P}[C]$ is decidable.*

Before we go to the proof, we will look at what kinds of maps are representable within the new pca. Let $A \cdot B$ denote the application in $\mathcal{P}[C]$. The following things can be defined using the construction of $\mathcal{P}[C]$:
Let $A \circ B := \{m : \exists n(e_n \subset B, \langle n, m\rangle \in A)\}$. This is the application within the original pca $\mathbf{P}$. In $\mathcal{P}[C]$ it has a representation r such that $r \cdot A \cdot B = A \circ B$. Let c be such that $c \cdot A = C(A) = A^{-1}$ and denote T and F as the boolean true and false of this space satisfying for all a, b: $T \cdot a \cdot b = a$ and $F \cdot a \cdot b = b$. Lastly, we define the interesting operation $A * B := A \circ (c \cdot B) = \{m :$

$\exists n(e_n \cap B = \emptyset, \langle n, m\rangle \in A)\}$, which can be represented by $s = \langle xy\rangle r \cdot x \cdot (c \cdot y)$.

This last operation is interesting, since it specifically checks the absence of certain finite sets. We have seen what it means for a certain subset A of a pca to be decidable. It can sometimes be unwieldy to handle a proof of the decidability of certain sets, which is why the following lemma gives us a handy tool.

Lemma 4.1.2. *There is a representable map sending $\emptyset$ to F and $\{0\}$ to T.*

Proof: Define the following set: $N := \{\langle 1, \langle 2, x\rangle\rangle : x \in T\} \cup \{\langle 0, \langle 1, x\rangle\rangle : x \in F\}$. With this set, define the term $n := \langle x\rangle r \cdot (r \cdot N \cdot x) \cdot (c \cdot x) = \langle x\rangle s \cdot (r \cdot N \cdot x) \cdot x = \langle x\rangle (N \circ x) * x$.
Since $e_0 \subset \emptyset$, $e_1 = \{0\} \notin \emptyset$ and $e_2 \cap \emptyset = \emptyset$, we have that $n \cdot \emptyset = (N \circ \emptyset) * \emptyset = \{\langle 1, x\rangle : x \in F\} * \emptyset = F$.
Since $e_0 = \emptyset \subset \{0\}$, $e_1 = \{0\}$, $e_1 \cap \{0\} \neq \emptyset$ and $e_2 \cap \{0\} = \{1\} \cap \{0\} = \emptyset$, so $n \cdot \{0\} = (N \circ \{0\}) * \{0\} = (\{\langle 2, x\rangle : x \in T\} \cup \{\langle 1, x\rangle : x \in F\}) * \{0\} = (\{\langle 2, x\rangle : x \in T\} * \{0\}) \cup (\{\langle 1, x\rangle : x \in F\} * \{0\}) = T \cup \emptyset = T$. So n represents the desired map.

□

We will henceforth denote the representer of the map in this lemma by n. Note that a map which does the opposite (sending F to $\emptyset$ and T to $\{0\}$) is true simply by the definition of Booleans. Denote n' as the representer of this opposite map. So with this lemma, we can simplify the condition for a set to be decidable. For $A \subseteq \mathbf{P}[C]$, A is decidable if and only if the following map is representable:

$$x \mapsto \begin{cases} \{0\} & \text{if } x \in A \\ \emptyset & \text{otherwise} \end{cases}$$

We call this function the *zero-map* of A. We have already seen that finite intersections of decidable sets are also decidable. A representation of the decidability of the intersection of two decidable sets in terms of zero-maps is $v := \langle xy\rangle (nx(ny)\emptyset)$, for which $v\{0\}\{0\} = \{0\}$ and $v\emptyset\{0\} = v\{0\}\emptyset = v\emptyset\emptyset = \emptyset$. The map C gives us another property on decidability.

Lemma 4.1.3. *Let A be decidable. Then $C(A) = A^c$ is decidable.*

Proof: Let a represent the zero-map of A. Note that $\{\langle 1,0\rangle\} * \emptyset = \{0\}$ and $\{\langle 1,0\rangle\} * \{0\} = \emptyset$, so with $w := \langle xy\rangle r \cdot \{\langle 1,0\rangle\} \cdot (c \cdot (x \cdot y)) = \langle xy\rangle \{\langle 1,0\rangle\} * (x \cdot y)$ we get that $w \cdot a = \langle y\rangle \{\langle 1,0\rangle\} * (a \cdot x)$ represents the zero map of A^c. So A^c is decidable.

□

Take w as above. Now we have all tools to prove the main result:

Proof theorem 4.1.1: Note that $e_{2^x} = \{x\}$. Define $P := \{\langle 2^x, \langle 2^x, 0\rangle\rangle : x \in \mathbf{N}\}$. Then for any set A and B, we have $P \circ A = \{\langle 2^x, 0\rangle : x \in A\}$, so $P \circ A \circ B$ is $\emptyset$ if $B \cap A = \emptyset$, and $\{0\}$ otherwise. Hence with $s := \langle xy\rangle r \cdot (r \cdot P \cdot x) \cdot y = \langle xy\rangle P \circ x \circ y$, we have that $s \cdot A = \langle y\rangle P \circ A \circ y$ characterizes the set $\{B : B \cap A \neq \emptyset\}$
We can take the inverse using the previous lemma: with w as before, we find that $w \cdot (s \cdot A)$ characterizes $\{B : B \cap A \neq \emptyset\}^c = \{B : B \cap A = \emptyset\} = \mathcal{P}(A^c)$. So we have that $w \cdot (s \cdot (c \cdot A)) = w \cdot (s \cdot A^c)$ characterizes $\mathcal{P}(A)$.
We also have that $c \cdot w \cdot (s \cdot A) = C(w \cdot (s \cdot A))$ characterizes $C(\mathcal{P}(A^c)) = \{B : A \subset B\}$.
Using intersection (with v) we see that $\mathcal{P}(A) \cap C(\mathcal{P}(A^c)) = \{A\}$ is characterized by $v \cdot (w \cdot (s \cdot (c \cdot A))) \cdot (c \cdot w \cdot (s \cdot A))$. So if we define $d := \langle xy\rangle n \cdot (v \cdot (w \cdot (s \cdot (c \cdot x))) \cdot (c \cdot w \cdot (s \cdot x)) \cdot y)$ we get the following:
Take A and B two sets. Then $d \cdot A \cdot B = n \cdot (v \cdot (w \cdot (s \cdot (c \cdot A))) \cdot (c \cdot w \cdot (s \cdot A)) \cdot B) = n \cdot \chi_{\{A\}}(B)$, where $\chi_{\{A\}}$ denotes the deciding map for $\{A\}$. Meaning it is T if $B \in \{A\}$ and it is F if $B \notin \{A\}$.
We can conclude that d is a decider, and hence $\mathbf{P}[C]$ is decidable.

□

4.1.2 Topologies on $\mathbf{P}$ and $\mathbf{P}[C]$

Let us look at the topologies on these pca's. We have already seen that the Sierpinski product topology forms a pca-topology pair with $\mathbf{P}$. We can however, consider another topology we will call the *inclusion topology* on $\mathcal{P}(\omega)$. Here U is open if for all $V \in U$ and $W \subset V$ we have $W \in U$. Note that $\{\mathcal{P}(S) : S \subset \mathbf{N}\}$ gives a basis of this topology.

Lemma 4.1.4. *The inclusion topology on $\mathcal{P}(\omega)$ is a repcon topology for $\boldsymbol{P}$.*

Proof: Let $A \in \mathcal{P}(\omega)$. We consider the map $f : X \mapsto (A \cdot X)$. Take $U = \mathcal{P}(S)$ an element in the basis of the inclusion topology. We have that $f^{-1}(U) = \{B : A \cdot B \in U\} = \{B : A \cdot B \subset S\} = \{B : \{m : \exists n(e_n \subset B, \langle n, m\rangle \in A)\} \subset S\} = \{B : \forall n, (e_n \subset B \to (\forall m, \langle n, m\rangle \in A \to m \in S))\}$. For $V \in f^{-1}(U)$ and $W \subset V$. We have that for all n with $e_n \subset W \Rightarrow e_n \subset V$ so for those n we have $(\forall m, \langle n, m\rangle \in A \to m \in S)$. Hence $W \in f^{-1}(U)$. We can conclude that $f^{-1}(U)$ is open.

$\square$

Theorem 4.1.5. ***P** has precisely two minimal repcon topologies; the Sierpinski product topology and the inclusion topology.*

Proof: Let T be a repcon topology on **P** containing at least one open U which is not $\emptyset$ or $\mathcal{P}(\omega)$. We consider two separate cases:
Case 1: Assume $\emptyset \notin U$. Take a set $C \in U$, a finite set p and a natural number n such that $e_n = p$. Define the following set $A := \{\langle n, m\rangle : m \in C\}$ and consider the map $f : X \mapsto (A \cdot X)$. Since T is a repcon topology, f is continuous, hence $f^{-1}(U)$ is open. $f^{-1}(U) = \{B : A \cdot B \in U\} = \{B : \{m : \exists k, (e_k \subset B, \langle k, m\rangle \in A)\} \in U\} = \{B : \{m : p \subset B, m \in C\} \in U\}$. Since $\emptyset \notin U$ we get that $f^{-1}(U) = \{B : p \subset B\} = U_p$. Hence U_p is open in T. This is for all finite sets p. We can conclude that all opens from the Sierpinski product topology are open in T.
Case 2: Assume $\emptyset \in U$. Take $C \in (\mathcal{P}(\omega) - U)$ and $S \in \mathcal{P}(\omega)$. We are going to prove that $\mathcal{P}(S)$ is open in T. Define the set $A := \{\langle 2^n, m\rangle : n \notin S, m \in C\}$. Note that $e_{2^n} = \{n\}$. So for B a set, we have that if $B \in \mathcal{P}(S)$, then for all $n \notin S$ we have e_{2^n} is not a subset of B, so $A \cdot B = \emptyset$. If $B \notin \mathcal{P}(S)$, then there is an element $n \in B$ with $n \notin S$. This means $e_{2^n} \subset B$ and $A \cdot B = C$. We can conclude that $f^{-1}(U) = \mathcal{P}(S)$. So all opens in the inclusion topology are opens in T.
We can conclude that any repcon topology which is not indiscrete must either contain the Sierpinski product topology or the inclusion topology.

$\square$

Corollary 4.1.6. *The only repcon topologies for **P**$[C]$ are the trivial ones.*

Proof: Consider T a non-trivial topology repcon w.r.t. $\mathcal{P}(\omega)[C]$. Hence it has an open U which is not $\emptyset$ or $\mathcal{P}(\omega)$.

Now we have that since the complement function is representable, both U and U^c are open. So T contains an open with $\emptyset$ and one without it. So by theorem 4.1.5 and the fact that all representable functions of $\mathbf{P}$ are representable here, we have that T contains both the Sierpinski product topology and the Inclusion topology. So for any $S \in \mathcal{P}(\omega)$, both $\mathcal{P}(S)$ and $C(\mathcal{P}(\mathbf{N} - S))$ are open. Hence $\{S\} = \mathcal{P}(S) \cap C(\mathcal{P}(\mathbf{N} - S))$ is open. We must conclude that T is the discrete topology, leading to a contradiction.

□

Now we look at the Cantor topology $(\mathcal{P}(\omega))_{Cantor}$ with basic opens of the form U_p^q. Since not all maps continuous by the Sierpinski product topology are continuous by the Cantor topology, we see that the Cantor topology is not repcon to $\mathbf{P}$. Added to that, since C is continuous in this topology, we can already conclude that the Cantor topology is not conrep with $\mathbf{P}$. But the Cantor topology does have some relation with $\mathbf{P}[C]$.

Lemma 4.1.7. *The Cantor topology is a conrep topology for the pca* $\boldsymbol{P}[C]$

Proof: We use the same notation as in section 4.1.1. We denote $\cdot$ the operation in the pca $\mathcal{P}(\omega)[C]$ and $\circ$ the operation in the Graph model $\mathcal{P}(\omega)$. Let r be such that $r \cdot a \cdot b = a \circ b$ (by applicative morphism ι_C). Let c be such that $c \cdot a = a^c$ (by representability of C). Define $s := \langle xy \rangle (r \cdot x \cdot (c \cdot y)) = \langle xy \rangle (x \circ C(y))$.
Take $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ a continuous partial map. Since it is continuous, we have that for all n, $\{A \in dom(F) : n \in F(A)\}$ is open in $dom(f)$, hence there is an open set V_1^n containing that set but not intersecting any elements in $\{A \in dom(F) : n \notin F(A)\}$. So we can write V_1^n as $\bigcup_{i \in I_n} U_{p_i^n}^{q_i^n}$ with I_n an index set and for each $i \in I_n$ finite sets p_i^n and q_i^n. Define the following set, $W_1 := \{\langle a, \langle b, k \rangle \rangle : k \in \mathbf{N}, \exists i, j \in I_k (p_i^k = e_a, q_i^k = e_b)\}$. With this, we have that $z := \langle x \rangle s \cdot (r \cdot W_1 \cdot x) \cdot x$ represents F. Let us check this last fact:
Take $A \in \mathrm{dom}(F)$, then $z \cdot A = s \cdot (r \cdot W_1 \cdot A) \cdot A = s \cdot (W_1 \circ A) \cdot A$. Now, $W_1 \circ A = \{m : \exists n (e_n \subset A, \langle n, m \rangle \in W_1)\} = \{\langle b, k \rangle : \exists a (e_a \subset A, \exists i, j \in I_k (p_i^k = e_a, q_i^k = e_b))\} = \{\langle b, k \rangle : \exists i, j \in I_k (p_i^k \subset A, q_i^k = e_b)\}$.
So $s \cdot (W_1 \circ A) \cdot A = \{m : \exists n (A \cap e_n = \emptyset, \langle n, m \rangle \in (W_1 \circ A))\} = \{k : \exists b (A \cap e_b = \emptyset, \exists i, j \in I_k (p_i^k \subset A, q_i^k = e_b))\} = \{k : \exists i, j \in I_k (p_i^k \subset A, A \cap q_i^k = \emptyset))\} = \{k : A \in W_1^k\} = \{k : k \in F(A)\} = F(A)$. So we have that for $A \in dom(f)$, $z \cdot A = F(A)$.

We can conclude that any continuous partial map $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ is representable in $\mathcal{A}[C]$.

$\square$

The Cantor topology does not form a pca-topology pair with $\mathbf{P}[C]$, which is a result by corollary 4.1.6. So we can ask ourselves, is their a pca with which the Cantor topology does form such a pair? We will discuss this in section 4.3. For the section, we will study how the extension of $\mathbf{P}$ into $\mathbf{P}[C]$ translates to other two related pca's.

4.2 System of extensions

In this chapter, we will use the results in Bauer's thesis [1] discussed in section 2.4, to compare the extension $\mathbf{P}[C]$ of the Scott Graph model with extensions of both Kleene's second model and the universal domain model. These will form a system of extensions consisting of pull-back squares.

4.2.1 An extension of $\mathcal{K}_2$

By theorem 2.3.5 we have a decidable single-valued applicative morphism $\gamma_p : \mathbf{P} \to \mathbf{P}[C]$, given by $a \mapsto \{a\}$. We will now study an extension of $\mathcal{K}_2$, which is related to $\mathbf{P}[C]$.
We define the map $S : \mathcal{K}_2 \to \mathcal{K}_2$ which sends α to the sequence:

$$S(\alpha)(n) = \begin{cases} n+1 & \text{if } \forall k, \alpha(k) \neq n+1 \\ 0 & \text{otherwise} \end{cases}$$

Remember that $\mathcal{K}_2$ forms a pca-topology pair with the traditional topology on $\mathbf{N}^{\mathbf{N}}$. S is not continuous in this topology, since for instance $S^{-1}(U_{\langle 1 \rangle})$ with $U_{\langle 1 \rangle} = \{\alpha : \alpha(0) = 1\}$ is not open. So it does not have a representation. Hence, extending $\mathcal{K}_2$ with S yields new pca $\mathcal{K}_2[S]$ not isomorphic to the old one. There is a single-valued decidable applicative morphism $\gamma_k : \mathcal{K}_2 \to \mathcal{K}_2[S]$ given by $\alpha \mapsto \{\alpha\}$.

Remember the applicative retraction $\iota \dashv \delta$ between $\mathcal{K}_2$ and $\mathbf{P}$. By concatenation, we get a decidable applicative morphism $\gamma_k \circ \delta : \mathbf{P} \to \mathcal{K}_2[S]$ given by $A \mapsto \delta(A)$. We have the following result.

Lemma 4.2.1. *C is representable with respect to $\gamma_k \circ \delta$.*

Proof: Take a set $A \in \mathbf{P}$ and a sequence $\alpha \in \delta(A)$. We have that $S(\alpha)(n) \neq 0$ is true precisely when $\forall k, \alpha(k) \neq n+1$, which in turn is true precisely when $n \notin A$ (by definition of δ). So $S(\alpha)(n) \neq 0 \Leftrightarrow \forall k, \alpha(k) \neq n+1 \Leftrightarrow n \notin A \Leftrightarrow n \in A^c$. Hence $\exists k, S(\alpha)(k) = n+1$ if and only if $n \in A^c$. We can conclude that $S(\alpha) \in \delta(A^c) = \gamma_k \circ \delta(A^c)$ which means that C is representable with respect to $\gamma_k \circ \delta$.

□

γ_p has been created using theorem 2.3.5 which has some extra properties to be exploited. We can use the result of the theorem together with the fact stated in the last lemma to conclude that there is a decidable applicative morphism $\delta' : \mathbf{P}[C] \to \mathcal{K}_2[S]$ such that $\delta' \circ \gamma_p = \gamma_k \circ \delta$. We can easily see that δ' acts the same way as δ on the underlying sets.
Now look at the single-valued decidable applicative morphism $\gamma_p \circ \iota : \mathcal{K}_2 \to \mathbf{P}[C]$ sending α to $\iota(\alpha)$. We have a similar result.

Lemma 4.2.2. *S is representable w.r.t. $\gamma_p \circ \iota$.*

Proof: For all n we define:
$M_n := \{A : \exists a \in \mathbf{N}^*, (\exists i < length(a), a(i) = n+1) \wedge seq(a) \in A\}$. Here, $a(i)$ stands for the i-th component of the finite sequence a. M_n is a union of the opens $U_{\{seq(a)\}}$ (where the finite sequences a range over those that somewhere contain $n+1$). So it is open in the Sierpinski product topology. Now we have for $\alpha \in \mathcal{K}_2$, $\iota(\alpha) \in C(M_n) \Leftrightarrow \forall k, \alpha(k) \neq n+1$.
For $a \in \mathbf{N}^*$ define $L_a := \{A : \forall b \in \mathbf{N}^*, b \sqsubseteq a \Rightarrow seq(b) \in A\}$ which is an open since it is also a union of opens of the form $U_{\{seq(b)\}}$. It is the collection of sets which contains all encoded initial segments of a. Let $N_n := \bigcup_{a \in M_n} L_a$, which is again an open set in the Sierpinski product topology. This set is the collection of sets which contains at least one encoded finite sequence with $n+1$ somewhere, together with all its encoded initial sequences. So we have that for $\alpha \in \mathcal{K}_2$, $\iota(\alpha) \in N_n \Leftrightarrow \exists k, \alpha(k) = n+1$.
Let Q denote the set of elements $x \in \mathbf{N}$ for which there is a $b \in \mathbf{N}^*$ with $seq(b) = x$. If such a b exists, it is unique. Let $x \in Q$ and let b be the unique finite sequence such that $seq(b) = x$. Then $x \in \iota(S(\alpha)) \Leftrightarrow b \sqsubseteq S(\alpha) \Leftrightarrow \forall k < length(b) : ((b(k) = 0 \wedge \exists n, \alpha(n) = k+1) \vee (b(k) = k+1 \wedge \forall n, \alpha(n) \neq k+1)) \Leftrightarrow$

$\forall k < length(b) : ((b(k) = 0 \wedge \iota(\alpha) \in N_k) \vee (b(k) = k+1 \wedge \iota(\alpha) \in C(M_k)).$
So there are opens V_x and W_x in the Sierpinski product topology such that $x \in \iota(S(\alpha)) \Leftrightarrow \iota(\alpha) \in V_x \cap C(W_x)$. We define the following map:
$g : \mathbf{P}[C] \to \mathbf{P}[C]$ given by $A \mapsto \{2x : x \in Q, A \in V_x\} \cup \{2x+1 : x \in Q, A \in W_x\}$. This map is continuous in the Sierpinski product topology, so it has a representation in $\mathbf{P}$ and hence also in $\mathbf{P}[C]$.
Also define the map $h : \mathbf{P}[C] \to \mathbf{P}[C]$ defined by $A \mapsto \{x : A \in U^{\{2x+1\}}_{\{2x\}}\}$. This map is continuous in the Cantor topology, so it has a representation in $\mathbf{P}[C]$. We can conclude that $h \circ g$ is representable in $\mathbf{P}[C]$. By construction, we have that if $A \in \iota(\alpha)$ then $x \in h \circ g(A)$ if and only if $(2x) \in g(A)$ and $(2x+1) \notin g(A)$, which is precisely when $x \in Q$, $A \in V_x$ and $A \in C(W_x)$, meaning x is in the O with $\{O\} = \iota(S(\alpha))$. If $x \in O$, then $x \in h \circ g(A)$ by construction. So the representable map $h \circ g$ represents S with respect to $\gamma_p \circ \iota$.

□

We have found that $\gamma_p \circ \iota : \mathcal{K}_2 \to \mathbf{P}[C]$ is a single-valued decidable applicative morphism which represents S. So by theorem 2.3.5, there is a decidable applicative morphism $\iota' : \mathcal{K}_2[S] \to \mathbf{P}[C]$ such that $\iota' \circ \gamma_k = \gamma_p \circ \iota$. Note that ι' works the same way as ι on the underlying sets. In particular, we know it to be single-valued.

Remarks:
(1) We again have a retraction $\iota' \dashv \delta'$ since we can use the three representing elements for $\iota \dashv \delta$ and transpose them with γ_k and γ_p.
(2) Since ι' is discrete and single-valued (injective) and δ' is decidable, the decidability of $\mathbf{P}[C]$ translates to $\mathcal{K}_2[S]$. This can also be concluded by observing that $(\alpha, \beta) \mapsto |\alpha - \beta|$ is continuous and we can check whether a sequence is always zero.
(3) S effectively checks for all values $n > 0$ whether or not a function of natural numbers has n in its image. Since $\alpha \mapsto \alpha + 1$ and $\alpha \mapsto max(0, 1-\alpha)$ are both continuous, we have that the following function is realizable:

$$S'(\alpha)(n) = \begin{cases} 1 & \text{if } \exists k, \alpha(k) = n \\ 0 & \text{otherwise} \end{cases}$$

This function is probably a more desirable substitute for our more mechanically constructed S-function. Obviously, $\mathcal{K}_2[S] \simeq \mathcal{K}_2[S']$.

(4) $\mathcal{K}_2[S]$ does not check how much times certain numbers occur in a sequence. So further extensions could include checking which values occur at least two times.

A function that is more frequently studied is $E : \mathbf{N}^{\mathbf{N}} \to \mathbf{N}$ defined by $E(\alpha) = 0$ if $\exists k, \alpha(k) = 0$ else $E(\alpha) = 1$. We can see that if this operation can be effectively represented within the space, it could be possible to effectively represent S.

4.2.2 An extension of U

For the universal domain model, we can find a similar extension. Remember the applicative inclusion $\eta \dashv \zeta$ and the sets B_n and C_n used to construct the applicative morphisms. Consider the following partial map $Z : \mathbf{U} \rightharpoonup \mathbf{U}$ which has as its domain the Cantor-clopen sets and sends these to their complement. So it is the function $V \mapsto V^c$ which is defined everywhere it can be defined. Consider the Scott-open set $\uparrow U$ with U a non-empty clopen. Let $A \subset U$ be a clopen such that $A \neq U$. It is not difficult to see that such a clopen exists. We have that $Z^{-1}(\uparrow U) \subseteq Z^{-1}(A)$ but $Z^{-1}(\uparrow U) \neq Z^{-1}(A)$. So it is not upwards closed, hence not open in the Scott topology.

So we have that $\mathbf{U}[Z]$ is not isomorphic to $\mathbf{U}$. We can consider $\gamma_u : \mathbf{U} \to \mathbf{U}[Z]$. We have the following results:

Lemma 4.2.3. *C is representable w.r.t. $\gamma_u \circ \zeta$ and S is representable w.r.t. $\gamma_p \circ \eta$.*

Proof: Let $A \subset N$. Then $\zeta(A) = \bigcup_{n \in A} C_n$ and $\zeta(C(A)) = \bigcup_{n \in C(A)} C_n = \bigcup_{n \notin A} C_n$. Now since the C_n's form a partition, we can see that $\zeta(A)$ is a clopen and $Z(\zeta(A)) = \zeta(C(A))$, which is what we wanted.

For the second statement, take S a clopen. $\eta S = \{n : B_n \subset S\}$ and $\eta(Z(S)) = \{n : B_n \subset Z(S)\} = \{n : B_n \cap S \neq \emptyset\}$. Now note the following. If for some n, $B_n \cap S \neq \emptyset$, then $B_n \cap S$ is a clopen. So there is an m such that $B_m = B_n \cap S$. Define the following set $A := \{\langle 2^m, n\rangle : B_m \subset B_n\}$, then we have with $*$ the application in $\mathbf{P}$ represented in $\mathbf{P}[C]$ that:
$B_n \cap S \neq \emptyset \Leftrightarrow \exists m, B_m \subset B_n \wedge B_m \subset S \Leftrightarrow \exists m, m \in \eta S \wedge \langle 2^m, n\rangle \in A$. So $A * (\eta S) = (\eta Z(S))^c$. Hence Z is indeed representable w.r.t. η.

$\square$

So there is a decidable applicative morphism $\zeta' : \mathbf{P}[C] \to \mathbf{U}[Z]$ such that $\zeta' \circ \gamma_p = \gamma_u \circ \zeta$ and there is a decidable applicative morphism $\eta' : \mathbf{U}[Z] \to \mathbf{P}[C]$ such that $\eta' \circ \gamma_u = \gamma_p \circ \eta$. This gives us the full system of extensions:

$$\begin{array}{ccccc} \mathcal{K}_2 & \leftrightarrows & \mathbf{P} & \leftrightarrows & \mathbf{U} \\ \downarrow & & \downarrow & & \downarrow \\ \mathcal{K}_2[S] & \leftrightarrows & \mathbf{P}[C] & \leftrightarrows & \mathbf{U}[Z] \end{array}$$

By translating the representers of the applicative inclusion $\eta \dashv \zeta$ via the gamma functions, we can see that we have another applicative inclusion $\eta' \dashv \zeta'$.

4.2.3 Pull-back squares

The diagrams that were found in the first two parts of this section have an interesting property.

Theorem 4.2.4. *The commutative diagram*

$$\begin{array}{ccc} \mathcal{K}_2 & \xrightarrow{\iota} & \boldsymbol{P} \\ \gamma_k \downarrow & & \downarrow \gamma_p \\ \mathcal{K}_2[S] & \xrightarrow{\iota'} & \boldsymbol{P}[C] \end{array}$$

is a pull-back square and a pseudo-pullback square.

Proof: Let $\mathcal{A}$ be a pca which has both an applicative morphism to $\mathbf{P}$ and to $\mathcal{K}_2[S]$. We call those morphisms α_p and α_k respectively. We assume $\gamma_p \circ \alpha_p \sim \iota' \circ \alpha_k$.
We define the following morphism $\alpha : \mathcal{A} \to \mathcal{K}_2$. Take n to be a representation of $\delta \circ \iota \prec \mathbf{1}_{\mathcal{K}_2}$. For $x \in \mathcal{A}$, define $\alpha(x) := \{ny : \exists z \in \alpha_p(x), y \in \delta(z)\}$. So $\iota \circ \alpha(x) = \alpha_p(x) \cap im(\iota)$.
We first prove that α is an applicative morphism. Take $x, y \in \mathcal{A}$ such that $xy \downarrow$. Since $\delta \circ \alpha_p$ is applicative, there is an element $a \in \mathcal{K}_2$ such that $\forall x' \in \delta \circ \alpha_p(x)$ and $\forall y' \in \delta \circ \alpha_p(y)$ we have $ax'y' \in \delta \circ \alpha_p(xy)$. So $n(ax'y') \in \alpha(xy)$. We can effectively send any $a \in \alpha(x)$ to an $x' \in \delta \circ \alpha_p(x)$ using $\mathbf{1}_{\mathcal{K}_2} \prec \delta \circ \iota$. We can conclude that α is applicative.

Note that $\iota' \circ \gamma_k \circ \alpha = \gamma_p \circ \iota \circ \alpha = \gamma_p \circ \alpha_p \cap im(\iota')$. Using representations of ι', $\gamma_p \circ \alpha_p \sim \iota' \circ \alpha_k$, δ' and $\delta' \circ \iota' \sim \mathbf{1}_{\mathcal{K}_2[S]}$ (in that order) we can prove that $\gamma_k \circ \alpha \sim \alpha_k$. Hence we can conclude that it is a pseudo-pullback square.
You can easily see that it is also a pullback square. Just use the proof above with the new assumption $\gamma_p \circ \alpha_p = \iota' \circ \gamma_k$. The defined α is still an applicative morphism, and the definition of α immediately implies $\gamma_k \circ \alpha = \alpha_k$. So it is a pullback square.

$\square$

We only used that $\delta \circ \iota \sim \mathbf{1}_{\mathcal{K}_2}$ and $\delta' \circ \iota' \sim \mathbf{1}_{\mathcal{K}_2[S]}$. Hence, since $\eta \circ \zeta \sim \mathbf{1}_{\mathbf{P}}$ and $\eta' \circ \zeta' \sim \mathbf{1}_{\mathbf{P}[C]}$ we can also conclude in the same way that:

Corollary 4.2.5. *The commutative diagram*

$$\begin{array}{ccc} \boldsymbol{P} & \xrightarrow{\zeta} & \boldsymbol{U} \\ \gamma_u \downarrow & & \downarrow \gamma_p \\ \boldsymbol{P}[C] & \xrightarrow{\zeta'} & \boldsymbol{U}[Z] \end{array}$$

is a pull-back square and a pseudo-pullback square.

4.3 Double Graph model

In section 4.1.2, we have established that neither $\mathbf{P}$ nor $\mathbf{P}[C]$ forms a pca-topology pair with the Cantor topology. But there does exist a pca on $\mathcal{P}(\omega)$ all representable maps are Cantor continuous, and all continuous maps are representable. In this chapter we will construct such a pca, named the Double Graph Model.

4.3.1 Definition

We define the following coding that is going to be used in the construction of a pas. Take $\langle .,. \rangle$ the usual pairing on the natural numbers. Then define for $a, b, i, x \in \mathbf{N}$ the following functions from $\mathbf{N}^4$ to $\mathbf{N}$: $\langle a, b, i, x \rangle_1 := 2 * \langle \langle a, b \rangle, \langle i, x \rangle \rangle$ and $\langle a, b, i, x \rangle_2 := 2 * \langle \langle a, b \rangle, \langle i, x \rangle \rangle + 1$. Note that by construction, both functions are injective and their range is different. So each input and choice of function gives us a unique value. We can use these functions to code four natural numbers and one binary value into one natural number. Now we construct two fundamental relations. For each $x \in \mathbf{N}$, define two relations on $\mathcal{P}(\omega)^2$ by:

$$I_x(A,B) := \exists p,q,i : (B \subset U_{e_p}^{e_q} \ \& \ \langle p,q,i,x\rangle_1 \in A)$$

$$II_x(A,B) := \exists p,q,i : (B \subset U_{e_p}^{e_q} \ \& \ \langle p,q,i,x\rangle_2 \in A)$$

The similarities with Scott's Graph model should become apparent. The relations gives us a way to code within A a way to check whether B is within certain opens from the basis of the Cantor topology. We now define the following partial application map $(.\cdot.) : \mathcal{P}(\omega) \times \mathcal{P}(\omega) \rightharpoonup \mathcal{P}(\omega)$. For A and B two sets in $\mathcal{P}(\omega)$, we say:

$$(A \cdot B \downarrow) \Leftrightarrow \forall x \in \mathbf{N} : (I_x(A,B) \vee II_x(A,B)) \wedge \neg(I_x(A,B) \wedge II_x(A,B))$$

So we have that for all x, one of the two relations must be true but not both at the same time. We finish of the definition of the map by defining for A and B,

$$(A \cdot B \downarrow) \Rightarrow A \cdot B = \{x \in \mathbf{N} : I_x(A,B)\}$$

Note that by the definition of the domain of the application map, we also have that $A \cdot B \downarrow$ implies $A \cdot B = \{x \in \mathbf{N} : II_x(A,B)\}^c$. So if the application is defined, we can check both $x \in A \cdot B$ and its negation by looking if B is in certain Cantor opens.

To see the connection between this application map and the Cantor topology, consider the following. Take a partial map $F : \mathcal{P}(\omega) \rightharpoonup \mathcal{P}(\omega)$ which is continuous in the Cantor topology. By the definition of the topology, we have that for all x, the sets $V_x^1 := \{A \in dom(F) : x \in F(A)\}$ and $V_x^2 := \{A \in dom(F) : x \notin F(A)\} = (dom(F) - V_x^1)$ must both be open in $dom(F)$. Hence there are opens W_x^1 and W_x^2 in the Cantor topology such that $V_x^1 = W_x^1 \cap dom(F)$ and $V_x^2 = W_x^2 \cap dom(F)$. With these W's, we can define an extension F' of F, where:

$$dom(F') := \bigcap_{x \in \mathbf{N}} ((W_x^1 \cup W_x^2) - (W_x^1 \cap W_x^2))$$

And for all $A \in dom(F')$ we have:

$$F'(A) := \{x \in \mathbf{N} : A \in (W_x^1 - W_x^2)\} = \{x \in \mathbf{N} : A \notin (W_x^2 - W_x^1)\}$$

Note that the domain of F' is bigger than the domain of F since for all x, $dom(F) = V_x^1 \cup V_x^2 \subset ((W_x^1 \cup W_x^2) - (W_x^1 \cap W_x^2))$. Also, for $A \in dom(F)$, $F(A) = \{x : A \in V_1^x\} \subset \{x : A \in (W_x^1 - W_x^2)\} = F'(A) = \{x : A \notin (W_x^2 - W_x^1)\} \subset \{x : x \notin V_x^2\} = F(A)$ since $V_x^1 \subset (W_x^1 - W_x^2)$ and $V_x^2 \subset (W_x^2 - W_x^1)$. So $F(A) = F'(A)$, hence F' is indeed an extension.
F' is defined only using the collection of opens W_x^1 and W_x^2 for all x. We will call a map defined this way a *coded-continuous map*. We say that the collection of opens W_x^1 and W_x^2 gives us the *coding* of this map. It should be apparent that any endomap representable by our pas is a coded-continuous map, where if A is representer the coding opens are given by $W_x^1 := \{B : I_x(A, B)\}$ and $W_x^2 := \{B : II_x(A, B)\}$. So the Cantor topology is repcon to our pas. Below, we are going to prove a lemma which gives us that the topology is also conrep.

4.3.2 Combinatory completeness of DP

Lemma: For any coded-continuous map F, there is an element $A \in \mathcal{P}(\omega)$ such that for all $B \in \mathcal{P}(\omega)$: $B \in dom(F) \Leftrightarrow (A \cdot B)\downarrow \Leftrightarrow F(B) = (A \cdot B)$.

Proof: Note that any open in the Cantor topology can be written as a countable union of opens of the form $U_p^q = \{A : p \in A, q \cap A = \emptyset\}$, with p and q finite. For p finite, denote $f(p)$ as the unique number such that $e_{f(p)} = p$ (Where e_n is the n-th finite set). Take for all x the opens W_x^1 and W_x^2 as the sets that code F. Since all W-s are open, we can write them as follows: for all x we have a choice of sets I_x, J_x, and for all $i \in I_x$ and $j \in J_x$ a choice of finite sets p_i^x, q_i^x, k_j^x and r_j^x such that $W_x^1 = \bigcup_{i \in I_x} U_{p_i^x}^{q_i^x}$ and $W_x^2 = \bigcup_{j \in J_x} U_{k_j^x}^{r_j^x}$. With those, define $A \in \mathcal{P}(\omega)$ as:

$$A := \{\langle f(p_i^x), f(q_i^x), i, x\rangle_1 : x, i \in I_x\} \cup \{\langle f(k_j^x), f(r_j^x), j, x\rangle_1 : x, j \in J_x\}$$

With this set, it should be apparent from the definition of the application map that $(A \cdot B)\downarrow$ if and only if for all x, $B \in ((W_x^1 \cup W_x^2) - (W_x^1 \cap W_x^2))$ and that if that is the case, $(A \cdot B) = \{x : B \in (W_x^1 - W_x^2)\} = \{x : B \notin (W_x^2 - W_x^1)\}$. So this A represents F in the sense of the lemma.

$\square$

We can conclude that the Cantor topology is conrep w.r.t. our pas. We now just want to see if the pas is also a pca. We need to look at multi-valued maps. Consider a partial continuous map $F : \mathcal{P}(\omega)^n \to \mathcal{P}(\omega)$. The same way as before, we can say that they are *coded-continuous* if they are coded using opens W_x^1 and W_x^2 in $\mathcal{P}(\omega)^n$ for all x. Knowing the opens, the construction of the map is the same as before. An open in $\mathcal{P}(\omega)^n$ can be written as a countable union of opens of the form $U_{a_1}^{b_1} \times ... \times U_{a_n}^{b_n}$. Meaning you can code the opens within the sets. We will prove that maps with two and three variables are representable.

Lemma 4.3.1. *For any coded-continuous map $F : \mathcal{P}(\omega)^2 \to \mathcal{P}(\omega)$ there is an R such that for all A_1, A_2 we have:*
(I) $R \cdot A_1 \downarrow$
(II) $F(A_1, A_2) \downarrow \Rightarrow F(A_1, A_2) = R \cdot A_1 \cdot A_2$
(III) $R \cdot A_1 \cdot A_2 \downarrow \Rightarrow R \cdot A_1 \cdot A_2 = F(A_1, A_2)$

Proof: For two natural numbers a and b, define $U_a^b := U_{e_a}^{e_b}$. Let for all x, W_x^1 and W_x^2 be the opens that code the map. So $W_x^1 = \bigcup_{i \in I_x} U_{a_i^x}^{b_i^x} \times U_{p_i^x}^{q_i^x}$ and $W_x^2 = \bigcup_{j \in J_x} U_{c_j^x}^{d_j^x} \times U_{k_j^x}^{r_j^x}$ for some choice of sets I_x, J_x and natural numbers $a_i^x, b_i^x, p_i^x, q_i^x, c_j^x, d_j^x, k_j^x$ and r_j^x. We define the representer as follows, starting with a core set:
$K_0 := \{\langle a_i^x, b_i^x, i, \langle p_i^x, q_i^x, i, x\rangle_1\rangle_1 : x \in \mathbf{N}, i \in I_x\}$
$\cup\{\langle c_j^x, d_j^x, j, \langle k_j^x, r_j^x, j, x\rangle_2\rangle_1 : x \in \mathbf{N}, j \in J_x\}$.
We need to add elements to the core such that the first application always yields a result. Note that K_0 consists of elements of the form $\langle a, b, i, y\rangle_1$, and for such an y, there are no c, d, j with $(c, d, j) \neq (a, b, i)$ such that $\langle c, d, j, y\rangle_1 \in K_0$. This implies the command to add y to $K_0 \cdot B$ if $B \in U_{e_a}^{e_b}$. But if $B \notin U_{e_a}^{e_b}$, $K_0 \cdot B \uparrow$. We need to add elements to K_0 to create a new set K_1 such that $II_y(K_1, B)$ is true if and only if B is not in $U_{e_a}^{e_b}$.
Let K_1 be the smallest set containing all elements of K_0 such that for all $\langle a, b, i, y\rangle_1 \in K_0$ we have:
(1) For all $x \in e_a$, $\langle 0, 2^x, i, y\rangle_2 \in K_1$.
(2) For all $x \in e_b$, $\langle 2^x, 0, i, y\rangle_2 \in K_1$.
Note that with this K_1 and such an y, that for all B either but not both $I_y(K_1, B)$ and $II_y(K_1, B)$ are true. Also note $I_y(K_1, B)$ is true if and only if $B \in U_{e_a}^{e_b}$.
We still have unmentioned y-s, for which we neither have $I_y(K_1, B)$ nor $II_y(K_1, B)$ is true for any B. For those we want to add $\langle 0, 0, 0, y\rangle_2$ which

basically asserts $II_y(K_1, B)$ for all B. So let R be as follows:
$R := K_1 \cup \{\langle 0,0,0,y\rangle_2 : \forall a,b,i,(\langle a,b,i,y\rangle_1 \notin K_1 \wedge \langle a,b,i,y\rangle_2 \notin K)\}$
Now for this R, we have that for all y and B, either but not both $I_y(R, B)$ or $II_y(R, B)$ is true. So $R \cdot A_1 \downarrow$ is defined for all A_1. By looking at the definition of K_1, we see that (II) is also true. So R represents F.

$\square$

Note that the map $K : \mathcal{P}(\omega)^2 \to \mathcal{P}(\omega)$ sending (A_1, A_2) to A_1 is a total continuous map. Hence it is coded-continuous. So it can be represented.

Lemma 4.3.2. *For any coded-continuous map $F : \mathcal{P}(\omega)^3 \to \mathcal{P}(\omega)$ there is an R such that for all A_1, A_2, A_3 we have:*
(I) $R \cdot A_1 \cdot A_2 \downarrow$
(II) $F(A_1, A_2, A_3) \downarrow \Rightarrow R \cdot A_1 \cdot A_2 \cdot A_3 = F(A_1, A_2, A_3)$

Proof: Let for all x, W_x^1 and W_x^2 be the defining opens. So $W_x^1 = \bigcup_{i \in I_x} U_{a_i^x}^{b_i^x} \times U_{p_i^x}^{q_i^x} \times U_{v_i^x}^{w_i^x}$ and $W_x^2 = \bigcup_{j \in J_x} U_{c_j^x}^{d_j^x} \times U_{k_j^x}^{r_j^x} \times U_{h_j^x}^{l_j^x}$. We define the representer as follows, starting with its core:
$K_0 := \{\langle a_i^x, b_i^x, i, \langle p_i^x, q_i^x, i, \langle v_i^x, w_i^x, i, x\rangle_1\rangle_1\rangle_1 : x \in \mathbf{N}, i \in I_x\} \cup$
$\{\langle c_j^x, d_j^x, j, \langle k_j^x, r_j^x, j, \langle h_j^x, l_j^x, j, x\rangle_2\rangle_1\rangle_1 : x \in \mathbf{N}, j \in J_x\}$.
The core now contains all the data of the map F. Like before we want to add elements to make (I) true. We work backwards. Note that K_0 consists of elements of the form $\langle a, b, i, \langle c, d, i, y\rangle_1\rangle_1$, and there is at most one such element for each y. Let K_1 be the smallest set containing K_0 such that for all $\langle a, b, i, \langle c, d, i, y\rangle_1\rangle_1 \in K_0$:
(1) For all $x \in e_c$, $\langle a, b, i, \langle 0, 2^x, i, y\rangle_2\rangle_1 \in K_1$.
(2) For all $x \in e_d$, $\langle a, b, i, \langle 2^x, 0, i, y\rangle_2\rangle_1 \in K_1$.
This implies that in the second application step '$K_1 \cdot A_1 \cdot A_2$', that if $A_1 \in U_a^b$ and $K_1 \cdot A_1 \downarrow$ we have that either but not both $I_y(K_1 \cdot A_1, A_2)$ and $II_y(K_1 \cdot A_1, A_2)$ are true, and $A_2 \in U_c^d \Leftrightarrow I_y(K_1 \cdot A_1, A_2)$. Now, such a K_1 consists of elements of the form $\langle a, b, i, z\rangle_1$ which are unique for each z. Each of these elements z is equal to some $z = \langle c, d, i, y\rangle_v$ with $v \in \{1, 2\}$. So we can define the following set: Let K_2 be the smallest set containing K_1 such that for all $\langle a, b, i, z\rangle_1 \in K_1$, $z = \langle c, d, i, y\rangle_v$:
(1) $\langle a, b, i, \langle 0, 0, i, y\rangle_2\rangle_2 \in K_2$.
(2) For all $x \in e_a$, $\langle 0, 2^x, i, z\rangle_2 \in K_2$ and $\langle 0, 2^x, i, \langle 0, 0, i, y\rangle_2\rangle_1 \in K_2$.
(3) For all $x \in e_b$, $\langle 2^x, 0, i, z\rangle_2 \in K_2$ and $\langle 2^x, 0, i, \langle 0, 0, i, y\rangle_2\rangle_1 \in K_2$.

This looks a bit different than before since we have to make sure that such an y always gets mentioned in the second step, making either $I_y(K_1 \cdot A_1, A_2)$ or $II_y(K_1 \cdot A_1, A_2)$ true if $K \cdot A_1 \downarrow$. Now all mentioned elements in the first and second step have a 'complete domain' (Either but not both I_y or II_y is true for all y). We have that if (I) is true, then (II) is true.
We still have to add the unmentioned elements. First for the second step, let R_0 be as follows:
$R_0 := K_2 \cup \{\langle 0,0,0,\langle 0,0,0,y\rangle_2\rangle_1 : \forall a,b,...,w\langle a,b,i,\langle c,d,j,y\rangle_v\rangle_w \notin K_1\}$
Then define $R := R_0 \cup \{\langle 0,0,0,y\rangle_2 : \forall a,b,i,v, \langle a,b,i,y\rangle_v \notin R_0\}$
This R represents F.

□

Now, if the map $S : \mathcal{P}(\omega)^3 \to \mathcal{P}(\omega)$ sending (A,B,C) to $(A \cdot C \cdot (B \cdot C))$ is continuous, it has a coded-continuous extension and we can conclude that this application gives a pca. First to prove that the application map is continuous, we can make that statement a bit stronger.

Lemma 4.3.3. *The application map* $p : \mathcal{P}(\omega)^2 \to \mathcal{P}(\omega)$, $(A,B) \mapsto (A \cdot B)$ *is coded-continuous.*

Proof: Note that with (A,B) in the domain of p, we have that for all x, $I_x(A,B) \Leftrightarrow \neg II_x(A,B)$ and $I_x(A,B) \vee II_x(A,B)$. Take x a natural number. Define $W_x^1 := \{(A.B) : I_x(A,B)\} = \bigcup_{p,q,i} U_{\{\langle p,q,i,x\rangle_1\}} \times U_{e_p}^{e_q}$ and $W_x^2 := \{(A,B) : II_x(A,B)\} = \bigcup_{p,q,i} U_{\{\langle p,q,i,x\rangle_2\}} \times U_{e_p}^{e_q}$. These are obviously opens. With these we have that:
$V_1^x := \{(A,B) \in dom(p) : x \in p(A,B)\} = \{(A,B) : I.x \wedge \neg II.x \wedge \forall y : ((I.x \wedge \neg II.x) \vee (II.y \wedge \neg I.y))\} = (W_x^1 - W_x^2) \cap \bigcap_y((W_y^1 - W_y^2) \cup (W_y^2 - W_y^1))$.
The same way we get that $V_x^2 := \{(A,B) \in dom(p) : x \notin p(A,B)\} = (W_x^2 - W_x^1) \cap \bigcap_y((W_y^1 - W_y^2) \cup (W_y^2 - W_y^1))$.
So p is coded-continuous.

□

Theorem 4.3.4. $\mathcal{P}(\omega)$ *with the defined application is a pca.*

Proof: We have already seen that $K : (a,b) \mapsto a$ is representable with this application by lemma 4.3.1. Now for $S : (a,b,c) \mapsto (a \cdot c \cdot (b \cdot c))$. Note that

we can decompose this map into:
$(a, b, c) \mapsto (a, b, a, c) \mapsto (a \cdot b, a, c) \mapsto (a \cdot b, a \cdot c) \mapsto (a \cdot c \cdot (a \cdot c))$.
We see that all these maps are continuous, so S is continuous. We can conclude that it has a coded-continuous extension, hence by lemma 4.3.2 we know it is represented. By theorem 2.1.4 we now know the pas is a pca.

□

We call this pca the double graph model. The fact that the Cantor topology forms a pca-topology pair with this model is now an immediate result. Note that since the application map is continuous, the Cantor topology also is full-repcon and full-conrep to this pca.

4.3.3 From DP to $\mathcal{K}_2$

In this section, we will see that there is a decidable single-valued applicative morphism form D**P** to $\mathcal{K}_2$.
First note that we can find another basis for the topology on $\mathbf{N}^{\mathbf{N}}$, given by the sets of the form $U_{i,n} = \{\alpha : \alpha(i) = n\}$. We have that $U_\sigma = \bigcap_{0 \leq i < length(\sigma)} U_{i,\sigma_i}$ and $U_{i,n} = \bigcup_{x_0} \cdots \bigcup_{x_{n-1}} U_{(x_1,\ldots,x_{n-1},n)}$, so it is indeed another basis.

Define the following single-valued morphism $\kappa : D\mathbf{P} \to \mathcal{K}_2$ where:

$$\kappa(A)(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Take $\mathcal{K}_2' := im(\kappa)$ and define the map $r : \mathcal{K}_2' \times \mathcal{K}_2' \to \mathcal{K}_2$ where $(\kappa(A), \kappa(B)) \mapsto \kappa(A * B)$ if $A * B \downarrow$, otherwise leave it undefined. Here $*$ denotes application in D**P**. Let $l : D\mathbf{P} \times D\mathbf{P} \to \mathcal{K}_2$ be the map which sends (A, B) to $\kappa(A * B)$ if defined. When we look at the definition of $*$, we can see that for A and B with $A * B \downarrow$, we have that:
$l(A, B) \in U_{x,y} \Leftrightarrow \kappa(A * B)(x) = y \Leftrightarrow (y = 0 \wedge x \notin A * B) \vee (y = 1 \wedge x \in A * B) \Leftrightarrow (y = 0 \wedge (A, B) \in \bigcup_{n,m,i} U_{\{\langle n,m,i,x \rangle_2\}} \times U_{e_n}^{e_m}) \vee (y = 1 \wedge (A, B) \in \bigcup_{n,m,i} U_{\{\langle n,m,i,x \rangle_1\}} \times U_{e_n}^{e_m})$. So $l^{-1}(U_{x,y})$ is open in the domain of l. Let $V_{x,y}$ be the Cantor open set such that $V_{x,y} \cap dom(l) = l(U_{x,y})^{-1}$. Then $V_{x,y} = \bigcup_{i \in I} U_{p_i}^{q_i} \times U_{r_i}^{s_i}$ for some I, p_i etc. Note that for $i \in I$ we have:
$(A, B) \in U_{p_i}^{q_i} \times U_{r_i}^{s_i} \Leftrightarrow (\kappa(A), \kappa(B)) \in (\bigcap_{x \in p_i} U_{x,1} \times \mathcal{K}_2) \cap (\bigcap_{x \in q_i} U_{x,0} \times \mathcal{K}_2) \cap (\bigcap_{x \in r_i} \mathcal{K}_2 \times U_{x,1}) \cap (\bigcap_{x \in s_i} \mathcal{K}_2 \times U_{x,0})$.

This is an open set. So $r^{-1}(U_{x,y})$ is a finite union of opens, hence itself also open. We can conclude that r is continuous and hence has a representation in $\mathcal{K}_2$. So κ is an applicative morphism.

$\square$

This also means that D**P** can be modelled in **P**.

4.4 Power set of a pca

Many structures defined on pca's work with subsets of its underlying set, like assemblies, the decidability of sets or images of representable functions. Given some pca, there is a way to construct a model on its set of all non-empty subsets. This new pca gives us more information about these set-like structures.

If $\mathcal{A}$ is a pca, we define a pas on $\mathcal{P}^*(\mathcal{A}) := (\mathcal{P}(\mathcal{A}) - \{\emptyset\})$ in the following way. Remember the coding of sequences: $[u_0, ..., u_{n-1}] = p\overline{n}j^n(u_0, ..., u_{n-1})$ (where j^n can be represented by an element p_n). We define a map $M : \mathcal{P}^*(\mathcal{A}) \to \mathcal{P}^*(\mathcal{A})$, where for $A \in \mathcal{P}^*(\mathcal{A})$ we take $M(A) := \{[a_0, a_1, ..., a_{n-1}] : n > 0, a_0, ..., a_{n-1} \in A\} \subseteq \mathcal{A}$, which is the encoding of finite non-empty sequences of elements from A. We define an application a follows: For all $A, B \in \mathcal{P}^*(\mathcal{A})$, $AB\downarrow$ if there are $a \in A$ and $b \in M(B)$ such that $ab\downarrow$. If $AB\downarrow$, then:

$$AB := \{ab : a \in A, b \in M(B), ab\downarrow\}$$

We have a few functions with which we can study and alter finite sequences:
$l_s[] = \overline{0}$, $l_s[a_0, ..., a_{n-1}] = \overline{n}$
$b_s[u_0, ..., u_{n-1}]\overline{i} = u_i$ if $i < n$.
$c_s[u_0, ..., u_{n-1}]\overline{i}\overline{j} = [u_i, ..., u_{j-1}]$ if $i \geq 0$, $j \leq n$ and $i < j$
We can hence find an element $v := \langle x\rangle bx\overline{0}$ such that $v[u_0, ..., u_{n-1}] = u_0$. To study and alter numerals we have S giving successors, P giving predecessors and Z checking whether the numeral is zero and giving Booleans accordingly. So for all $0 \leq i < j$ we can find an element $s_{(i,j)}$ such that $s_{(i,j)}[u_0, ..., u_n] = [u_i, ..., u_{j-1}]$ if $n \geq j$. For $j > 1$, we can design it in such a way that $s_{(i,j)}[u_0, ..., u_{n-1}] = [u_0]$ if $j > n$ (this can be achieved using l_s, P, Z and $s_{(0,0)}$). We do not really care what happens when $n = 0$. In the same way, we

can also take for all $i \geq 0$ an element e_i such that $e_i[u_0, ..., u_{n-1}] = u_{min(i,n-1)}$.

Now we have all the tools to prove the pas is a pca. We take the set $K := \{\langle x \rangle k(vx)\}$. Then for $A, B \in \mathcal{P}^*(\mathcal{A})$ we have $K \cdot A \cdot B = \{k(fa) : a \in M(A), k(va) \downarrow\} \cdot B = \{ka : a \in A\} \cdot B = \{kab : a \in A, b \in M(B)\} = A$. So K acts as the k in theorem 2.1.4.
Now for the s in theorem 2.1.4, which is more difficult. For all n and all $0 \leq m_0 < m_1 < ... < m_n$ we take the element:
$t^n_{(m_0,...,m_n)} := \langle xyz \rangle vx(s_{(0,m_0)}z)(p_n(e_0 y(s_{(m_0,m_1)}z))...(e_{n-1}y(s_{(m_{n-1},m_n)}z))$.
Note that for $A, B, C \in \mathcal{P}^*(\mathcal{A})$ and $x \in M(A)$, $y \in M(B), z \in M(C)$ we have that $t^n_{(m_0,...,m_n)}xyz$ gives us an element of the form $a'c'd'$ with $a' \in A$, $c' \in M(C)$ and $d' \in M(BC)$. Hence $t^n_{(m_0,...,m_n)}xyz \in AC(BC)$. We define $S := \{t^n_{(m_0,...,m_n)} : n > 0, 0 < m_0 < m_1 < ... < m_n\}$. Since for any X, the sequences in $M(X)$ can be of arbitrary length, it is easy to see that $SABC$ gives us all the applications of the form described above. So if $AC(BC) \downarrow$, then $SABC = AC(BC)$. Hence K and S satisfy theorem 2.1.4. We can conclude that the pas is a pca. We denote this pca by $\mathcal{P}^*(\mathcal{A})$.

Remarks:
1) There is an applicative morphism $\gamma : \mathcal{A} \to \mathcal{P}^*(\mathcal{A})$ given by $a \mapsto \{a\}$ and represented by $r := \{\langle xy \rangle vx(vy)\}$.
2) Let $f : \mathcal{A}^n \rightharpoonup \mathcal{A}$ be a partial map represented by r, take $R := \{\langle x_0, ..., x_{n-1} \rangle r(vx_0)...(vx_{n-1})\}$. Then for $A_0, ..., A_{n-1} \in \mathcal{P}^*(\mathcal{A})$ we have $R \cdot A_0 \cdot ... \cdot A_{n-1} = f(A_0, ..., A_{n-1})$, the image of f over $A_0 \times ... \times A_{n-1}$.
3) Any morphism of assemblies can be seen as a partial map $f : \mathcal{P}^*(\mathcal{A}) \to \mathcal{P}^*(\mathcal{A})$ which is representable.

We will look at some interesting applicative morphisms.

Consider Kleene's first model $\mathcal{K}_1$, and let all notations of sequences, operators and M map be borrowed from that model. We can create $\mathcal{P}^*(\mathcal{K}_1)$. The underlying set of this model is almost the same as the underlying set of Scott's Graph Model. There is an applicative morphism $\gamma : \mathcal{P}^*(\mathcal{K}_1) \to \mathbf{P}$ sending $A \mapsto \{M(A)\}$. We define its representation. Let R be the set containing all elements of the following form:
Let $n > 0, 0 < m_1 < ... < m_n, a_0, ..., a_{n-1}, b_0, ..., b_{m_n} \in \mathbf{N}$. Now take $x := [a_0, ..., a_{n-1}], y := [b_0, ..., b_{m_n}]$. If $\phi_{a_0}([b_0, ..., b_{m_1}]) \downarrow$,...,$\phi_{a_{n-1}}([b_{m_{n-1}}, ..., b_{m_n}]) \downarrow$, then we include $\langle 2^x, \langle 2^y, [\phi_{a_0}([b_0, ..., b_{m_1}]), ..., \phi_{a_{n-1}}([b_{m_{n-1}}, ..., b_{m_n}])] \rangle \rangle$ in R.

Note that sequential notation is borrowed from $\mathcal{K}_1$, while the pairing notation is the one from $\mathbf{P}$. So the sequential coding from $\mathcal{K}_1$ is coded into R. Now, for $A, B \in \mathcal{P}(\mathcal{K}_1)^*$, $X \in \gamma(A)$ and $Y \in \gamma(B)$, we have $X = M(A)$ and $Y = M(B)$. Then $R \cdot A \cdot B$ in $\mathbf{P}$ is always defined, and consists of $[\phi_{a_0}([b_0, ..., b_{m_1}]), ..., \phi_{a_{n-1}}([b_{m_{n-1}}, ..., b_{m_n}])]$ such that $[a_0, ..., a_{n-1}] \in X = M(A)$ and $[b_0, ..., b_{m_n}] \in Y = M(B)$. These are precisely all combination such that $a_0, ..., a_{n-1} \in A$ and $[b_0, ..., b_{m_1}], ..., [b_{m_{n-1}}, ..., b_{m_n}] \in M(B)$. So we get $M(AB)$. We can conclude that $R \cdot A \cdot B \in \{M(AB)\} = \gamma(AB)$.

Another applicative morphism flows from a more traditional structure on the powerset of a pca $\mathcal{A}$. We can define a partial applicative structure on $\mathcal{P}^*(\mathcal{A})$ as follows:
For all $A, B \in \mathcal{P}^*(\mathcal{A})$, let $A \cdot B \downarrow$ if for all $a \in A$ and $b \in B$ we have $ab \downarrow$. If that is the case, we define

$$A \cdot B = \{ab : a \in A, b \in B\}$$

We denote this pas by $\mathcal{O}(\mathcal{A})$. This is not a pca, but it does satisfy the conditions of what is called an *order-pca*. This is a pas with an order defined on its set, satisfying a condition of combinatory completeness in which terms only need to be represented by a representable map which is always smaller than that term (we will not need to go further into this definition). In this case, the order is given by the inclusion of sets. Useful facts about $\mathcal{O}(\mathcal{A})$ are laid out in [16].
Though we have not explored the concepts of order-pca's, we can consider applicative morphisms from these structures to normal pca's. They are defined the same way, needing some representing element in the target pca. There is an applicative morphism from $\delta : \mathcal{O}(\mathcal{A}) \to \mathcal{P}^*(\mathcal{A})$ defined as $A \mapsto \{A\}$. Define a term in $\mathcal{P}^*(\mathcal{A})$ as $R := \{\langle xy \rangle vx(vy)\}$. For all $A, B \in \mathcal{O}(\mathcal{A})$ such that $A \cdot B \downarrow$ in $\mathcal{O}(\mathcal{A})$, and for all $X \in \delta(A)$, $Y \in \delta(B)$ we have that if $RXY \downarrow$, then $RXY = RAB = \{\langle y \rangle va(vy) : a \in M(A)\}B = \{\langle y \rangle a(vy) : a \in A\}B = \{a(vb) : a \in A, b \in M(B), a(vb) \downarrow\} = \{ab : a \in A, b \in B, ab \downarrow\}$. Since $A \cdot B \downarrow$, we have that $ab \downarrow$ for all $a \in A$ and $b \in B$. So $RXY \downarrow$ and $RXY = A \cdot B \in \{A \cdot B\} = \delta(A \cdot B)$. We can conclude that δ is an applicative morphism. An applicative morphism the other way has not yet been found.

4.5 Recursive subsystem

We have seen many examples of pca's, though their complexity might hinder implementation in the real world. There are for instance some infinitely complex sets which are impossible to code into computers. It is therefore handy to look at the recursive parts of pca's. Preliminary results are discussed in Bauer's thesis [1].

4.5.1 Sub-pca's

We consider the set of recursive functions on the natural numbers. This is the smallest set which contains the constant functions, the successor function and the projection functions, and is closed under composition, primitive recursion and μ-recursion.

We can look at the set of total recursive functions in one variable, which form a subset of $\mathbf{N}^{\mathbf{N}}$. This is the same set on which Kleene's second model acts. It has been established that in $\mathcal{K}_2$, we can actually find total recursive functions k and s that act as in theorem 2.1.4 (mainly μ-recursion is used). So the total recursive endofunctions with application borrowed from $\mathcal{K}_2$ form a sub-pca $\mathcal{K}_2^{\#}$.

We can do the same for Scott's Graph model. We consider a subset A of the natural numbers to be recursive if there is some recursive endofunction f satisfying $f^{-1}(\{x \in \mathbf{N} : x > 0\}) = A$. The set of recursive sets borrowing the application map from $\mathbf{P}$ form a sub-pca $\mathbf{P}^{\#}$.

Lastly, we can consider a Cantor-open set $x \in \mathbf{U}$ to be computable if the set of all clopens contained in x is itself recursively enumerable. The embedding-projection pair used to define the applicative structure on $\mathbf{U}$ can be chosen to be computable, making the subset of computable opens into a sub-pca $\mathbf{U}^{\#}$.

4.5.2 Relative representability

We have considered three pca's, each with an interesting sub-pca. It is possible to now consider what we can represent relative to these subsystems.

Given two pca's $\mathcal{A}' \subset \mathcal{A}$, we call a function $f : \mathcal{A}^n \to \mathcal{A}$ *relative representable* w.r.t. $\mathcal{A}'$ if it is representable in $\mathcal{A}$ and has a representing element from $\mathcal{A}'$. We do the same to define relative representability for applicative morphisms and relations between applicative morphisms, by demanding their representing elements to be in the sub-pca. It has for instance been established in [1] that the applicative retraction between $\mathbf{P}$ and $\mathcal{K}_2$, and the applicative inclusion between $\mathbf{P}$ and $\mathbf{U}$ are both representable relative to their sub-pca's, meaning that the morphisms are relative representable and the applicative retraction and inclusion relations too.

Consider two pca's $\mathcal{A}' \subseteq \mathcal{A}$ with inclusion $\iota : \mathcal{A}' \to \mathcal{A}$ and a map $f : \mathcal{A} \to \mathcal{A}$ which is not necessarily representable. Now we look at both $\gamma_f : \mathcal{A} \to \mathcal{A}[f]$ and $\gamma'_f : \mathcal{A}' \to \mathcal{A}'[f]$, which are both given by the identity function. Since f is representable w.r.t. γ_f, it is representable w.r.t. $\gamma_f \circ \iota$. ι is obviously decidable, so we can conclude that there is a decidable applicative morphism $\iota_f : \mathcal{A}'[f] \to \mathcal{A}[f]$ such that $\iota_f \circ \gamma'_f = \gamma_f \circ \iota$. By looking at the specific definitions of the maps, we can see that ι_f is also an inclusion. Hence, $\mathcal{A}'[f]$ is a sub-pca of $\mathcal{A}[f]$. So, even in $\mathcal{A}[f]$, f can be represented with an element from $\mathcal{A}'[f]$. Also, since γ_f is represented by $i \in \mathcal{A}'[f]$, we can say it is relatively representable by that sub-pca. γ'_f can be considered as the applicative morphism γ_f limited with its domain limited to $\mathcal{A}'$.

We can now look at our system of extensions and see in what extend it carries this relative representability. As discussed, the original applicative retraction and inclusion are relative representable, and so are the morphisms towards the extensions. If we look back at the proofs in 4.2 about the representability of S, C and Z w.r.t. to the applicative morphisms, we see that they only use representations which are also valid in the sub-pca's and only use recursive constructions. Hence we get a system of extensions on the sub-pca's forming pull-back squares like in the original models:

$$\begin{array}{ccccc}
\mathcal{K}_2^{\#} & \substack{\longleftarrow \\ \longrightarrow} & \mathbf{P}^{\#} & \substack{\longleftarrow \\ \longrightarrow} & \mathbf{U}^{\#} \\
\downarrow & & \downarrow & & \downarrow \\
\mathcal{K}_2^{\#}[S] & \substack{\longleftarrow \\ \longrightarrow} & \mathbf{P}^{\#}[C] & \substack{\longleftarrow \\ \longrightarrow} & \mathbf{U}^{\#}[Z]
\end{array}$$

Chapter 5

Conclusion

Many pca's have been studied in order to further understand concepts of realizability. As of now, a full picture of what this topic entails has yet to be found. But one need not always look at the big picture. Sometimes, it may be nice to just look at the little things.

This thesis adds some new pca's to the fold. Scott's Graph Model, a structure with the capabilities to simulate the untyped lambda calculus, can be extended to be made decidable. We get the least pca to represent the complement function and simulate the Graph Model.

Relations with other famous models can be used to find more extensions. Decidable applicative morphisms could be lifted to these extensions, making full use of the properties in theorem 2.3.5. The complement function induces an image checking function in Kleene's second model, and a partial complement function in the universal domain model. So non-representable functions can have 'siblings' acting similarly in other related pca's.

To fully understand the possibilities of simulation within a pca, one can sometimes find a topology to describe the set of representable functions. This is possible for the three fundamental pca's described above, but not always. And not all topologies describe the set of representable functions for some pca. We can however construct a pca for the Cantor topology. Can it be done for others?

Some of the models could be implemented within a computer, which could yield real life applications. But further research must be done. It is my hope that many people will study this increasingly interesting field.

Bibliography

[1] A. Bauer. *The Realizability Approach to Computable Analysis and Topology.* PhD thesis, Carnegie Mellon University, Pittsburgh, 2000.

[2] I. Bethke. *Notes on Partial Combinatory Algebras.* PhD thesis, Universiteit van Amsterdam, 1988

[3] E. Faber and J. Van Oosten. *Effective Operations of Type 2 in Pcas*, university of Utrecht, 2014

[4] E. Faber and J. Van Oosten. *More on geometric morphisms between realizability toposes*, Theory and Application of Categories, 29:1-22, 2014

[5] S. Feferman. *A language and axioms for explicit mathematics.* In J.N. Crossley, editor, Algebra and Logic, pages 87139. Springer-Verlag, 1975.

[6] J.W. Hofstra. *Partial Combinatory Algebras and Realizability Toposes*, University of Ottowa, 2004

[7] P. Hofstra and J. van Oosten. *Ordered partial combinatory algebras.* Math. Proc. Camb. Phil. Soc., 134:445-463, 2003

[8] 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.

[9] P.T. Johnstone. *Geometric morphisms of Realizability Toposes.* Theory and Appliation of Categories, 28(32):1139-1152,2013

[10] S.C. Kleene. *On the interpretation of intuitionistic number theory.* Journal of Symbolic Logic, 10:109124, 1945.

[11] J.R. Longley. *Realizability toposes and language semantics*, university of Edinburgh, 1994

[12] M. Schönfinkel. *Über die Bausteine der Mathematischen Logik*. Mathematische Annalen, 92:305316, 1924.

[13] D.S. Scott. *Continuous lattices.* In W. Lawvere, editor, Toposes, Alebraic Geometry and Logic, pages 97136, Berlin, 1972. Springer (LNM 274).

[14] D.S. Scott. *Lambda calculus and recursion theory.* In S. Kanger, editor, Proc. Third Scand. Log. Symp, pages 154193. North Holland, 1975.

[15] J. van Oosten. *A combinatory algebra for sequential functionals of finite type.* Models and Computability, pages 389-406. Cambridge University Press, 1999.

[16] J. Van Oosten. *Realizability, an introduction to its categorical side*, the foundations of mathematics 152, Elsevier, 2008.

[17] J. van Oosten. *Partial Combinatory Algebras of functions*, Notre Dame Journal of formal logic, 52:4, 2011

Table of symbols

$\mathbf{N}$	Set of natural numbers.
$A, B, C, ...$	Non-unique instances of sets
A^c	$(\mathbf{N} - A)$, the complement of the set A
$\mathcal{A}, \mathcal{B}, \mathcal{C}...$	Non-unique instances of pca's
k, s	Elements of a pca satisfying theorem 2.1.4
X^*	Set of finite sequences with elements from X
$\mathcal{P}(X)$	Power set of X
$\mathcal{P}^*(X)$	Power set of X excluding the empty set.
X^Y	Set of functions from Y to X
$\mathbf{T}(\mathcal{A})$	Realizability tripos of the pca $\mathcal{A}$
$RT(\mathcal{A})$	Realizability topos of the pca $\mathcal{A}$
$\mathcal{I}$	The trivial pca
$\mathcal{K}_1$	Kleene's first model
$\mathbf{P}$	Scott's Graph Model
$\mathcal{K}_2$	Kleene's second model
$\mathbf{U}$	Universal domain model
$\mathcal{A}[f]$	Pca extended with the map f
$D\mathbf{P}$	Double graph model
$\mathcal{P}^*(\mathcal{A})$	Power set of the pca $\mathcal{A}$
$\mathcal{O}(\mathcal{A})$	order-pca on the powerset of the pca $\mathcal{A}$
$a \sqsubseteq \alpha$	The sequence α starts with the finite sequence a.